MG-RAST Manual for version 4, revision 0

October 3nd, 2016

Andreas Wilke[1,2], Wolfgang Gerlach[2,1], Travis Harrison[2,1], Tobias Paczian[2,1], William L. Trimble[2,1] and Folker Meyer[1,2]

[1]Argonne National Laboratory
[2]University of Chicago

Contents

Chapter 1

Introduction

1.1 Motivation

MG-RAST provides Science as a Service for environmental DNA at `http://metagenomics.anl.gov/`.

The National Human Genome Research Institute (NHGRI), a division of the National Institutes of Health, publishes information (see Figure 1.1) describing the development of computing costs and DNA sequencing costs over time [16]. The dramatic gap between the shrinking costs of sequencing and the more or less stable costs of computing is a major challenge for biomedical researchers trying to use next-generation DNA sequencing platforms to obtain information on microbial communities. Wilkening *et al.* [44] provide a real currency cost for the analysis of 100 gigabasepairs of DNA sequence data using BLASTX on Amazon's EC2 service: $300,000.[1] A more recent study by University of Maryland researchers [1] estimates the computation for a terabase of DNA shotgun data using their CLOVR metagenome analysis pipeline at over $5 million per terabase.

Nevertheless, the growth in data enabled by next-generation sequencing platforms also provides an exciting opportunity for studying microbial communities: 99% of the microbes in which have not yet been cultured [34]. Cultivation-free methods (often summarized as metagenomics) offer novel insights into the biology of the vast majority of life on Earth [38].

Several types of studies use DNA for environmental analyses:

1. Environmental clone libraries (functional metagenomics): use of Sanger sequencing (frequently) instead of more cost-efficient next-generation sequencing

2. Amplicon metagenomics (single gene studies, 16s rDNA): next-generation sequencing of PCR amplified ribosomal genes providing a single reference gene–based view of microbial community ecology

[1] This includes only the computation cost, no data transfer cost, and was computed using 2009 prices.

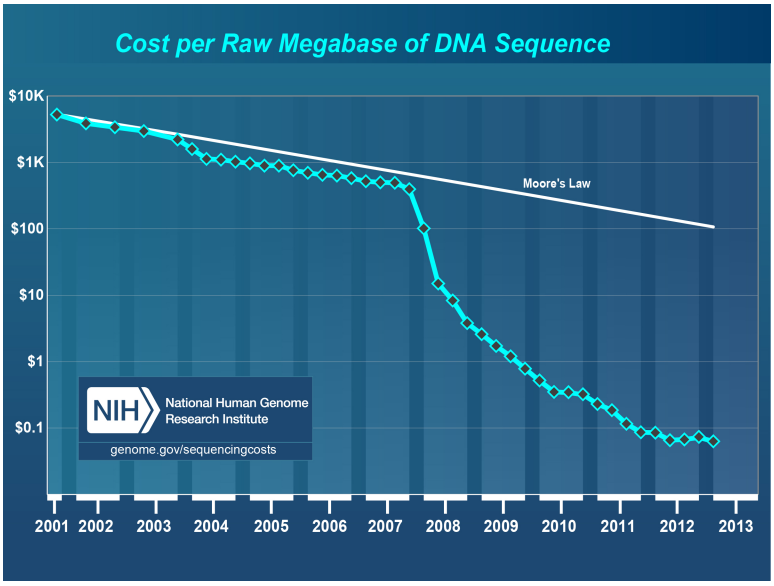


Figure 1.1: Chart showing shrinking cost for DNA sequencing. This comparison with Moore's law roughly describing the development of computing costs highlights the growing gap between sequence data and the available analysis resources. Source: NHGRI [16]

3. Shotgun metagenomics: use of next-generation technology applied directly to environmental samples

4. Metatranscriptomics: use of cDNA transcribed from mRNA

Each of these methods has strengths and weaknesses (see [38]), as do the various sequencing technologies (see [22]).

To support user-driven analysis of all types of metagenomic data, we have provided MG-RAST [26] to enable researchers to study the function and composition of microbial communities. The MG-RAST portal offers automated quality control, annotation, comparative analysis, and archiving services. At the time of writing MG-RAST has completed the analysis of over 100 terabasepairs of DNA data in over 250,000 datasets contributed by thousands of researchers worldwide.

The MG-RAST system provides answers to the following scientific questions:

- Who is out there? Identifying the composition of a microbial community either by using amplicon data for single genes or by deriving community composition from shotgun metagenomic data using sequence similarities.

- What are they doing? Using shotgun data (or metatranscriptomic data) to derive the functional complement of a microbial community using similarity searches against a number of databases.

- Who is doing what? Based on sequence similarity searches, identifying the organisms encoding specific functions.

The system supports the analysis of the prokaryotic content of samples, analysis of viruses and eukaryotic sequences is not currently supported, due to software limitations.

MG-RAST users can upload raw sequence data in fastq, fasta and sff format; the sequences will be normalized (quality controlled) and processed and summaries automatically generated. The server provides several methods to access the different data types, including phylogenetic and metabolic reconstructions, and the ability to compare the metabolism and annotations of one or more metagenomes, individually or in groups. Access to the data is password protected unless the owner has made it public, and all data generated by the automated pipeline is available for download in variety of common formats.

1.2 Brief description

The MG-RAST pipeline performs quality control, protein prediction, clustering and similarity-based annotation on nucleic acid sequence datasets using a number of bioinformatics tools (see Section C.2.1. MG-RAST was built to analyze large shotgun metagenomic data sets ranging in size from megabases to terabases. We also support amplicon (16S, 18S, and ITS) sequence datasets and metatranscriptome (RNA-seq) sequence datasets. The current MG-RAST pipeline is not capable of predicting coding regions from eukaryotes and thus will be of limited use for eukaryotic shotgun metagenomes and/or the eukaryotic subsets of shotgun metagenomes.

Data on MG-RAST is private to the submitting user unless shared with other users or made public by the user. We strongly encourage the eventual release of data and require metadata ("data describing data") for data sharing or publication. Data submitted with metadata will be given priority for the computational queue.

You need to provide (raw or assembled) nucleotide sequence data and sample descriptions ("metadata"). The system accepts sequence data in FASTA, FASTQ and SFF format and metadata in the form or GSC (`http://gensc.org/`) standard compliant checklists (see Yilmaz et al, Nature Biotech, 2011). Uploads can be put in the system via either the web interface or a command line tool. Data and metadata are validated after upload.

You must choose quality control filtering options at the time you submit your job. MG-RAST provides several options for quality control (QC) filtering for nucleotide sequence data, including removal of artificial duplicate reads, quality-based read trimming, length-based read trimming, and

screening for DNA of model organisms (or humans). These filters are applied before the data are submitted for annotation.

The MG-RAST pipeline assigns an accession number and puts the data in a queue for computation. The similarity search step is computationally expensive. Small jobs can complete as fast as hours, while large jobs can spend a week waiting in line for computational resources.

MG-RAST performs a protein similarity search between predicted proteins and database proteins (for shotgun) and a nucleic-acid similarity search (for reads similar to 16S and 18S sequences).

MG-RAST presents the annotations via the tools on the analysis page which prepare, compare, display, and export the results on the website. The download page offers the input data, data at intermediate stages of filtering, the similarity search output, and summary tables of functions and organisms detected.

MG-RAST can compare thousands of data sets run through a consistent annotation pipeline. We also provide a means to view annotations in multiple different namespaces (e.g. SEED functions, K.O. Terms, Cog Classes, EGGnoggs) via the M5Nr.

The publication "Metagenomics-a guide from sampling to data analysis" (PMID 22587947) in Microbial Informatics and Experimentation, 2012 is a good review of best practices for experiment design for further reading.

1.3 URL

`http://metagenomics.anl.gov/`

1.4 Citing MG-RAST

A significant number of papers have been published about MG-RAST itself and the supporting platform, however we ask that if you use the system please cite:
The Metagenomics RAST server — A public resource for the automatic phylogenetic and functional analysis of metagenomes
F. Meyer, D. Paarmann, M. D'Souza, R. Olson , E. M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, J. Wilkening, and R. A. Edwards
BMC Bioinformatics 2008, 9:386
`http://www.biomedcentral.com/1471-2105/9/386`.

In addition if you also use the API please cite:
A RESTful API for Accessing Microbial Community Data for MG-RAST
A. Wilke, J. Bischof, T. Harrison, T. Brettin, M. D'Souza, W. Gerlach, H. Matthews, T. Paczian, J. Wilkening, E. M. Glass, N. Desai, F. Meyer

PLOS Comp Biology 2015, DOI: 10.1371/journal.pcbi.1004008
http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004008.

1.5 Version history

Version 1

The original version of MG-RAST was developed in 2007 by Folker Meyer, Andreas Wilke, Daniel Paarman, Bob Olson, and Rob Edwards. It relied heavily on the SEED [28] environment and allowed upload of preprocessed 454 and Sanger data.

Version 2

Version 2, released in 2008, had numerous improvements. It was optimized to handle full-sized 454 datasets and was the first version of MG-RAST that was not fully SEED based. Version 2.0 used BLASTX analysis for both gene prediction and functional classification [26].

Version 3

While version 2 of MG-RAST was widely used, it was limited to datasets smaller than a few hundred megabases, and comparison of samples was limited to pairwise comparisons. Version 3 is not based on SEED technology; instead, it uses the SEED subsystems as a preferred data source. Starting with version 3, MG-RAST moved to github.

Version 3.6

With version 3.6 MG-RAST was containerized, moving from a bare metal infrastructure to a set of docker containers running in a Fleet/SystemD/etcD environment.

Version 4

Version 4.0 brings a new web interface, fully relying on the API for data access and moves the bulk of the data stored from Postgres to Cassandra. The new web interface moves the data visualization burden from the web server to the clients machine, using Javascript and HTML5 heavily.

In version 4.0 we have moved the changed the backend store for profiles. While previous version stored a pre-computed mapping of observed abundances to functional or taxonomic categories, this is now computed on the fly. The number of profiles stored is reduced to the MD5 and LCA profiles. The API has been augmented to allow dynamic mapping to categories, to provide the required bandwidth we have migrated the profile store from Postgres to Cassandra.

The web interface of the previous version predated the API, the user interface for version 4.0 now uses the API. The web interface has been re-written in JavaScript/HTML5. Unlike previous version the web interface now is executed on the client (inside the browser) and now supports any recent browser.

Comparison of versions 2 and 3

Version 3 added the ability to analyze massive amounts of Illumina reads by introducing a significant number of changes to the pipeline and the underlying platform technology. In version 3 we introduced the notion of the API as the central component of the system.

In the 3.0 version, datasets of tens of gigabases can be annotated, and comparison of taxa or functions that differ between samples is now limited only by the available screen real estate. Figure 1.2 shows a comparison of the analytical and computational approaches used in MG-RAST v2 and v3. The major changes are the inclusion of a dedicated gene-calling stage using FragGenescan [33], clustering of predicted proteins at 90% identified by using uclust [10], and the use of BLAT [20] for the computation of similarities. Together with changes in the underlying infrastructure, this version has allowed dramatic scaling of the analysis with the limited hardware available.

Similar to version 2.0, the new version of MG-RAST does not pretend to know the correct parameters for the transfer of annotations. Instead, users are empowered to choose the best parameters for their datasets.

Comparison of versions 3 and 4

The roadmap for version 4 has a number of key elements that will be implemented step-by-step, currently the following features are implemented:

4.0 New JavaScript web interface using the API

4.0 Cassandra instead of Postgres as main data store for profiles

The new version of MG-RAST represents a rethinking of core processes and data products, as well as new user interface metaphors and a redesigned computational infrastructure. MG-RAST supports a variety of user-driven analyses, including comparisons of many samples, previously too computationally intensive to support for an open user community.

Scaling to the new workload required changes in two areas: the underlying infrastructure needed to be rethought, and the analysis pipeline needed to be adapted to address the properties of the newest sequencing technologies.

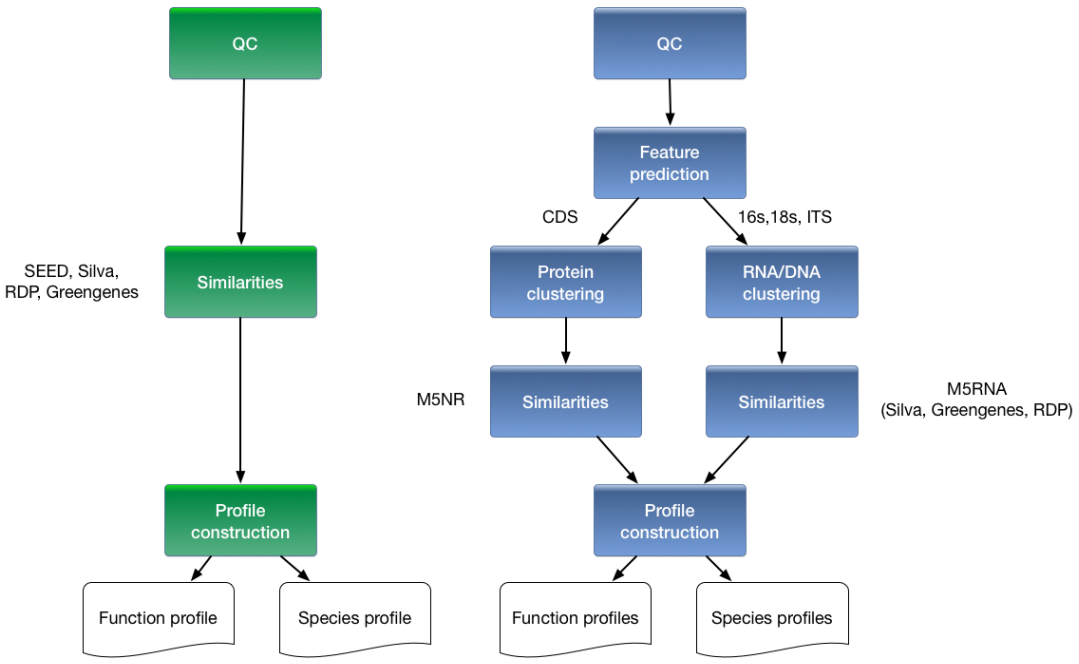


Figure 1.2: Overview of processing pipeline in (left) MG-RAST v2 and (right) MG-RAST v3. In the old pipeline, metadata was rudimentary, compute steps were performed on individual reads on a 40-node cluster that was tightly coupled to the system, and similarities were computed by BLAST to yield abundance profiles that could then be compared on a per sample or per pair basis. In the new pipeline, rich metadata can be uploaded, normalization and feature prediction are performed, faster methods such as BLAT are used to compute similarities, and the resulting abundance profiles are fed into downstream pipelines on the cloud to perform community and metabolic reconstruction and to allow queries according to rich sample and functional metadata.

1.6 The MG-RAST team

MG-RAST was started by Rob Edwards and Folker Meyer in 2007. The MG-RAST team has significantly expanded in the past few years. The team is listed below.

- Andreas Wilke
- Wolfgang Gerlach
- Travis Harrison
- Tobias Paczian
- William L. Trimble
- Folker Meyer

MG-RAST alumni

The following people were associated with MG-RAST in the past:

- Daniel Paarmann, 2007-2008
- Rob Edwards, 2007-2008
- Mike Kubal, 2007-2008
- Alex Rodriguez, 2007-2008
- Bob Olson, 2007-2009
- Daniela Bartels, 2007-2011
- Yekaterina Dribinsky, 2011
- Jared Wilkening, 2007-2013
- Mark D'Souza, 2007-2014
- Hunter Matthews 2009-2014
- Narayan Desai, 2011-2014
- Wei Tang, 2012-2015
- Elizabeth M. Glass, 2008-2016

- Jared Bischof, 2010-2016
- Kevin Keegan, 2009-2016
- Daniel Braithwaite, 2012-2015

Chapter 2

Under the hood: The MG-RAST technology platform

2.1 The backend

While originally MG-RAST data was stored in a shared filesystem and a MySQL database, the backend store evolved with growing popularity and demand.

Currently a number of data stores are used to provide the underpinning for various parts of the MG-RAST API.

An approximate mapping of stores to functions in version 4.0 is provided in table 2.1.

The backend infrastructure and the overall system layout is shown in figure 2.1.

As of version 3.6 the majority of the services are provisioned as containers, provisioned as a set of Fleet units described in `https://github.com/MG-RAST/MG-RAST-infrastructure/tree/master/fleet-units`.

Function	data store	comment
Search	Apache SOLR	
Profiles	Cassandra	
M5NR	Cassandra	
Authentication	MySQL	
Project	MySQL	
Access control	MySQL	
Metadata	MySQL	
Files	SHOCK	

Table 2.1: Mapping of API functions to data stores

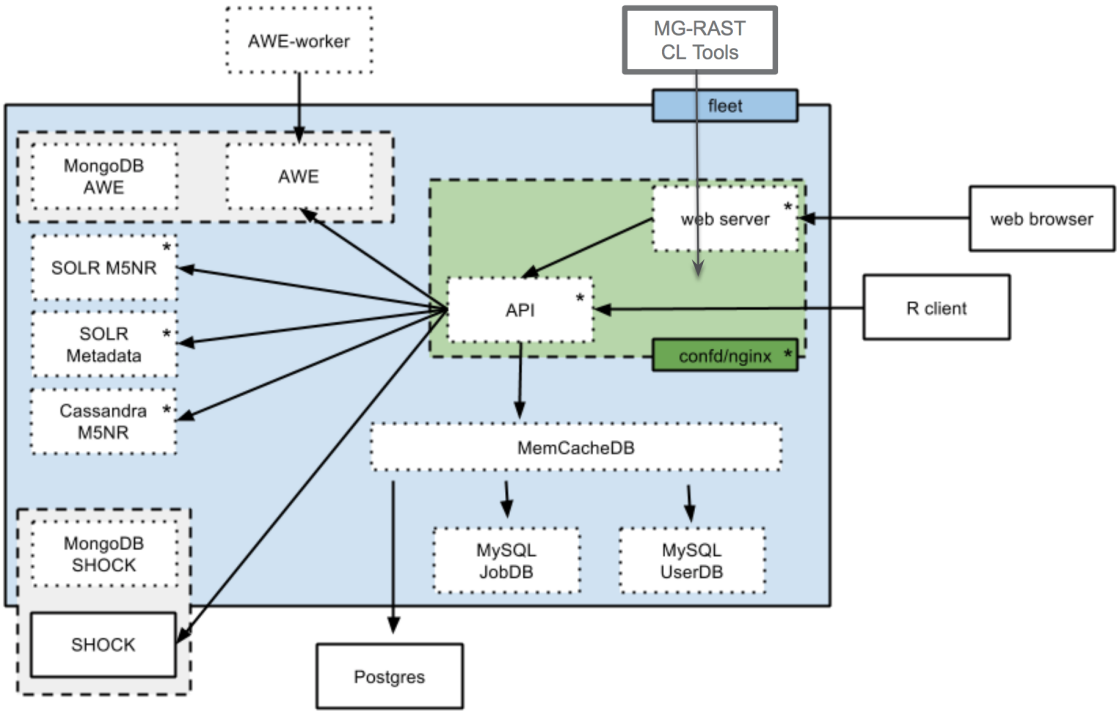


Figure 2.1: Overview of the production system in mid 2016. Fleet is used to manage a number of containerized services (shown with dashed lines). Two services are provisioned outside the Fleet system: SHOCK (providing 0.7 Petabyte of storage) and a Postgres clusters. We note the significant number of different databases used to serve data required for the API.

2.2 The supporting technologies: Skyport, AWE and SHOCK

One key aspect of scaling MG-RAST to large numbers of modern NGS datasets is the use of cloud computing[1], which decouples MG-RAST from its previous dedicated hardware resources.

We use AWE [42] an efficient, open source resource manager to execute the MG-RAST workflow. We expanded AWE to work with Linux containers forming the Skyport system [12]. AWE and Skyport use RESTful interfaces thus allowing the addition of clients without the need to add firewall exceptions and/or massive system reconfiguration.

The main MG-RAST data store is the the SHOCK data management system [43] developed alongside AWE. SHOCK like AWE relies on a RESTful interface instead of a more traditional shared file system.

When we introduced the technologies described above to replace a shared file system (Sun NFS mounted on several hundred nodes), we saw a speed up of a factor of 750x on identical hardware.

2.3 Data model

The MG-RAST data model (see Figure 2.2) has changed dramatically in order to handle the size of modern next-generation sequencing datasets. In particular, we have made a number of choices that reduce the computational and storage burden.

We note that the size of the derived data products for a next-generation dataset in MG-RAST is typically about 10x the size of the actual dataset. Individual datasets now may be as large as a terabase[2], with the on-disk footprint significantly larger than the basepair count because of the inefficient nature of FASTQ files, which basically double the on-disk size for FASTQ representations.

- Abundance profiles. Using abundance profiles, where we count the number of occurrences of function or taxon per metagenomic dataset, is one important factor that keeps the datasets manageable. Instead of growing the dataset sizes (often with several hundred million individual sequences per dataset), the data products now are more or less static in size.

- Single similarity computing step per feature type. By running exactly one similarity computation for proteins and another one for rRNA features, we have limited the computational requirements.

- Clustering of features. By clustering features at 90% identity, we reduce the number of times we compute on similar proteins. Abundant features will be even more efficiently clustered, leading to more compression for abundant species.

[1] We use the term *cloud* as a shortcut for Infrastructure as a Service (IaaS).

[2] This would be for several metagenomes that are part of the JGI Prairie pilot.

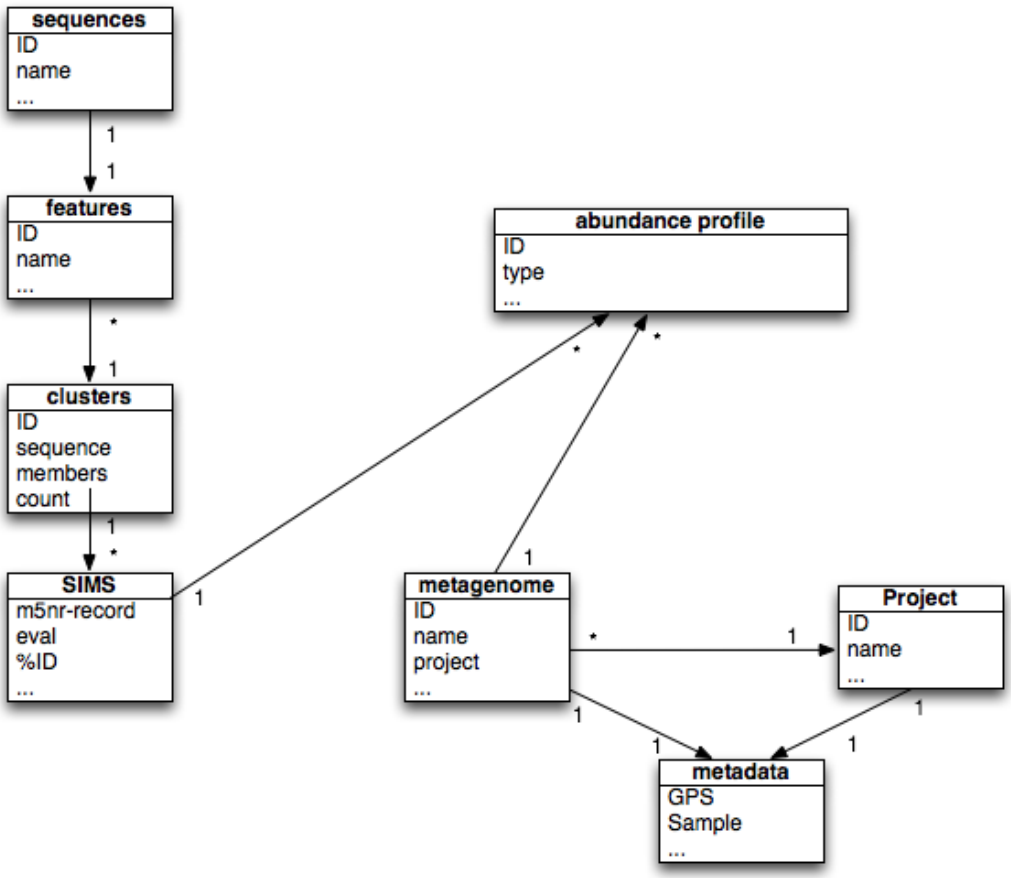


Figure 2.2: MG-RAST v3 data model.

THIS NEEDS TO BE REDONE!!!!!!

Figure 2.3: Analysis database schema: static objects (blue) and per metagenome (variable) objects (green).

As shown in Figure 2.2, MG-RAST relies on abundance profiles to capture information for each metagenome. The following abundance profiles are calculated for every metagenome.

- MD5s – number of sequences (clusters) per database entry in the M5nr.
- functions – summary of all the MD5s that match a given function.
- ontologies – summary of all the MD5s that match a given hierarchy entry.
- organisms – summary of all MD5s that match a given taxon entry.
- lowest common ancestors

The static helper tables (show in blue in Figure 2.3) help keep the main tables smaller, by normalizing and providing integer representations for the entities in the abundance profiles.

Chapter 3

The MG-RAST pipeline

MG-RAST provides automated processing of environmental DNA sequences via a pipeline. The pipeline has multiple steps that can be grouped into five stages:

We restrict the pipeline annotations to protein coding genes and ribosomal RNA (rRNA) genes.

- Data hygiene:
  Quality control and removal of artifacts.
- Feature extraction:
  Identification of protein coding and rRNA features (aka "genes")
- Feature annotation:
  Identification of putative functions and taxonomic origins for each of the features

  TRAVIS: DID WE EVER INCLUDE THE CONSENSUS FOR LONG CONTIGS
- Profile generation:
  Creation of multiple on disk representations of the information obtained above.
- Data loading:
  Loading the representations into the appropriate databases.

The pipeline shown in Figure 3.1 contains a significant number of improvements over previous versions and is optimized for accuracy and computational cost.

Using the M5nr [41] (an MD5 nonredundant database), the new pipeline computes results against many reference databases instead of only SEED. Several key algorithmic improvements were needed to support the flood of user-generated data (see Figure **??**). Using dedicated software to perform gene prediction instead of using a similarity-based approach reduces runtime requirements. The additional clustering of proteins at 90% identity reduces data while preserving biological signals.

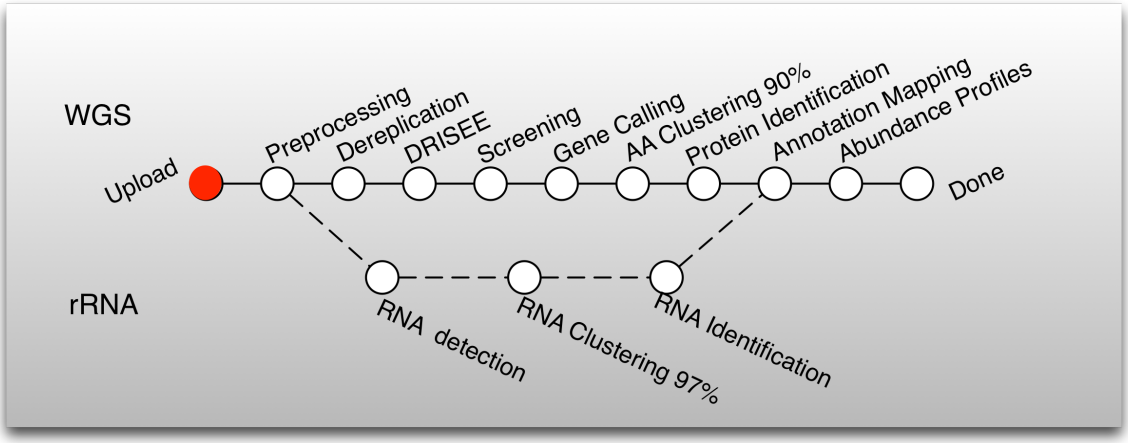


Figure 3.1: Details of the analysis pipeline for MG-RAST version 3

Below we describe each step of the pipeline in some detail. All datasets generated by the individual stages of the processing pipeline are made available as downloads. Appendix A lists the available files for each dataset.

3.1 Data hygiene

Preprocessing

After upload, data is preprocessed by using SolexaQA [8] to trim low-quality regions from FASTQ data. Platform-specific approaches are used for 454 data submitted in FASTA format: reads more than than two standard deviations away from the mean read length are discarded following [14]. All sequences submitted to the system are available, but discarded reads will not be analyzed further.

Dereplication

For shotgun metagenome and shotgun metatranscriptome datasets we perform a dereplication step. We use a simple k-mer approach to rapidly identify all 20 character prefix identical sequences. This step is required in order to remove Artificial Duplicate Reads (ADRs) [13]. Instead of simply discarding the ADRs, we set them aside and use them later for error estimation.

We note that dereplication is not suitable for amplicon datasets that are likely to share common prefixes.

DRISEE

MG-RAST v3 uses DRISEE (Duplicate Read Inferred Sequencing Error Estimation) [19] to analyze the sets of Artificial Duplicate Reads (ADRs) [13] and determine the degree of variation among prefix-identical sequences derived from the same template. See Section 4.2 for details.

Screening

The pipeline provides the option of removing reads that are near-exact matches to the genomes of a handful of model organisms, including fly, mouse, cow, and human. The screening stage uses Bowtie [21] (a fast, memory-efficient, short read aligner), and only reads that do not match the model organisms pass into the next stage of the annotation pipeline.

Note that this option will remove all reads similar to the human genome and render them inaccessible. This decision was made in order to avoid storing any human DNA on MG-RAST.

3.2 Feature identification

Protein coding gene calling

The previous version of MG-RAST used similarity-based gene predictions, an approach that is significantly more expensive computationally than de novo gene prediction. After an in-depth investigation of tool performance [39], we have moved to a machine learning approach: FragGeneScan [33]. Using this approach, we can now predict coding regions in DNA sequences of 75 bp and longer. Our novel approach also enables the analysis of user-provided assembled contigs.

We note that FragGeneScan is trained for prokaryotes only. While it will identify proteins for eukaryotic sequences, the results should be viewed as more or less random.

rRNA detection

NEEDS UPDATE An initial BLAT [20] search against a reduced RNA database efficiently identifies RNA. The reduced database is a 90% identity clustered version of the SILVA database and is used to rapidly identify sequences with similarities to ribosomal RNA.

3.3 Feature annotation

AA clustering

NEEDS UPDATE MG-RAST builds clusters of proteins at the 90% identity level using the uclust [10] implementation in QIIME [6] preserving the relative abundances. These clusters greatly

reduce the computational burden of comparing all pairs of short reads, while clustering at 90% identity preserves sufficient biological signals.

Protein identification

Once created, a representative (the longest sequence) for each cluster is subjected to similarity analysis. Instead of BLAST we use sBLAT, an implementation of the BLAT algorithm [20], which we parallelized using OpenMP [4] for this work.

Once the similarities are computed, we present reconstructions of the species content of the sample based on the similarity results. We reconstruct the putative species composition of the sample by looking at the phylogenetic origin of the database sequences hit by the similarity searches.

Sequence similarity searches are computed against a protein database derived from the M5nr [41], which provides nonredundant integration of many databases: GenBank, [3], SEED [28], IMG [24], UniProt [23], KEGG [18], and eggNOGs [17].

rRNA clustering

The rRNA-similar reads are then clustered at 97% identity, and the longest sequence is picked as the cluster representative. **NEEDS UPDATE**

rRNA identification

A BLAT similarity search for the longest cluster representative is performed against the M5rna database which integrates SILVA [29], Greengenes [9], and RDP [7].

3.4 Profile generation

In the final stage, the data computed so far is integrated into a number of data products. The most important one are the abundance profiles.

Abundance profiles represent a pivoted and aggregated version of the similarity files. We compute best hit, representative hit and LCA abundance profiles (see 4.5).

3.5 Database loading

In the final step the profiles are loaded into the respective databases.

Chapter 4

MG-RAST data products

MG-RAST provices a number of data products in a variety of formats.

Data formats

- Fasta and FastQ
  Sequence data can be downloaded via the API and web interface as Fasta (or FastQ) files
- JSON
  Metadata and Tables and other structured data can be downloaded via the APi or the web site in JSON format.
- Spreadsheet
  Metadata and Tables can be downloaded as spreadsheets via the web interface.
- SVG and PNG
  Images can be downloaded via the web site interface in SVG and PNG formast.
- BIOM v1
  BIOM [25] files can be downloaded via the web interface for use with e.g., QIIME [6].

Data types

- Sequence data
  The originally submitted sequence data as well as the various subsets resulting from processing can be downloaded.
- Metadata
  data describing data in GSC-compliant format.

- Analysis results – results of running the MG-RAST pipeline. The list includes all intermediate data products and is intended to serve as a basis for further analysis outside the MG-RAST pipeline.

  Details on the individual files are in Appendix A.

4.1 Abundance profiles

Abundance profiles are the primary data product that MG-RAST's user interface uses to display information on the datasets.

Using the abundance profiles, the MG-RAST system defers making a decision on when to transfer annotations. Since there is no well-defined threshold that is acceptable for all use cases, the abundance profiles contain all similarities and require their users to set cut-off values.

The threshold for annotation transfer can be set by using the following parameters: e-value, percent identity, and minimal alignment length.

The taxonomic profiles use the NCBI taxonomy. All taxonomic information is projected against this data. The functional profiles are available for data sources that provide hierarchical information. These currently comprise the following.

- SEED Subsystems

  The SEED subsystems [28] represent an independent reannotation effort that powers, for example, the RAST [2] effort. Manual curation of subsystems makes them an extremely valuable data source.

  Subsystems represent a four-level hierarchy:

  1. Subsystem level 1 – highest level
  2. Subsystem level 2 –
  3. Subsystem level 3 – similar to a KEGG pathway
  4. Subsystem level 4 – actual functional assignment to the feature in question

  The page at `http://pubseed.theseed.org/SubsysEditor.cgi` allows browsing the subsystems.

- KEGG Orthologs

  We use the KEGG [18] enzyme number hierarchy to implement a four-level hierarchy.

  1. KEGG level 1 – first digit of the EC number (EC:X.*.*.*)
  2. KEGG level 2 – first two digits of the EC number (EC:X.Y.*.*)
  3. KEGG level 3 – first three digits of the EC number (EC:X:Y:Z:.*)

4. KEGG level 4 – entire four digits EC number

We note that KEGG data is no longer available for free download. We thus have to rely on using the latest freely downloadable version of the data.

The high-level KEGG categories are as follows.

1. Cellular Processes
2. Environmental Information Processing
3. Genetic Information Processing
4. Human Diseases
5. Metabolism
6. Organizational Systems

- COG and EGGNOG Categories

  The high-level COG and EGGNOG categories are as follows.

  1. Cellular Processes
  2. Information Storage and Processing
  3. Metabolism
  4. Poorly Characterized

  We note that for most metagenomes the coverage of each of the four namespaces is quite different. The "source hits distribution" (see Section **??**) provides information on how many sequences per dataset were found for each database.

4.2 DRISEE profile

DRISEE [19] is a method for measuring sequencing error in whole-genome shotgun metagenomic sequence data that is independent of sequencing technology and overcomes many of the shortcomings of Phred. It utilizes artificial duplicate reads (ADRs) to generate internal sequence standards from which an overall assessment of sequencing error in a sample is derived. The current implementation of DRISEE is not suitable for amplicon sequencing data or other samples that may contain natural duplicated sequences (e.g., eukaryotic DNA where gene duplication and other forms of highly repetitive sequences are common) in high abundance. DRISEE results are presented on the Overview page for each MG-RAST sample for which a DRISEE profile can be determined. Total DRISEE error presents the overall DRISEE-based assessment of the sample as a percent error:

$$Total\ DRISEE\ Error = \frac{base\_errors}{total\_bases} * 100$$

where $base\_errors$ refers to the sum of DRISEE-detected errors and $total\_bases$ refers to the sum of all bases considered by DRISEE. Beneath the Total DRISEE Error, a barchart indicates the error for the sample (the red vertical bar) as well as the minimum (barchart initial value), maximum (barchart final value), mean (μ), mean +/- one standard deviation (σ), and mean +/- two standard deviations (2σ) Total DRISEE Errors observed among all samples in MG-RAST for which a DRISEE profile has been computed.

The DRISEE plot presents a more detailed view of the DRISEE profile; the DRISEE percent error is displayed per base. Individual errors (A,T,C,G, and N substitution rates as well as the InDel rate) are presented as well as a cumulative total.

Users can download DRISEE values as a tab-separated file. The first line of the file contains headers for the values in the second line. The second line contains DRISEE percent error values for A substitutions (A_err), T substitutions (T_err), C substitutions (C_err), G substitutions (G_err), N substitutions (N_err), insertions and deletions (InDel_err), and the Total DRISEE Error. The third line indicates headers for all remaining lines. Rows 4 and 4+ present the DRISEE counts for the indexed position across all considered bins of ADRs. Column values represent the number of reads that match an A,T,C,G,N, or InDel at the indicated position relative to the appropriate consensus sequence followed by the number of reads that do not match an A,T,C,G,N, or InDel.

4.3 Kmer profiles

kmer digests are an annotation-independent method for describing sequence datasets that can support inferences about genome size and coverage. Here the Overview page presents several visualizations, evaluated at k=15: the kmer spectrum, kmer rank abundance, and ranked kmer consumed. All three graphs represent the same spectrum, but in different ways. The kmer spectrum plots the number of distinct kmers against kmer coverage; the kmer coverage is equivalent to number of observations of each kmer. The kmer rank abundance plots the relationship between kmer coverage and the kmer rank—answering the question "What is the coverage of the nth most-abundant kmer?". Ranked kmer consumed plots the largest fraction of the data explained by the nth most-abundant kmers only.

4.4 Nucleotide histograms

Nucleotide histograms are graphs showing the fraction of base pairs of each type (A, C, G, T, or ambiguous base "N") at each position starting from the beginning of each read.

Amplicon datasets (see Figure 4.1) should show biased distributions of bases at each position, reflecting both conservation and variability in the recovered sequences:

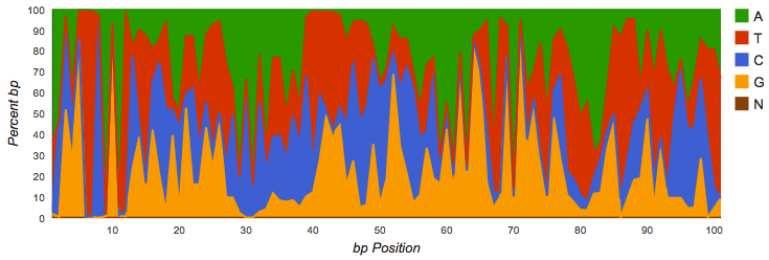


Figure 4.1: Nucleotide histogram with biased distributions typical for an amplicon dataset.

Shotgun datasets should have roughly equal proportions of A, T, G and C basecalls, independent of position in the read as shown in Figure 4.2.

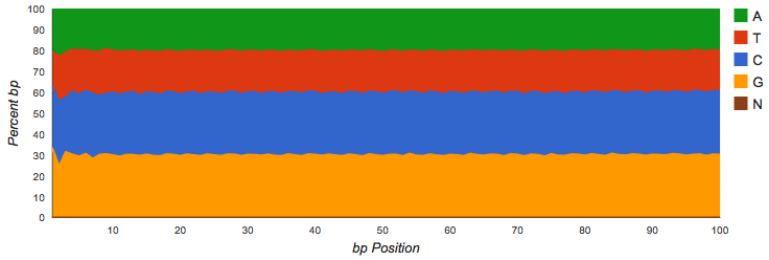


Figure 4.2: Nucleotide histogram showing ideal distributions typical for a shotgun metagenome.

Vertical bars at the beginning of the read indicate untrimmed (see Figure 4.3), contiguous barcodes. Gene calling via FragGeneScan [33] and RNA similarity searches are not impacted by the presence of barcodes. However, if a significant fraction of the reads is consumed by barcodes, it reduces the biological information contained in the reads.

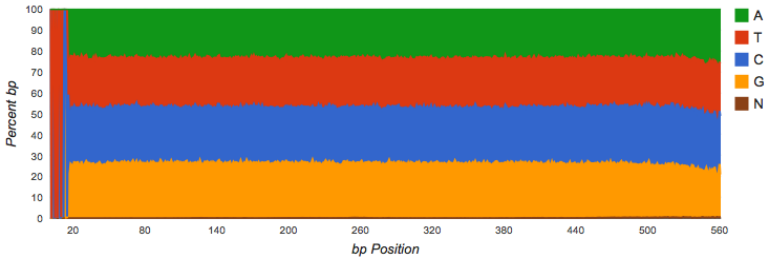


Figure 4.3: Nucleotide histogram with untrimmed barcodes.

If a shotgun dataset has clear patterns in the data, these indicate likely contamination with artificial sequences. The dataset shown in see Figure 4.4 had a large fraction of adapter dimers.

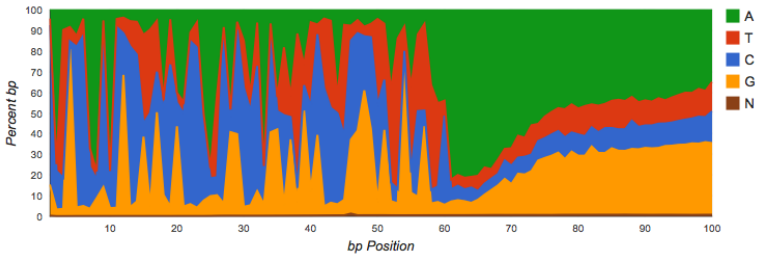


Figure 4.4: Nucleotide histogram with contamination.

4.5 Best hit, representative hit, and lowest common ancestor profiles

MG-RAST searches the nonredundant M5nr and M5rna databases in which each sequence is unique. These two databases are built from multiple sequence database sources, and the individual sequences may occur multiple times in different strains and species (and sometimes genera) with 100% identity. In these circumstances, choosing the "right" taxonomic information is not a straightforward process.

To optimally serve a number of different use cases, we have implemented three methods–best hit, representative hit, and lowest common ancestor—for end users to determine the number of hits (occurrences of the input sequence in the database) reported for a given sequence in their dataset.

Best hit

The best hit classification reports the functional and taxonomic annotation of the best hit in the M5nr for each feature. In those cases where the similarity search yields multiple same-scoring hits for a feature, we do not choose any single "correct" label. For this reason we have decided to double count all annotations with identical match properties and leave determination of truth to our users. While this approach aims to inform about the functional and taxonomic potential of a microbial community by preserving all information, subsequent analysis can be biased because of a single feature having multiple annotations, leading to inflated hit counts. For users looking for a specific species or function in their results, the best hit classification is likely what is wanted.

Representative hit

The representative hit classification selects a single, unambiguous annotation for each feature. The annotation is based on the first hit in the homology search and the first annotation for that hit in our database. This approach makes counts additive across functional and taxonomic levels

and thus allows, for example, the comparison of functional and taxonomic profiles of different metagenomes.

Lowest Common Ancestor (LCA)

To avoid the problem of multiple taxonomic annotations for a single feature, we provide taxonomic annotations based on the widely used LCA method introduced by MEGAN [15]. In this method all hits are collected that have a bit score close to the bit score of the best hit. The taxonomic annotation of the feature is then determined by computing the LCA of all species in this set. This replaces all taxonomic annotations from ambiguous hits with a single higher-level annotation in the NCBI taxonomy tree.

Comparison of methods

Users should be aware that the number of hits might be inflated if the best hit filter is used or that a favorite species might be missing despite a similar sequence similarity result if the representative hit filter is used (in fact, even if a 100% identical match to a favorite species exists).

One way to consider both the best hit and representative hit is that they overinterpret the available evidence. With the LCA classifier function, on the other hand, any input sequence is classified only down to a trustworthy taxonomic level. While naively this seems to be the best function to choose in all cases because it classifies sequences to varying depths, the approach causes problems for downstream analysis tools that might rely on everything being classified to the same level.

4.6 Numbers of annotations vs. number of reads

The MG-RAST v3 annotation pipeline does not usually provide a single annotation for each submitted fragment of DNA. Steps in the pipeline map one read to multiple annotations and one annotation to multiple reads. These steps are a consequence of genome structure, pipeline engineering, and the character of the sequence databases that MG-RAST uses for annotation.

The first step that is not one-to-one is gene prediction. Long reads (> 400bp) and contigs can contain pieces of two or more microbial genes; when the gene caller makes this prediction, the multiple predicted protein sequences (called fragments) are annotated separately.

An intermediate clustering step identifies sequences at 90% amino acid identity and performs one search for each cluster. Sequences that do not fall into clusters are searched separately. The "abundance" column in the MG-RAST tables presents the estimate of the number of sequences that contain a given annotation, found by multiplying each selected database match (hit) by the number of representatives in each cluster. The final step that is not one-to-one is the annotation process itself. Sequences can exist in the underlying data sources many times with different labels. When those sequence are the best hit similarity, we do not have a principled way to choosing the "correct"

label. For this reason we have decided to double count these annotations and leave determination of truth to our users. Note: Even when considering a single data source, double-counting can occur depending on the consistency of annotations. Also note: Hits refer to the number of unique database sequences that were found in the similarity search, **not** the number of reads. The hit count can be smaller than the number of reads because of clustering or larger due to double counting.

4.7 Metadata

MG-RAST is both an analytical platform and a data integration system. To enable data reuse, for example for meta-analyses, we require that all data being made publicly available to third parties contain at least minimal metadata. The MG-RAST team has decided to follow the minimal checklist approach used by the Genomics Standards Consortium (GSC) [11].

While the GSC provides a GCDML [31] encoding, this XML-based format is more useful to programmers than to end users submitting data. We have therefore elected to use spreadsheets to transport metadata. Specifically we use MIxS (Minimum information about any (x) sequence (MIxS) and MIMARKS (Minimum Information about a MARKer gene Survey) to encode minimal metadata [45].

The metadata describe the origins of samples and provide details on the generation of the sequence data. While the GSC checklist aims at capturing a minimum of information, MG-RAST can handle additional metadata if supplied by the user. The metadata is stored in a simple key value format and is displayed on the Metagenome Overview page.

Once uploaded, the metadata spreadsheets are validated automatically, and users are informed of any problems.

The presence of metadata enables discovery by end users using contextual metadata. Users can perform searches such as "retrieve soil samples from the continental U.S.A." If the users have added additional metadata (domain specific extension), additional queries are enabled: for example, "restrict the results to soils with a specific pH".

Chapter 5

The version 4.0 web interface

The MG-RAST system provides a rich web user interface that covers all aspects of the metagenome analysis, from data upload to ordination analysis. The web interface can also be used for data discovery. **Metadata** enables data discovery MG-RAST supports the widely used MIxS and MIMARKS Standards (Yilmaz, 2011) (as well as domain-specific plug-ins for specialized environments extending the minimal GSC standards).

One key aspect of the MG-RAST approach is the creation of **smart data products** enabling the user at the time of analysis to determine the best parameters for, for example, a comparison between samples. This is done without the need for recomputation of results.

We note that if you want to create links to the MG-RAST web site, you should use the *linkin* mechanism instead of linking to any web page directly. All pages intended for users to create external links provide the linkin feature, See Section 8.1.

Starting with version 4.0 MG-RAST supports an HTML5/JavaScript based web user interface. It supports any recent browser but is tested primarily with Firefox.

Below we are iterating through the various pages in MG-RAST version 4.
Figure 5.1 shows a sample analysis with MG-RAST.

5.1 The "My Data" page

After login the user is directed to their personal "My Data" page (see figure 5.2), their personal MG-RAST homepage.

This page is provides information on data sets currently being processed, data sets owned by the user as well as any upcoming tasks for the users (i.e. release data to the public after the expiration

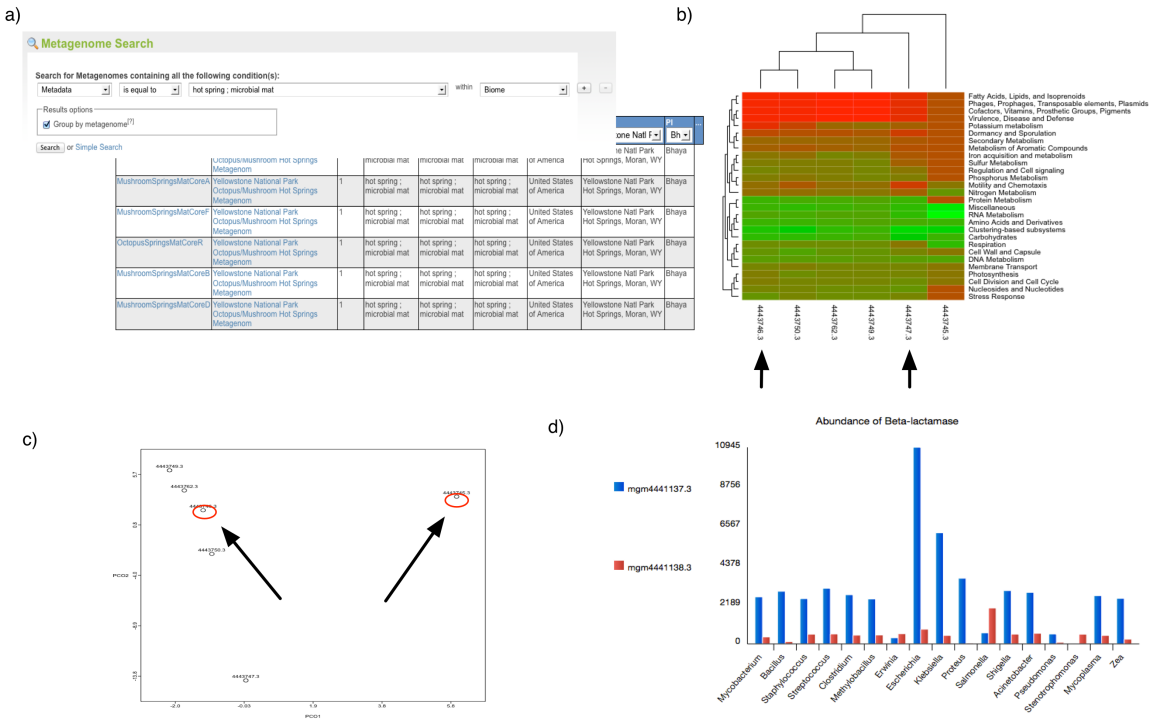


Figure 5.1: (a) Using the web interface for a search of metagenomes for microbial mats in hotsprings (GSC-MIMS-Keywords Biome="hotspring; microbial mat"), we find 6 metagenomes (refs: 4443745.3, 4443746.3, 4443747.3, 4443749.3, 4443750.3, 4443762.3). (b) Initial comparison reveals some differences in protein functional class abundance (using SEED subsystens level 1). (c) From the PCoA plot using normalized counts of functional SEED Subsystem–based functional annotations (level 2) and Bray-Curtis as metric, we attempt to find differences between two similar datasets (MG-RAST-IDs: 444749.3, 4443762.3). (d) Using exported tables with functional annotations and taxonomic mapping, we analyze the distribution of organisms observed to contain beta-lactamase and plot the abundance per species for two distinct samples.

of the quarantine period).

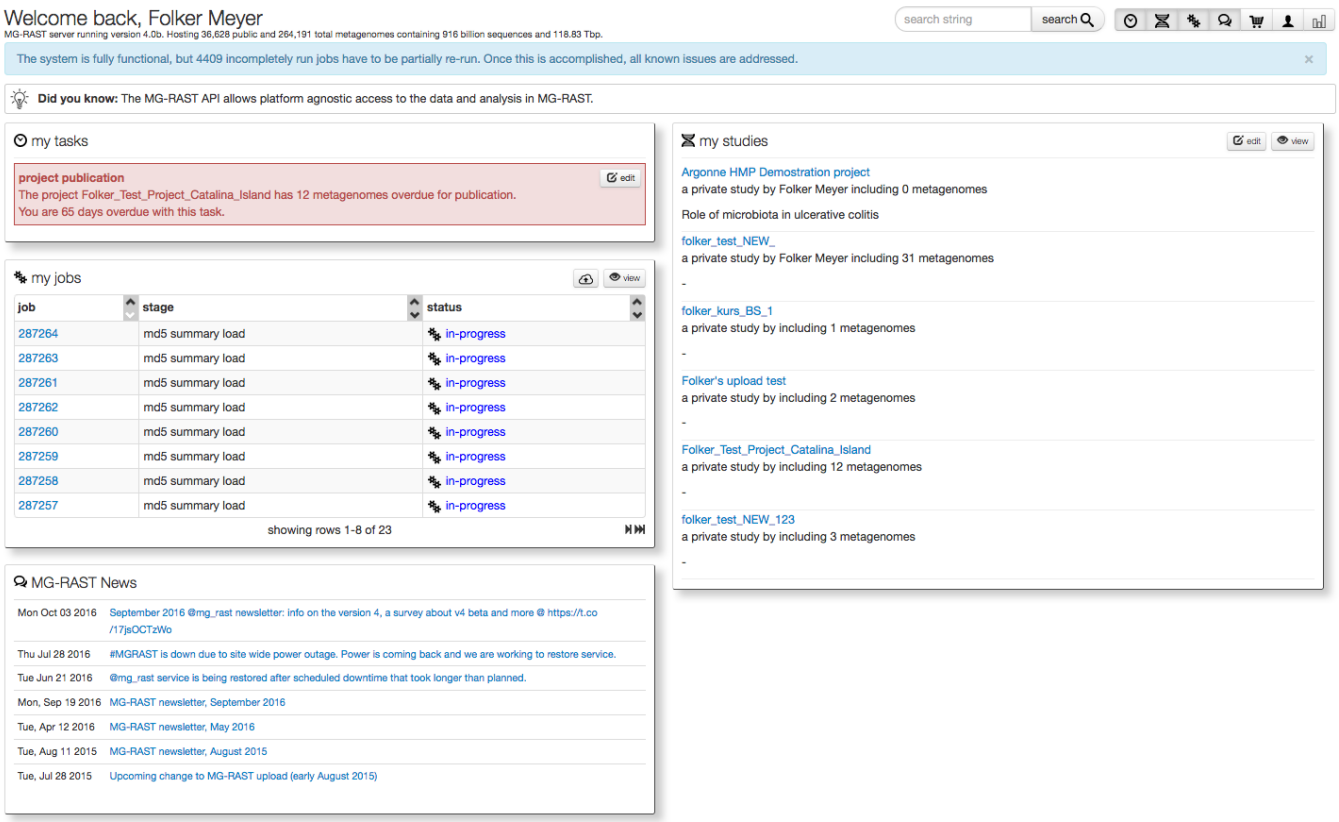


Figure 5.2: The page shows currently running jobs, the tasks the user needs to perform in the system, a list of their studies and more.

In addition to the data items mentioned above, the page also contains a list of the collections (see **??**) owned by the user.

5.2 Browsing, searching and viewing studies

5.2.1 The search page

The search page lists all available metagenomic data sets and allows filtering. The looking glass symbol provides access to the search page, there are also shortcuts to the search function on multiple pages.

The basic function of the Search page is to find data sets that (1) contain a search string in the metadata (dataset name, project name, project description, GSC metadata), (2) contain specific

functions (e.g., SEED functional roles, SEED subsystems, or GenBank annotations), or (3) contain specific organisms. The default search uses all three kinds of data.

In addition to a Google-like search that searches all data fields, we provide specialized searches in one of the three data types.

We note that due to data visibility (see 8.1) not all data sets are visible to all users.

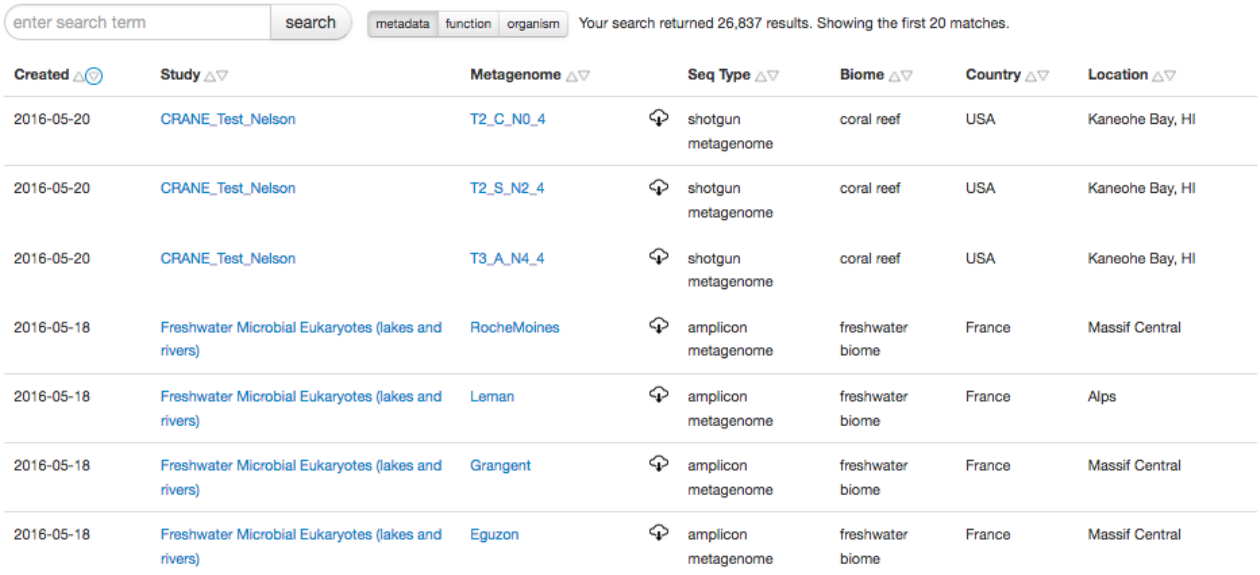


Figure 5.3: The search page.

The search page has two components, the output widget (see figure 5.3) and the refinement widget.

The refinement widget allows filtering, the creation of saved searches and the creation of collections.

5.2.2 The study page

Data in MG-RAST is organized in studies (formerly known as Projects), each study has an automatically generated page.

The study page displays a project title, project description and other study specific information such as funding information. Users are encouraged to provide information on the project in addition to the metadata. The study page also includes the ability to display analysis results generated with the MG-RAST user interface.

The study page provides a number of tools to the data set owner:

- Sharing
  Studies in MG-RAST while initially private (see 8.1) can be shared with others. Simply

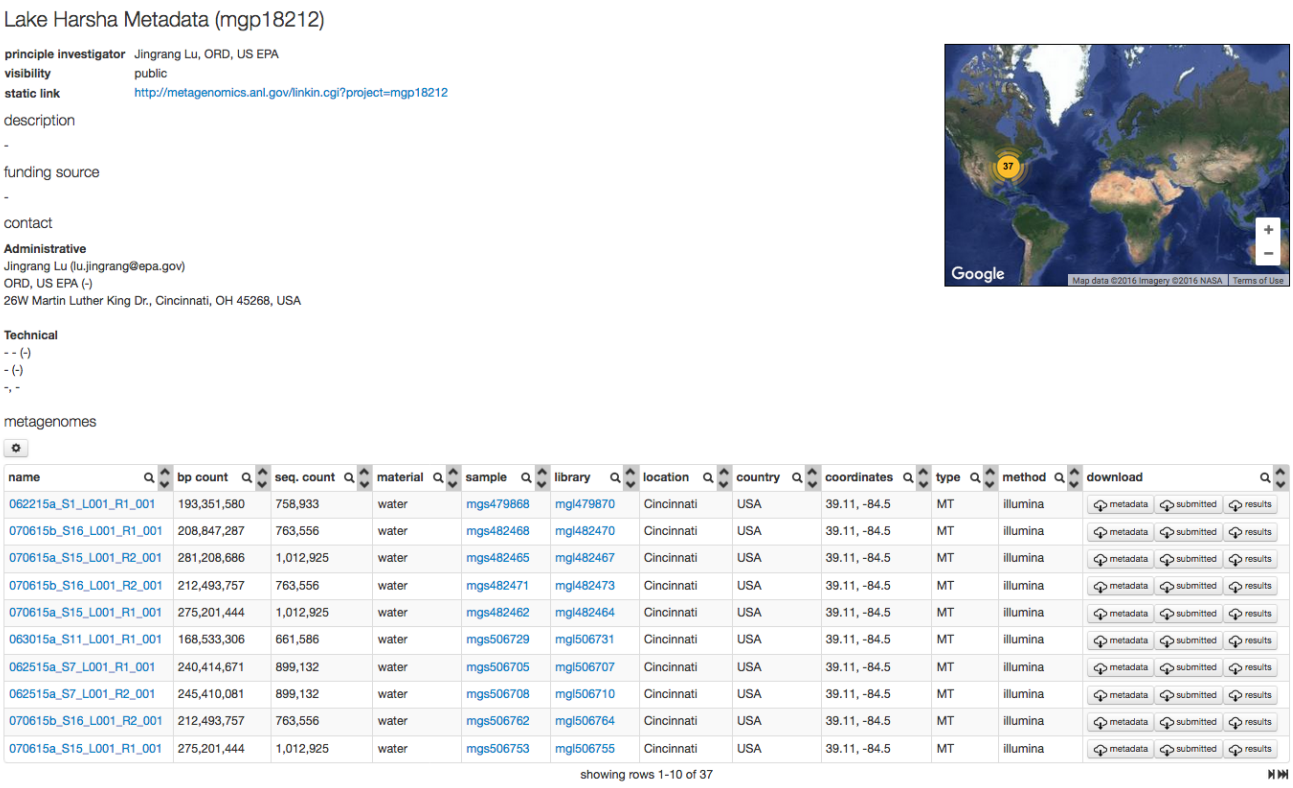


Figure 5.4: A study page.

provide any email address for an individual and they will be send a token that allows data access. Sharing is intended to allow pre-publication data sharing.

- Reviewer access
  Reviewer access tokens can be embedded in Manuscripts (or their cover letters) to allow reviewers and editors access to the data sets.

- Data Publication
  Data can be made public. This option will generate the only kind of identifiers that should be used in publications.

- Metadata editor
  Complete or correct the metadata.

5.3 Information about specific data sets (Overview page)

MG-RAST automatically creates an individual summary page for each dataset. This metagenome overview page provides a summary of the annotations for a single dataset. The page is made available by the automated pipeline once the computation is finished. The page is generated using default values for annotation transfer parameters (e.g. e-values) and thus likely does not represent good biological information, for that please use the Analysis page (see below).

However the Overview page is a good starting point for looking at a particular dataset. It provides a significant amount of information on technical details and biological content.

The page is intended as a single point of reference for metadata, quality, and data. It also provides an initial overview of the analysis results for individual datasets with default parameters. Further analyses are available on the Analysis page.

There are four different types of Overview pages that are displayed as required by the data:

- Amplicon metagenome overview page
- Shotgun metagenome overview page
- Assembled shotgun metagenome overview page
- Metatranscriptome overview page

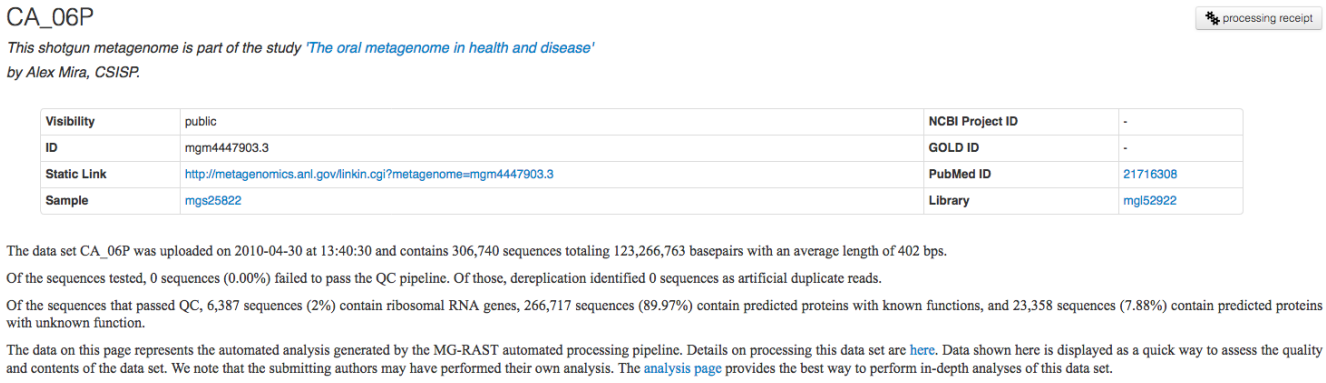
While the different types of overview pages are mostly identical, some visualizations are not relevant or even possible for certain data types. The decision which type of page to display is made based on the data, not the metadata provided by the user.

Previous version of MG-RAST provided almost complete **download** access to the underlying data, with version 4.0 we have expanded that to all tables and figures. The symbol shown in Figure **??**.

The Overview page provides the MG-RAST ID for a data set, a unique identifier that is usable as accession number for publications. Additional information such as the name of the submitting PI and organization and a user-provided metagenome name are displayed at the top of the page as well. A static URL for linking to the system that will be stable across changes to the MG-RAST web interface is provided as additional information (Figure 5.6).

We point the readers attention to the download symbols next to each figure and or table, providing access to the data and API calls underlying each display item.

We provide an automatically generated paragraph of text describing the submitted data and the results computed by the pipeline. By means of the project information we display additional information provided by the data submitters at the time of submission or later.


CA_06P

processing receipt

This shotgun metagenome is part of the study 'The oral metagenome in health and disease'
by Alex Mira, CSISP.

Visibility	public	**NCBI Project ID**	-
ID	mgm4447903.3	**GOLD ID**	-
Static Link	http://metagenomics.anl.gov/linkin.cgi?metagenome=mgm4447903.3	**PubMed ID**	21716308
Sample	mgs25822	**Library**	mgl52922

The data set CA_06P was uploaded on 2010-04-30 at 13:40:30 and contains 306,740 sequences totaling 123,266,763 basepairs with an average length of 402 bps.

Of the sequences tested, 0 sequences (0.00%) failed to pass the QC pipeline. Of those, dereplication identified 0 sequences as artificial duplicate reads.

Of the sequences that passed QC, 6,387 sequences (2%) contain ribosomal RNA genes, 266,717 sequences (89.97%) contain predicted proteins with known functions, and 23,358 sequences (7.88%) contain predicted proteins with unknown function.

The data on this page represents the automated analysis generated by the MG-RAST automated processing pipeline. Details on processing this data set are here. Data shown here is displayed as a quick way to assess the quality and contents of the data set. We note that the submitting authors may have performed their own analysis. The analysis page provides the best way to perform in-depth analyses of this data set.

Figure 5.6: Top of the metagenome Overview page.

Sequence and feature breakdown

One of the first places to look at for each data set are the function and feature breakdown at the top of each overview page.

The pie charts at the top of the overview page (Figure 5.7) classify the submitted sequences submitted into several categories according to their QC results, sequences are classified as having failed QC (grey), containing at least one feature (purple) and unknown if they do not contain any recognized feature (red). In addition the predicted features are broken up into unknown protein (yellow), annotated protein (green) and ribosomal RNA (blue) in a second pie chart.

5.3.0.1 What about other feature types?

We note that for performance reasons no other sequence features are annotated by the default pipeline. Other feature types such as small RNAs or regulatory motifs (e.g., CRISPRs [5]) not only will require significantly higher computational resources but also are frequently not supported by the unassembled short reads that constitute the vast majority of todays metagenomic data in MG-RAST. The quality of the sequence data coming from next-generation instruments requires careful design of experiments, lest the sensitivity of the methods is greater than the signal-to-noise ratio the data supports.

*

Metadata

The overview page also provides metadata for each dataset to the extent that such information has been made available. Metadata enables other researchers to discover datasets and compare annotations. MG-RAST requires standard metadata for data sharing and data publication. This is

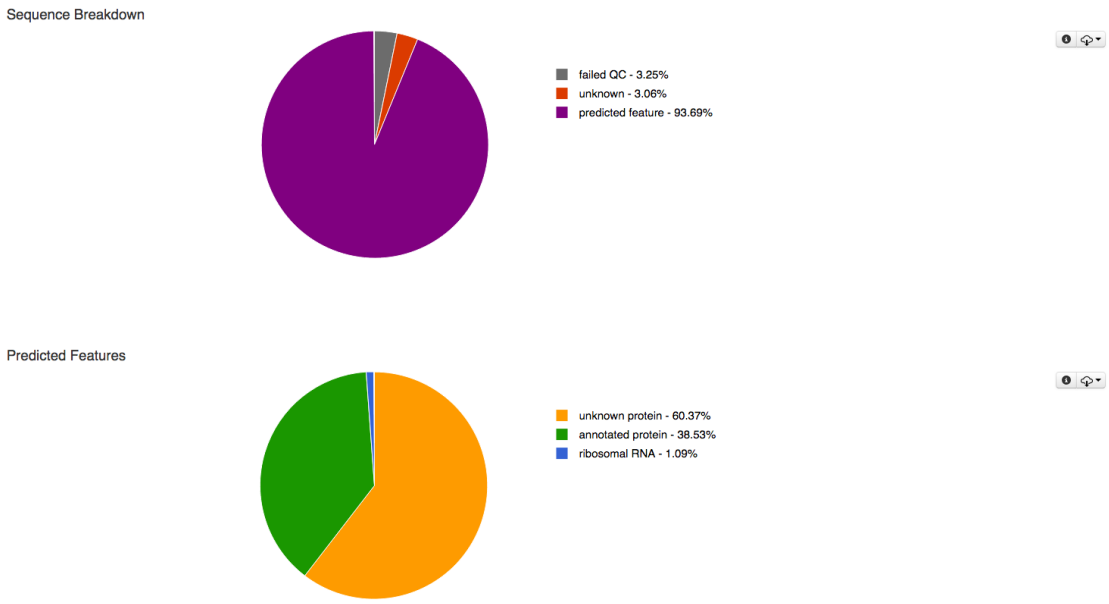


Figure 5.7: The first pie charts classifies the sequences submitted in this data set according to their QC results, the 2nd breaks down the detected features in to several categories.

implemented using the standards developed by the Genomics Standards Consortium. Figure 5.8 shows the metadata summary for a dataset.

All metadata stored for a specific dataset is available in MG-RAST; we merely display a standardized subset in this table. A link at the bottom of the table ("More Metadata") provides access to a table with the complete metadata. This enables users to provide extended metadata going beyond the GSC minimal standards. A mechanism to provide community consensus extensions to the minimal checklists and the environmental packages are explicitly encouraged but not required when using MG-RAST.

Functional and taxonomic breakdowns

A number of pie charts are computed, represening a breakdown of the data into different taxonomic ranks (domain, phylum, class, order, family, genus) an the top levels of the four supported controlled annotation namespaces (Subsystems, Kegg Orthologues (KOGS), COGs and Eggnogs (NOGS)).

Rank abundance

The rank abundance plot (Figure 5.9) provides a rank-ordered list of taxonomic units at a user-defined taxonomic level, ordered by their abundance in the annotations.

GSC MIxS INFO

Investigation Type	**metagenome**
Project Name	**The oral metagenome in health and disease**
Latitude and Longitude	**39.481448, 0.353066**
Country and/or Sea, Location	**Spain Valencia**
Collection Date	**2010-03-01 10:00:00 UTC**
Environment (Biome)	**human-associated habitat**
Environment (Feature)	**human-associated habitat**
Environment (Material)	**human-associated habitat**
Environmental Package	**human-oral**
Sequencing Method	**454**
More Metadata	

Figure 5.8: Information from the GSC MIxS checklist providing minimal metadata on the sample.

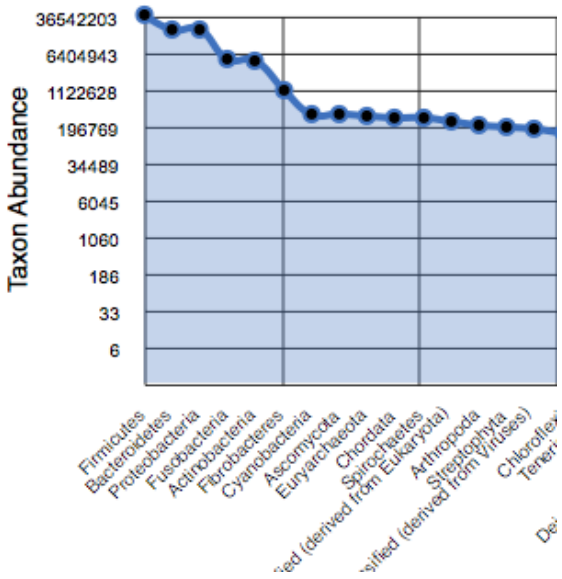


Figure 5.9: Sample rank abundance plot by phylum.

Rarefaction

The rarefaction curve of annotated species richness is a plot (see Figure 5.10 of the total number of distinct species annotations as a function of the number of sequences sampled. The slope of the right-hand part of the curve is related to the fraction of sampled species that are rare. On the left, a steep slope indicates that a large fraction of the species diversity remains to be discovered. If the curve becomes flatter to the right, a reasonable number of individuals is sampled: more intensive sampling is likely to yield only few additional species. Sampling curves generally rise quickly at first and then level off toward an asymptote as fewer new species are found per unit of individuals collected.

The rarefaction curve is derived from the protein taxonomic annotations and is subject to problems stemming from technical artifacts. These artifacts can be similar to the ones affecting amplicon sequencing [32], but the process of inferring species from protein similarities may introduce additional uncertainty.

Alpha diversity

In this section we display an estimate of the alpha diversity based on the taxonomic annotations for the predicted proteins. The alpha diversity is presented in context of other metagenomes in the same project (see Figure 5.11).

The alpha diversity estimate is a single number that summarizes the distribution of species-level annotations in a dataset. The Shannon diversity index is an abundance-weighted average of

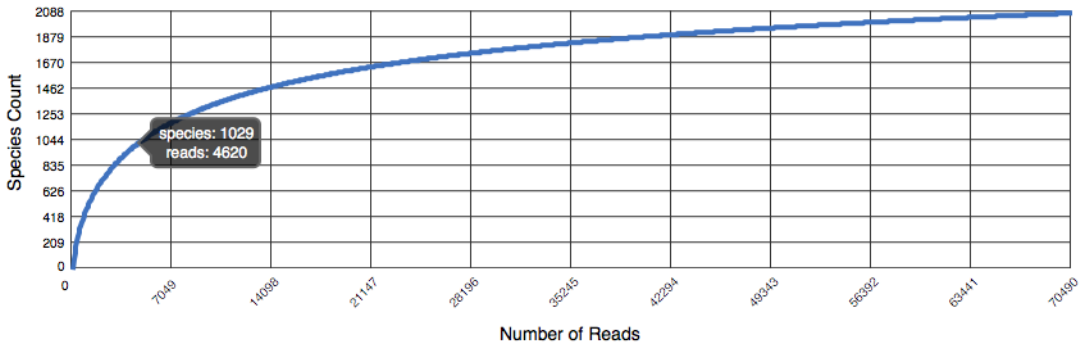


Figure 5.10: Rarefaction plot showing a curve of annotated species richness. This curve is a plot of the total number of distinct species annotations as a function of the number of sequences sampled.

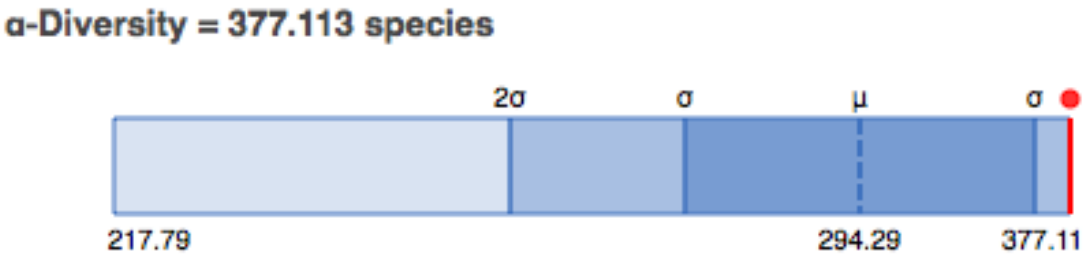


Figure 5.11: Alpha diversity plot showing the range of α-diversity values in the project the data set belongs to. The min, max, and mean values are shown, with the standard deviation ranges (σ and 2σ) in different shades. The α-diversity of this metagenome is shown in red.

the logarithm of the relative abundances of annotated species.

We compute the species richness as the antilog of the Shannon diversity:

$$\text{Richness} = 10^{-\sum_i p_i \log(p_i)}$$

where p_i are the proportions of annotations in each of the species categories. Shannon species richness has units of the "effective number of species". Each p is a ratio of the number of annotations for each species to the total number of annotations. The species-level annotations are from all the annotation source databases used by MG-RAST. The table of species and number of observations used to calculate this diversity estimate can be downloaded under "download source data" on the Overview page.

Functional categories

This section contains four pie charts providing a breakdown of the functional categories for KEGG [18], COG [37], SEED Subsystems [28], and eggNOGs [17]. The relative abundance of sequences per functional category can be downloaded as a spreadsheet, and users can browse the functional breakdowns via the Krona tool [27] integrated in the page.

A more detailed functional analysis, allowing the user to manipulate parameters for sequence similarity matches, is available from the Analysis page.

The sample page

For each sample MG-RAST displays a sample page shown in figure 5.13, the page displays all sample specific information. The information on this page is derived from the metadata.

The library page

For each set of sequences underlying a data set ("a library") MG-RAST provides a specific page with information extracted from the metadata.

5.4 The analysis page – Comparing data, extracting and downloading data

The Analysis page is the core of the MG-RAST system, it consumes the various profiles and allows adjusting of parameters.

. It provides a number of tools to compare data sets with different parameters as well as the ability to drill down into the data (e.g. selecting Actinobacteria or features related to a specific functional gene group (e.g. the Lysine Biosynthesis Subsystem).

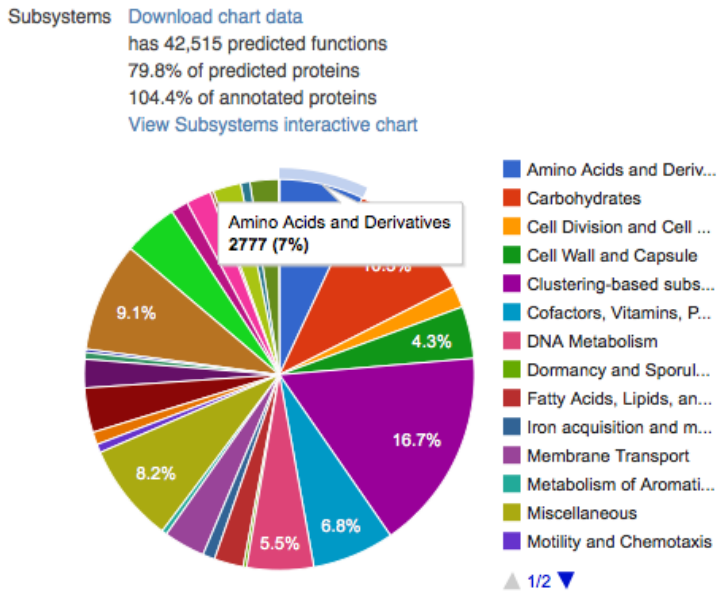


Figure 5.12: The Subsystems function piechart, showing reads classified into SEED subsystem level-one functions. In contrast to the COG, eggNOG, and KEGG classification schemes, there are over 20 top-level subsystem categories, creating a more highly resolved "fingerprint" for the metagenome.

sample mgs25824

study

altitude	13
biome	human-associated habitat
collection date	2010-03-01
collection time	10:00:00
collection timezone	UTC
continent	europe
country	Spain
depth	0
elevation	13
env package	human-oral
feature	human-associated habitat
isol growth condt	21716308
latitude	39.481448
location	Valencia
longitude	0.353066
material	human-associated habitat
mgrast id	mgs25824
misc param 1	geodetic_system: wgs_84
samp mat process	total DNA extraction - enzymes
samp size	1
sample name	mgs25824
sample strategy	human-associated habitat
temperature	37
age	42 ; Year
body product	Supragingival dental plaque from inside the cavity
body site	Cavity on tooth 4.6
disease stat	Diseased, CAO's index = 10
host subject id	CA_05
mgrast id	mge52927
misc param 1	env_package: human-oral
nose mouth teeth throat disord	Teeth
sex	Male
time last toothbrush	24 hours
created	2011-07-06 17:12:33

Figure 5.13: A sample page.

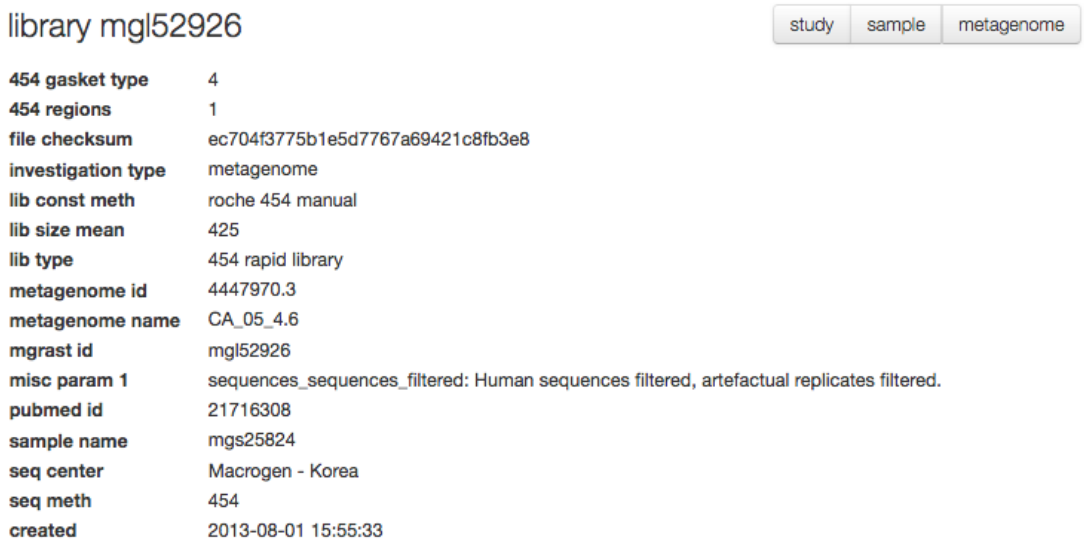


Figure 5.14: A library page.

Compared to previous version of MG-RAST the Analysis page has seen significant improvements, here we provide a step-by-step guide to using the page

Download profiles to local machine for analysis

Profiles to be compared, analyzed or visualized need to be downloaded. Figure 5.15 shows an example download of 8 profiles.

After the profiles have been downloaded, the analysis is no longer dependent on the MG-RAST server resources, instead using the computer the browser is running on. This is achieved via the JavaScript functionality in your browser (please make sure its enabled). Also data is stored in memory, providing you with a good reason to maximize the memory (RAM) of the machine you are running the analysis on.

Normalization

Normalization refers to a transformation that attempts to reshape an underlying distribution. MG-RAST now uses DEseq, which is an R package to analyse count data from high-throughput sequencing assays. DESeq, as it has been shown to outperform other methods of normalization - in particular, those that use any sort of linear scaling.

Standardization is a transformation applied to each distribution in a group of distributions so that all distributions exhibit the same mean and the same standard deviation. This removes some

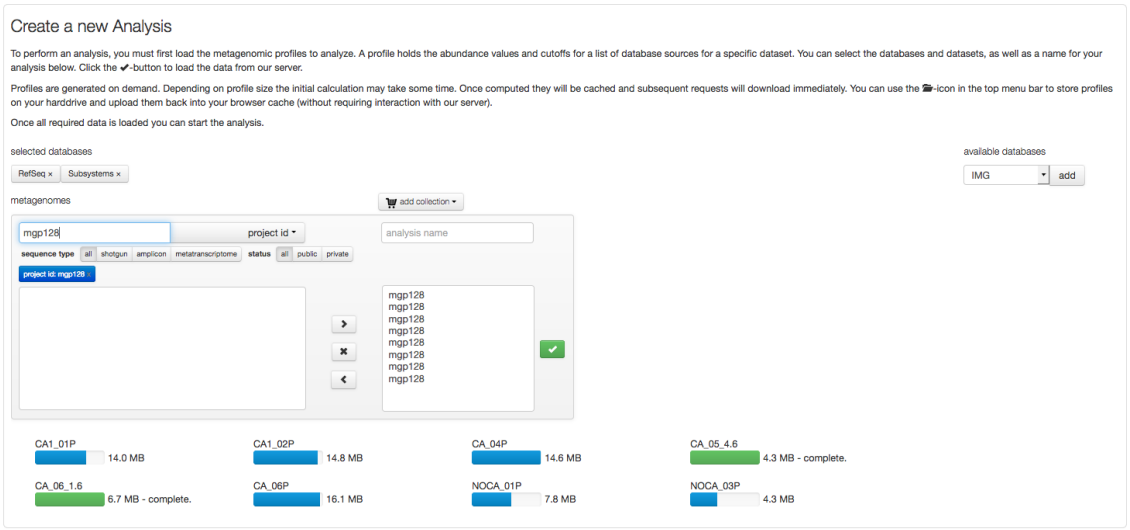


Figure 5.15: After selecting a project ("mgp128") the Refseq and Subsystem profiles for the respective data sets are loaded. Blue progress bars indicated profiles being uploaded, green bars indicate the download has completed.

aspects of intersample variability and can make data more comparable. This sort of procedure is analogous to commonly practiced scaling procedures but is more robust in that it controls for both scale and location.

The Analysis page calculates the ordination visualizations with either raw or normalized counts, at the user's option. The normalization procedure is as follows.

$$normalized\_value\_i = log2(raw\_counts\_i + 1)$$

The standardized values then are calculated from the normalized values by subtracting the mean of each sample's normalized values and dividing by the standard deviation of each sample's normalized values.

$$standardized\_i = (normalized\_i - mean(normalized\_i))/stddev(normalized\_i)$$

More about these procedures is available in a number of texts. We recommend Terry Speed's "Statistical Analysis of Gene Expression in Microarray Data" [36].

When data exhibit a nonnormal, normal, or unknown distribution, nonparametric tests (e.g., Man-Whitney or Kurskal-Wallis) should be used. Boxplots are easy to use, and the MG-RAST analysis page provides boxplots of the standardized abundance values for checking the comparability of samples (Figure 5.16).

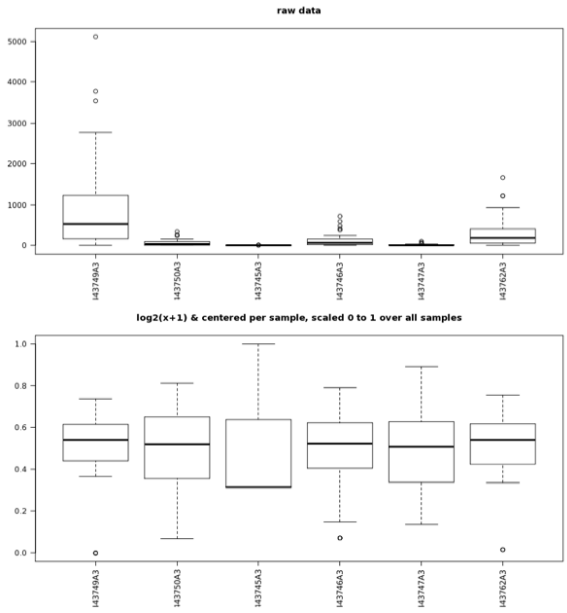


Figure 5.16: Boxplots of the abundance data for raw values (top) as well as values that have undergone the normalization and standardization procedures (bottom) described in the text. After normalization and standardization, samples exhibit value distributions that are much more comparable and that have a normal distribution; the normalized and standardized data are suitable for analysis with parametric tests; the raw data are not.

Rarefaction

The rarefaction view is available only for taxonomic data. The rarefaction curve of annotated species richness is a plot (see Figure 5.17) of the total number of distinct species annotations as a function of the number of sequences sampled. As shown in Figure 5.17, multiple data sets can be included.

The slope of the right-hand part of the curve is related to the fraction of sampled species that are rare. When the rarefaction curve is flat, more intensive sampling is likely to yield only a few additional species. The rarefaction curve is derived from the protein taxonomic annotations and is subject to problems stemming from technical artifacts. These artifacts can be similar to the ones affecting amplicon sequencing [32], but the process of inferring species from protein similarities may introduce additional uncertainty.

On the Analysis page the rarefaction plot serves as a means of comparing species richness between samples in a way independent of the sampling depth.

On the left, a steep slope indicates that a large fraction of the species diversity remains to be discovered. If the curve becomes flatter to the right, a reasonable number of individuals is sampled: more intensive sampling is likely to yield only a few additional species.

Sampling curves generally rise very quickly at first and then level off toward an asymptote as fewer new species are found per unit of individuals collected. These rarefaction curves are calculated from the table of species abundance. The curves represent the average number of different species annotations for subsamples of the the complete dataset.

KEGG mapper

The KEGG map tool allows the visual comparison of predicted metabolic pathways in metagenomic samples. It maps the abundance of identified enzymes onto a KEGG [18] map of functional pathways; note that the mapper is available only for functional data). Users can select from any available KEGG pathway map. Different colors indicate different metagenomic datasets.

The KEGG mapper works by providing two buffers that users can assign datasets to. After loading the buffers with the intended datasets, the KEGG mapper can highlight parts of the KEGG map that are present in the dataset. Several combinations of the two datasets can be displayed, as shown in Figure 5.18. Metagenomes can be assigned into one of two groups, and those groups can be visually compared (see Figure 5.19).

Bar charts

Figure 5.20 shows the bar chart visualization option on the Analysis page. One important property of the page is the built-in ability to drill down by clicking on a specific category. In this example we have expanded the domain Bacteria to show the normalized abundance (adjusted for sample

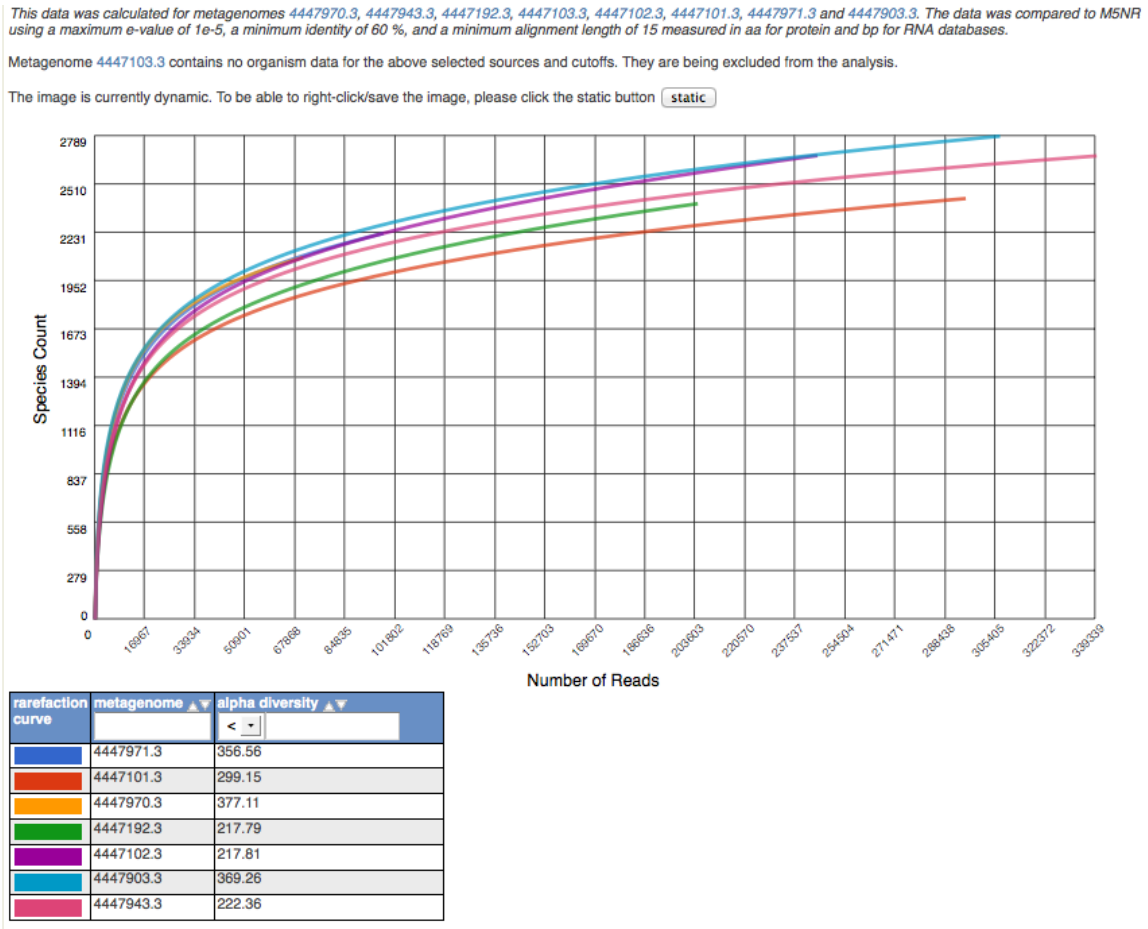


Figure 5.17: Rarefaction plot showing a curve of annotated species richness. This curve is a plot of the total number of distinct species annotations as a function of the number of sequences sampled.

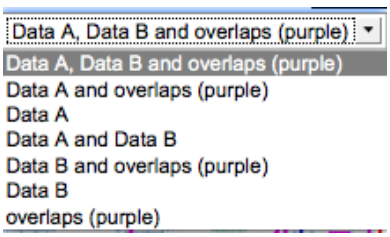


Figure 5.18: Options available for coloring the KEGG maps.

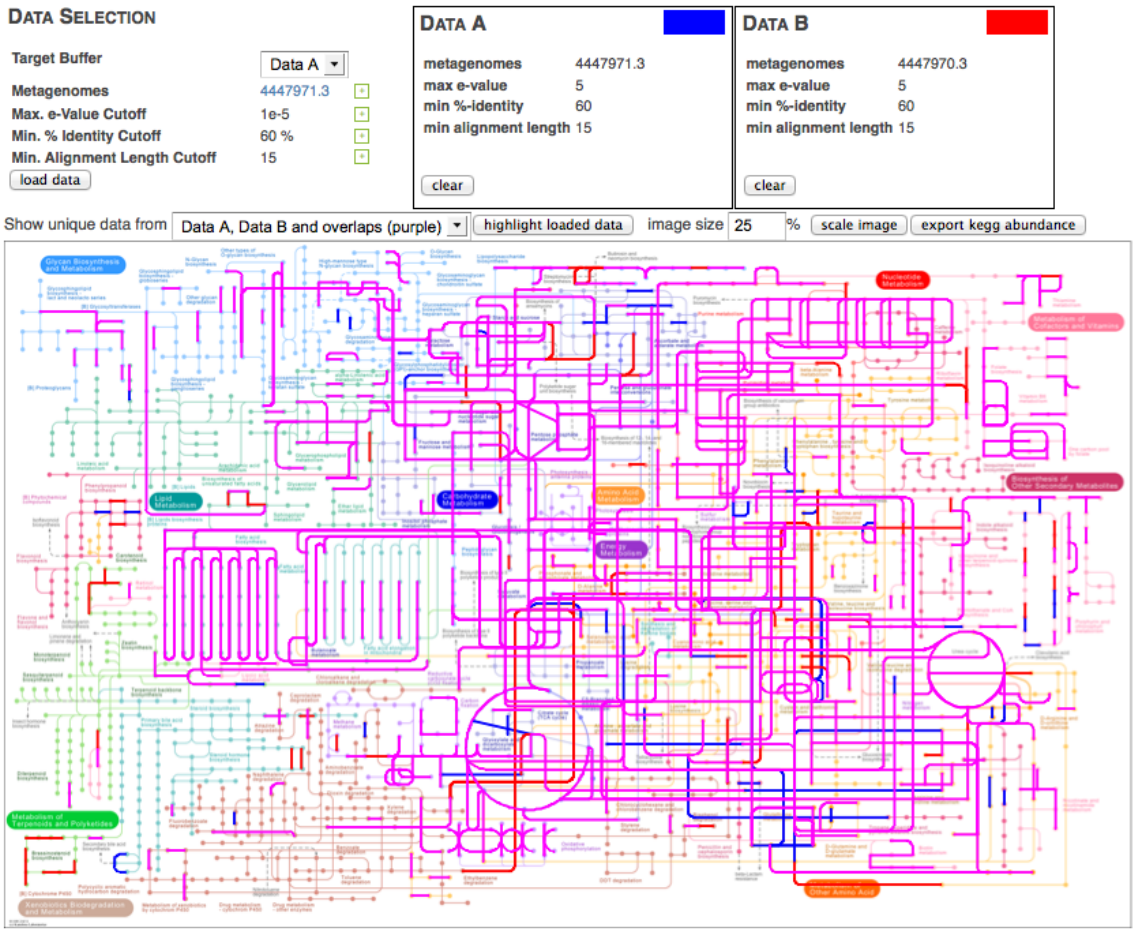


Figure 5.19: Comparison of two datasets using the KEGG mapper. Parts of metabolism common are shown in purple; unique to A are in blue; unique to B are in red.

sizes) of bacterial phyla. The abundance information displayed can be downloaded into a local spreadsheet. Once a subselection has been made (e.g., the domain Bacteria selected).

Heatmap/Dendrogram

The heatmap/dendrogram (Figure 5.21) allows an enormous amount of information to be presented in a visual form that is amenable to human interpretation. Dendrograms are trees that indicate similarities between annotation vectors. The MG-RAST heatmap/dendrogram has two dendrograms, one indicating the similarity/dissimilarity among metagenomic samples (x-axis dendrogram) and another indicating the similarity/dissimilarity among annotation categories (e.g., functional roles; the y-axis dendrogram). A distance metric is evaluated between every possible pair of sample abundance profiles. A clustering algorithm (e.g., ward-based clustering) then produces the den-

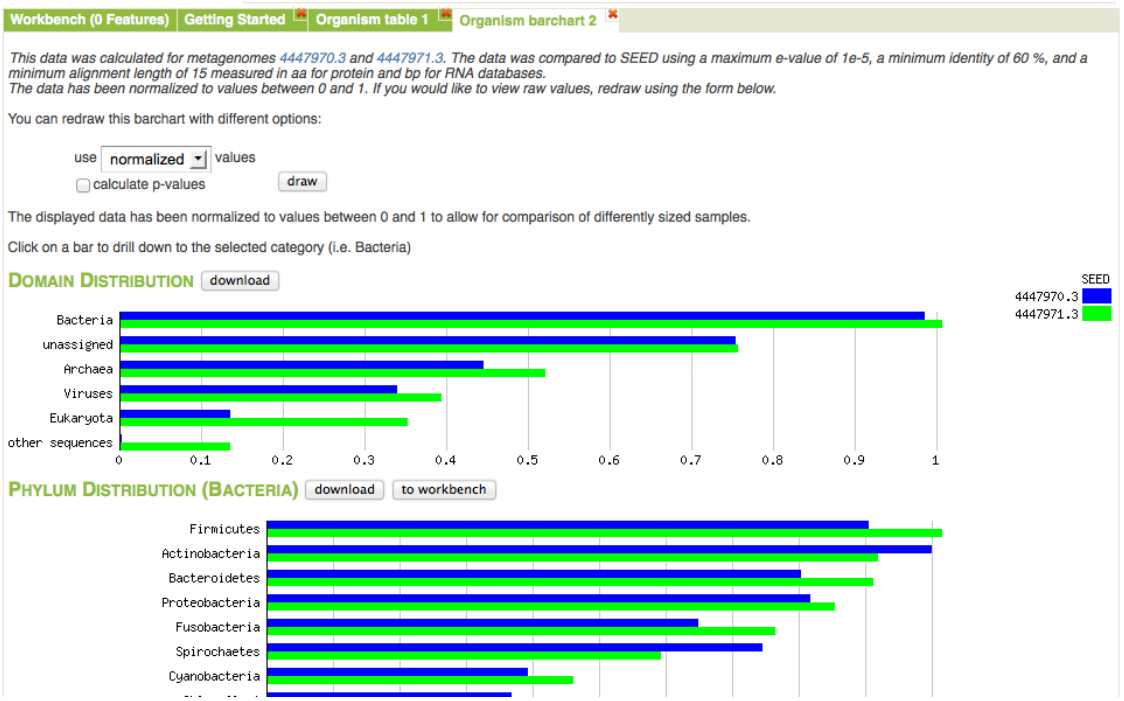


Figure 5.20: Bar chart view comparing normalized abundance of taxa. We have expanded the Bacteria domain to display the next level of the hierarchy.

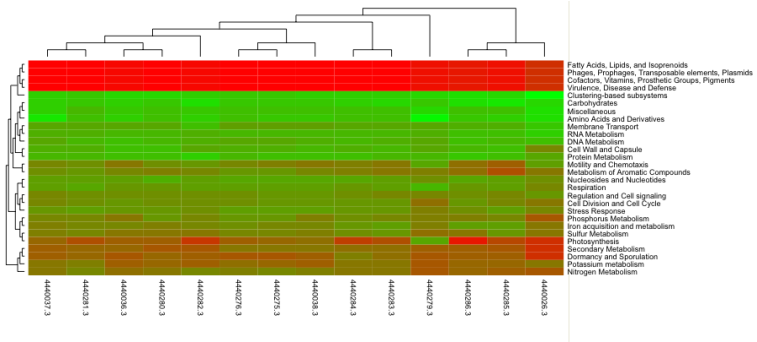


Figure 5.21: Heatmap/dendogram example in MG-RAST. The MG-RAST heatmap/dendrogram has two dendrograms, one indicating the similarity/dissimilarity among metagenomic samples (x axis dendrogram) and another indicating the similarity/dissimilarity among annotation categories (e.g., functional roles; the y-axis dendrogram).

drogram trees. Each square in the heatmap dendrogram represents the abundance level of a single category in a single sample. The values used to generate the heatmap/dendrogram figure can be downloaded as a table by clicking on the download button.

Ordination

MG-RAST uses Principle Coordinate Analysis (PCoA) to reduce the dimensionality of comparisons of multiple samples that consider functional or taxonomic annotations. Dimensionality reduction is a process that allows the complex variation found in a large datasets (e.g., the abundance values of thousands of functional roles or annotated species across dozens of metagenomic samples) to be reduced to a much smaller number of variables that can be visualized as simple two- or three-dimensional scatter plots. The plots enable interpretation of the multidimensional data in a human-friendly presentation. Samples that exhibit similar abundance profiles (taxonomic or functional) group together, whereas those that differ are found farther apart.

A key feature of PCoA-based analyses is that users can compare components not just to each other but to metadata recorded variables (e.g., sample pH, biome, DNA extraction protocol) to reveal correlations between extracted variation and metadata-defined characteristics of the samples. It is also possible to couple PCoA with higher-resolution statistical methods in order to identify individual sample features (taxa or functions) that drive correlations observed in PCoA visualizations. This coupling can be accomplished with permutation-based statistics applied directly to the data before calculation of distance measures used to produce PCoAs; alternatively, one can apply conventional statistical approaches (e.g., ANOVA or Kruskal-Wallis test) to groups observed in PCoA-based visualizations.

Table

The table tool creates a spreadsheet-based abundance table that can be searched and restricted by the user. Tables can be generated at user-selected levels of phylogenetic or functional resolution. Table data can be visualized by using Krona [27] or can be exported in BIOM [25] format to be used in other tools (e.g., QIIME [6]). The tables also can be exported as tab-separated text.

Abundance tables serve as the basis for all comparative analysis tools in MG-RAST, from PCoA to heatmap/dendrograms.

Consider the following example showing how to use the taxonomic information derived from an analysis of protein similarities found for the data set 4447970.3. We use the best hit classification, SEED database, 10^{-5} evalue, 60% identity, and a minimal alignment length of 15 amino acids. We select table output. The results are shown in Figure **??**.

The following control elements are connected to the table:

- group by – allows summarizing entries below the level chosen here to be subsumed.

- download table – downloads the entire table as a spreadsheet.
- Krona – invokes KRONA [27] with the table data.
- QIIME – creates a BIOM [25] format file with the data being displayed in the table.
- table size – changes the number of elements to display for the web page.

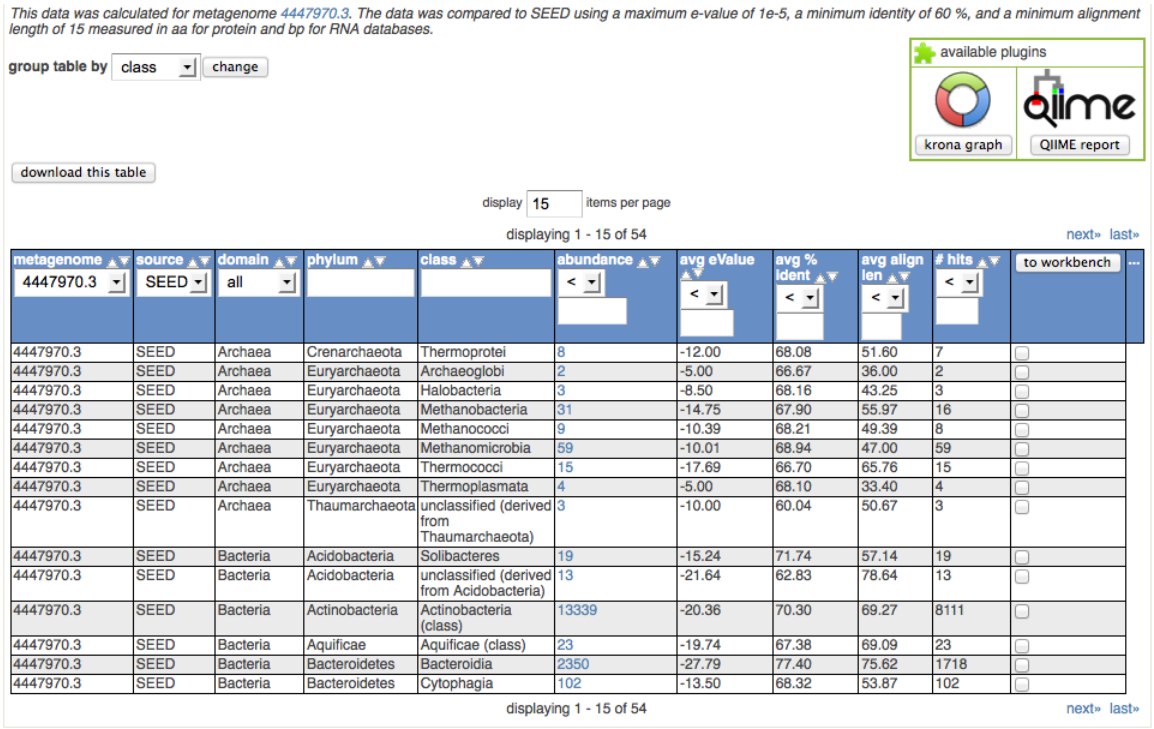


Figure 5.22: View of the analysis page table.

Below we explain the columns of the table and the functions available for them. For each column we allow sorting the table by clicking on the upward- and downward-pointing triangles.

- metagenome

  In the case of multiple datasets being displayed, this column allows sorting by metagenome ID or selection of a single metagenome.

- source

  This displays the annotation source for the data being displayed.

- domain

  The domain column allows subselecting from Archaea, Bacteria, Eukarya, and Viruses.

- phylum, class

  Since we have selected to group results at the class level, only phylum and class are being displayed. The text fields in the column headers allow subsection (e.g., by entering Acidobacteria or Actinobacteria in the phylum field). The searches are performed inside the web browser and are efficient.

  Any subselection will narrow down all datasets being displayed in the table.

  Users can elect to have the results grouped by other taxonomy levels (e.g., genus), creating more columns in the table view.

- abundance

  This indicates the number of sequences found with the parameters selected matching this taxonomic unit. (Note that the parameters chosen are displayed on top of the table.) Clicking on the abundance displays another page displaying the BLAT alignments underlying the assignments.

  The abundance is calculated by multiplying the actual number of database hits found for the clusters by the number of cluster members.

- avg. evalue, avg percent identity, average alignment length

  These indicate the average values for E value, percent identity, and alignment length.

- hits

  This is the number of clusters found for this entity (function or taxon) in the metagenome.

- ...

  This option allows extending the table to display (or hide) additional columns.

The parameter widget

TBA

Evalue, percent identity, length and minimum abundance filters

As shown in Figure **??** MG-RAST can changed the parameters for annotation transfer at analysis time. As each data and each analysis is different, we cannot provide a default parameter set for transferring annotations from the sequence databases to the features predicted for the environmental sequence data.

Instead we provide a tool that puts the user at the helm, providing the means to filter the sequences down by selecting only those matching certain criteria.

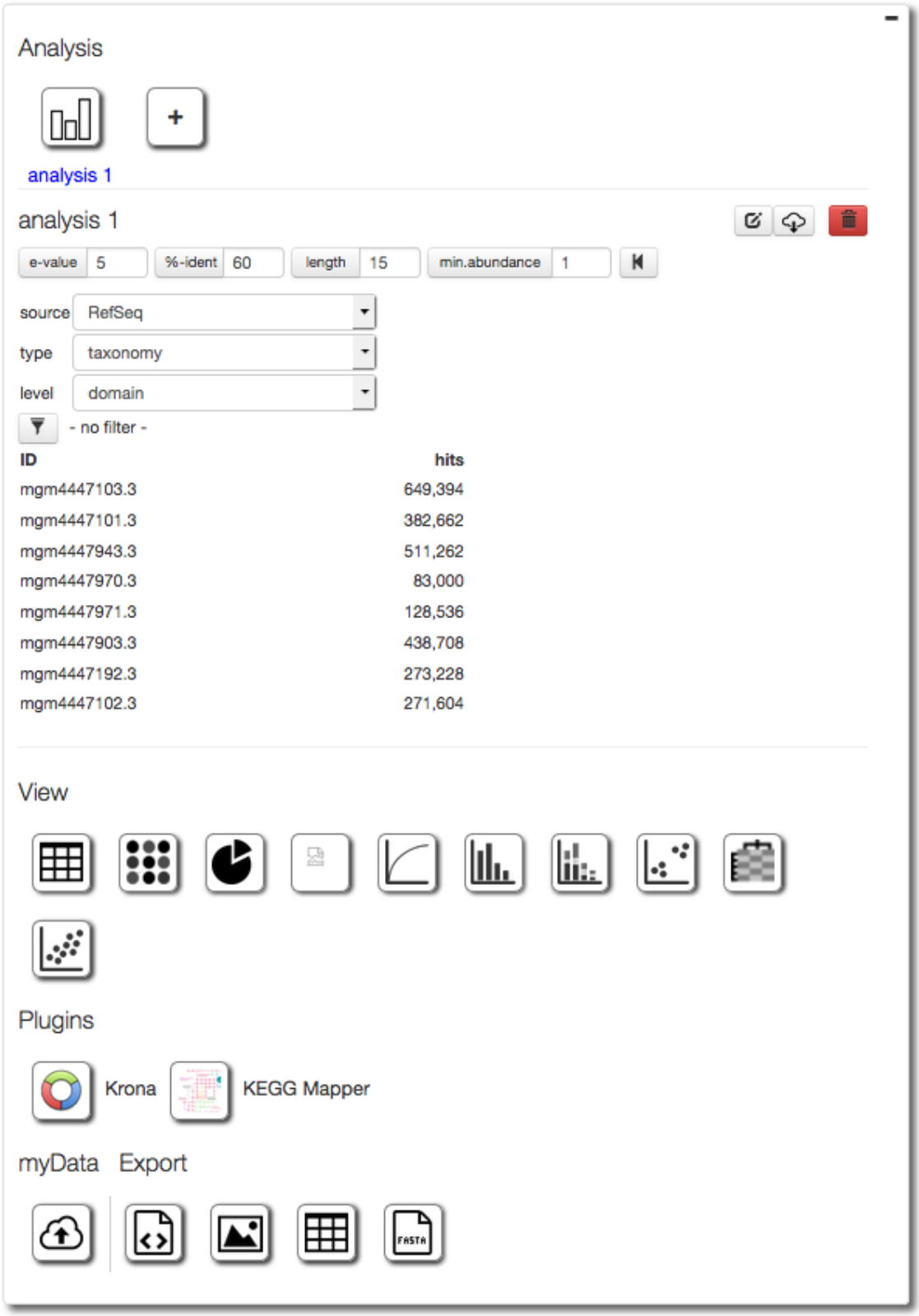


Figure 5.23: After loading all profiles, the analysis parameter widget is displayed.

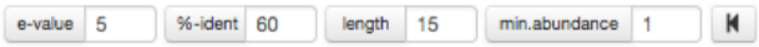


Figure 5.24: By changing the e-value, minimum required percent identity or alignment length the annotations to the features loaded, can be modified. We note that the number of hits listed below the filter is reduced and the display is adjusted instanteneously.

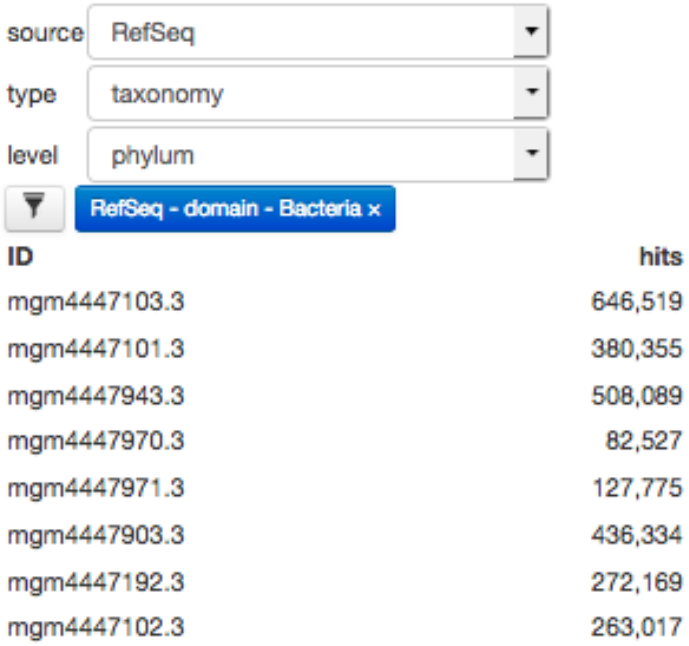


Figure 5.25: Adding a domain level filter for Bacteria. The filter is displayed as a blue box and is clearly labeled.

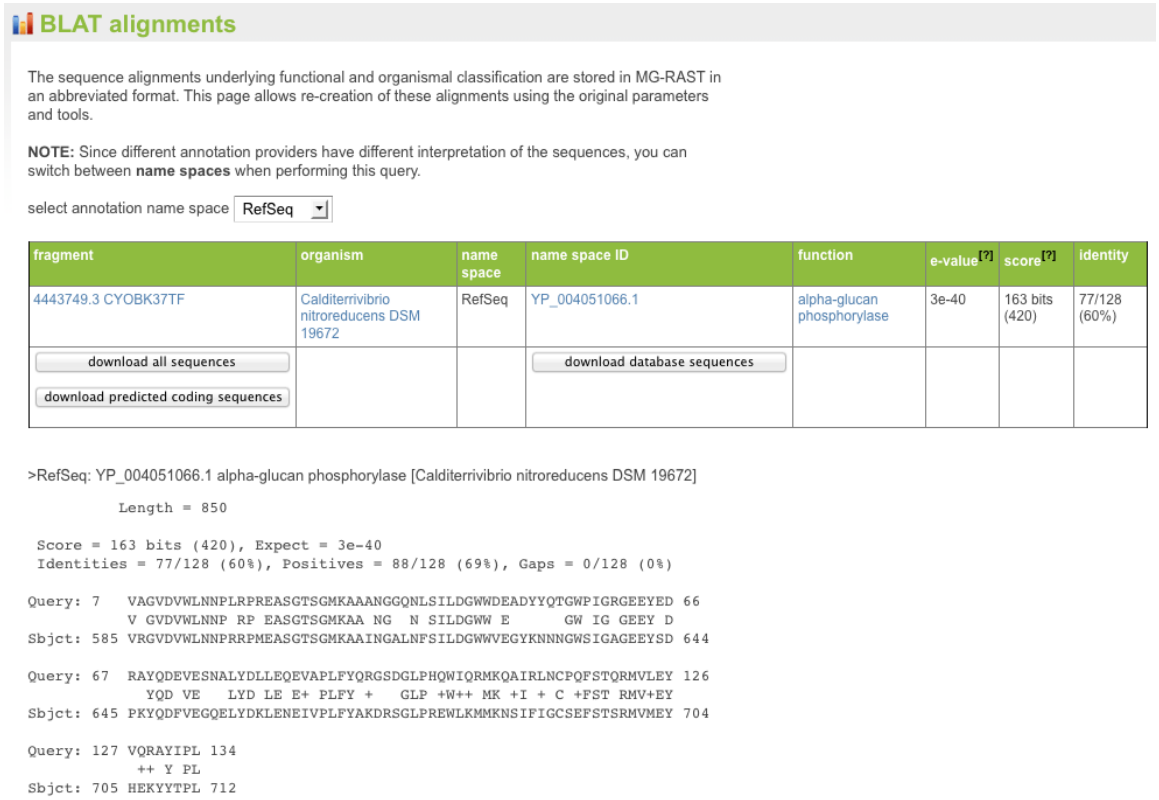
5.4.0.1 *Source type and level filters

Adding one or more filters will limit the scope of the sequences analyzed to e.g. a the domain Bacteria (see Figure 5.25). We note that multiple filters can be used and they can be individually erased when no longer needed. Thus the user can filter, e.g. a certain phylum and the identify reads associated with a specific functional gene group.

5.5 Viewing Evidence

For individual proteins, the MG-RAST page allows users to retrieve the sequence alignments underlying the annotation transfers (see Figure **??**). Using the M5nr [41] technology, users can

retrieve alignments against the database of interest with no additional overhead.



BLAT alignments

The sequence alignments underlying functional and organismal classification are stored in MG-RAST in an abbreviated format. This page allows re-creation of these alignments using the original parameters and tools.

NOTE: Since different annotation providers have different interpretation of the sequences, you can switch between **name spaces** when performing this query.

select annotation name space RefSeq

fragment	organism	name space	name space ID	function	e-value[?]	score[?]	identity
4443749.3 CYOBK37TF	Calditerrivibrio nitroreducens DSM 19672	RefSeq	YP_004051066.1	alpha-glucan phosphorylase	3e-40	163 bits (420)	77/128 (60%)
download all sequences download predicted coding sequences			download database sequences				

>RefSeq: YP_004051066.1 alpha-glucan phosphorylase [Calditerrivibrio nitroreducens DSM 19672]

```
          Length = 850

 Score = 163 bits (420), Expect = 3e-40
 Identities = 77/128 (60%), Positives = 88/128 (69%), Gaps = 0/128 (0%)

Query: 7   VAGVDVWLNNPLRPREASGTSGMKAAANGGQNLSILDGWWDEADYYQTGWPIGRGEEYED 66
           V GVDVWLNNP RP EASGTSGMKAA NG  N SILDGWW E      GW IG GEEY D
Sbjct: 585 VRGVDVWLNNPRRPMEASGTSGMKAAINGALNFSILDGWWVEGYKNNNGWSIGAGEEYSD 644

Query: 67  RAYQDEVESNALYDLLEQEVAPLFYQRGSDGLPHQWIQRMKQAIRLNCPQFSTQRMVLEY 126
             YQD VE   LYD LE E+ PLFY +   GLP +W++ MK +I + C +FST RMV+EY
Sbjct: 645 PKYQDFVEGQELYDKLENEIVPLFYAKDRSGLPREWLKMMKNSIFIGCSEFSTSRMVMEY 704

Query: 127 VQRAYIPL 134
           ++ Y PL
Sbjct: 705 HEKYYTPL 712
```

Figure 5.26: BLAT hit details with alignment.

Chapter 6

API — The MG-RAST Application Programming Interface

6.1 URLs

`http://api.metagenomics.anl.gov/`

Further documentation, with a complete parameter listing for all resources available is at:

`http://api.metagenomics.anl.gov/api.html`

Github repository of script tools, examples, and contributed code for using the MG-RAST API:

`https://github.com/MG-RAST/MG-RAST-Tools`

6.2 Introduction

Over 110,000 metagenomic data sets have been uploaded and analyzed in MG-RAST since 2007, totaling over 43 terabases (TBp). Data uploaded falls in three classes: shotgun metagenomic data, amplicon data, and, more recently, metatranscriptomic data. The MG-RAST pipeline normalizes all samples by applying a uniform pipeline with the appropriate quality control mechanisms for the various data sources. Uniform processing and robust sequence quality control enable comparison across experimental systems and, to some extent, across sequencing platforms. With the inclusion of standardized metadata MG-RAST has enabled meta-analysis available through its web-based user interface. This provides an easy-to-use way to upload and download data, perform analyses, and create and share projects.

As with most GUIs, however, there are limitations to what can be done, for example, regarding the number of samples processed in a single analysis, access to complete metadata, and easy access to raw data and quality metrics for each sample. As part of the DOE Systems Biology

knowledgebase project (KBase) we have implemented a web services application programmers interface (API) that exposes all data to (authenticated) programmers, enabling access to available data and functionality through software applications. This makes user access to MG-RAST's internal data structures possible.

The MG-RAST API enables programmatic access to data and analyses in MG-RAST without requiring local installations. Using the API, users can authenticate against the service, submit their data, download results, and perform extensive comparisons of data sets. The API uses the Representational State Transfer (REST) [3] architecture which allows download of data in ASCII format, allowing users to query the system via URLs and returning MG-RAST data objects in their native format (e.g. similarity tables or sequence files). For structured data (e.g. metadata or project information) the MG-RAST API uses JSON (Javascript Object Notation, a widely used standard) as its data format.

This allows users to use simple tools to download data files or view the JSON in their web browsers using one of the many available JSON viewers. In addition many programming languages have libraries for convenient HTTP interaction and JSON conversions. The API has a minimal number of prerequisites; and any language with HTTP and JSON support or command line utilities such as "curl" can easily integrate with the design.

If you are not a programmer or you are not willing to spend the time learning the API, the Example scripts (see chapter 7.)

6.3 Design and Implementation

The MG-RAST API enables programmatic access to data and analyses in MG-RAST without requiring local installations. Users can authenticate against the service, submit their data, download results, and perform extensive comparisons of data sets. We chose to use the Representational State Transfer (REST) [3] architecture. The REST approach allows download of data in ASCII format, allowing users to query the system via URLs and returning MG-RAST data objects in their native format (e.g. similarity tables or sequence files). For structured data (e.g. metadata or project information) the MG-RAST API uses JSON (Javascript Object Notation, a widely used standard) as its data format.

Using this approach users can use simple tools to download data files to their machines or view the JSON in their web browsers using one of the many available JSON viewers. In addition many programming languages have libraries for convenient HTTP interaction and JSON conversions.

Most of the API calls are simply URLs which can be entered in the address bar of a web browser to perform the download through the browser. These URLs can also be used with a command line tool like curl, in programing-language-specific libraries, or in command line scripts. The examples in the Results section illustrate the use of each of these methods. The example scripts are available on in the supplementary materials and on github (https://github.com/MG-RAST/MG-RAST-Tools)

along with other useful illustrative scripts.

The MG-RAST API covers most of the functionality available through the MG-RAST website, with access to annotations, analyses, metadata and access to the MG-RAST user inbox to view contents as well as upload files. All sequence data and data products from intermediate stages in the analysis pipeline are available for download. Other resources provide services not available through the website, e.g. the m5nr resource lets you query the m5nr database.

Each query to the API is represented as a URI beginning with

```
http://api.metagenomics.anl.gov/
```

and has a defined structure to pass the requests and parameters to the API server. These URI queries can be used from the command line, e.g. using curl, in a browser, or incorporated in a shell script or program.

Each URI has the form:

```
http://api.metagenomics.anl.gov/{version}/{resourcepath}?{querystring}
```

where

```
{version}
```

explicitly directs the request to a specific version of the API. If it is omitted the latest API version will be used. The current version number is '1'.

```
{resourcepath}
```

is constructed from the path parameters listed below to define a specific resource.

```
{querystring}
```

is used to filter the results obtained for the resource, this is optional.

For example, in:

```
http://api.metagenomics.anl.gov/1/annotation/sequence/mgm4447943.3?
   evalue=10&type=organism&source=SwissProt
```

the resource path

```
annotation/sequence/mgm4447943.3
```

defines a request for the annotated sequences for the MG-RAST job with ID 4447943.3. The optional query string

```
evalue=10&type=organism&source=SwissProt
```

modifies the results by setting an evalue cutoff, annotation type and database source.

The API provides an authentication mechanism for access to private MG-RAST jobs and users' inbox. The 'auth_key' (or 'webkey') is a 25 character long string, e.g.

```
j6FNL61ekNarTgqupMma6eMx5
```

which is used by the API to identify an MG-RAST user account and determine access rights to metagenomes. Note that the auth_key is valid for a limited time after which queries using the key will be rejected. You can create a new auth_key or view the expiration date and time of an existing auth_key on the MG-RAST website. An account can have only one valid auth_key and creating a new key will invalidate an existing key.

All public data in MG-RAST is available without an auth_key. All API queries for private data which either do not have an auth_key or use an invalid or expired auth_key will get a "insufficient permissions to view this data" response.

The auth_key can be included in the query string like:

```
http://api.metagenomics.anl.gov/1/annotation/sequence/mgm4447943.3?
   evalue=10&type=organism&source=SwissProt&auth_key=
   j6FNL61ekNarTgqupMma6eMx5
```

or in a request using curl like:

```
curl -X GET -H "auth: j6FNL61ekNarTgqupMma6eMx5" "http://api.
   metagenomics.anl.gov/1/annotation/sequence/mgm4447943.3?evalue=10&
   type=organism&source=SwissProt"
```

Note that for the curl command the quotes are necessary for the query to be passed to the API correctly.

If an optional parameter passed through the query string has a list of values only the first will be used. When multiple values are required, e.g. for multiple md5 checksum values, they can be passed to the API like:

```
curl -X POST -d '{"data":["000821a2e2f63df1a3873e4b280002a8","15
   bf1950bd9867099e72ea6516e3d602"]}' "http://api.metagenomics.anl.gov
   //m5nr/md5"
```

In some cases, the data requested is in the form of a list with a large number of entries. In these cases the 'limit' and 'offset' parameters can be used to step through the list, e.g.

```
http://api.metagenomics.anl.gov/1/project?order=name&limit=20&offset
   =100
```

will limit the number of entries returned to 20 with an offset of 100. If these parameters are not provided default values of `limit=10` and `offset=0` are used. The returned JSON structure will contain the 'next' and 'prev' (previous) URIs to simplify stepping through the list.

The data returned may be plain text, compressed gzipped files or a JSON structure.

Table 6.1: Top-level resources available through the MG-RAST-API

Resource/Object	Description
annotation	taxonomic and functional annotations made by comparison with the M5nr database
compute	resource to compute PCoA , heatmap, and normalization for a set of input metagenomes
download	download results of the MG-RAST pipeline
inbox	upload and listing of data in the staging area prior to execution of the MG-RAST pipeline
library	library information for uploaded metagenome provided by the user
matrix	abundance profiles in BIOM (5) format for a list of metagenomes
M5nr	access M5 nonredundant protein database used for annotation of metagenomic sequences
metadata	creation, export, and validation of metadata templates and spreadsheets
metagenome	container for sample, library, project, and precomputed data for an uploaded metagenomic sequence file
profile	returns a single data object in BIOM format
project	project summary for metagenome provided by user
sample	sample information provided by user
search	search MG-RAST by MG-ID, metadata, function, or taxonomy; or implement a more complex search.
validation	validates templates for correct structure and data

Most API queries are 'synchronous' and results are returned immediately. Some queries may require a substantial time to compute results, in these cases you can select the asynchronous option by adding `'&asynchronous=1'` to the end of the query string. This query will then return a URL which will return the query results when they are ready.

Most of the API calls are simply URLs which can be entered in the address bar of a web browser to perform the download through the browser. These URLs can also be used with a command line tool like curl, in programing-language-specific libraries, or in command line scripts. The examples below illustrate the use of each of these methods. The example scripts are available on the github site along with other useful illustrative scripts.

6.4 Examples

The API provides index-driven access to data subsets using the following data types as indices into the data: functions, functional hierarchy data, and taxonomic data. Whenever possible we have employed standards to expose data and metadata, such as the BIOM standard for encoding abundance profiles. The examples below are intended to illustrate usage for the various resources available, they do not cover the entire functionality of the API, see the documentation at the API website for the comprehensive listing.

- **annotation**

  ```
  http://api.metagenomics.anl.gov/1/annotation/sequence/mgm4440036.3?
     type=function&filter=protease&source=Subsystems
  ```

  Retrieve the reads from a metagenome with ID mgm4440036.3 which were annotated as protease in SEED Subsystems.

- **download**

  ```
  http://api.metagenomics.anl.gov/1/download/mgm4447943.3
  ```

  Retrieve information formatted as a JSON object about all the files available for download for metagenome mgm4447943.3 with information about the files and sequence statistics where applicable. Each file listed has a URL included which can be used to download the file, e.g.

  ```
  http://api.metagenomics.anl.gov/1/download/mgm4447943.3?file=650.1
  ```

  will download the protein.sims file containing the BLAT similarities.

- **inbox**

  ```
  curl -X POST -H "auth: auth_key" -F "upload=@sequences.fastq" "http
     ://api.metagenomics.anl.gov/1/inbox"
  ```

  Upload the file 'sequences.fastq' to your inbox. This API call requires user authentication using the auth_key described above. It can not be used in a browser, but needs to be run from the command line or from a script.

- **matrix**

  ```
  http://api.metagenomics.anl.gov/matrix/organism?group_level=family&
     source=SEED&evalue=5&id=mgm4440442.5&id=mgm4440026.3
  ```

  Retrieve the taxonomic abundance profile on family level for 2 metagenomes based on SEED assignments with an evalue cutoff of 1e-5.

- **metagenome**

```
http://api.metagenomics.anl.gov/1/metagenome/mgm4440026.3
```

List analysis submission parameters and other details for a metagenome.
The metagenome resource can also be used to search metadata, function and taxonomy.

```
http://api.metagenomics.anl.gov/metagenome?function=dnaA&organism=
   coli&biome=marine&match=all&order=created
```

This call will find all marine metagenomes with reads annotated as dnaA and have taxonomic assignment containing the text 'coli', the results will be ordered based on creation date for the metagenome.

- **project**

```
http://api.metagenomics.anl.gov/project/mgp31?verbosity=full
```

Retrieve available information about the project with ID mgp31.

- **sample**

```
http://api.metagenomics.anl.gov/1/sample/mgs12326?verbosity=full
```

Retrieve available information about individual samples, including IDs and metadata.

- **metadata**

```
http://api.metagenomics.anl.gov//metadata/template
```

Retrieve the static template for metadata object relationships and types used by MG-RAST.

```
http://api.metagenomics.anl.gov//metadata/export/mgp128
```

Retrieve all metadata for project mgp128.

```
http://api.metagenomics.anl.gov/metadata/cv
```

Retrieve a set of lists of all our controlled metadata terms, including the ontologies.

```
http://api.metagenomics.anl.gov/metadata/ontology?name=biome&
   version=2013-04-27
```

Retrieve a more detailed list (with relationships) for a specific version of the ontology.

- **m5nr**

```
http://api.metagenomics.anl.gov/1/m5nr/md5/
   ffc62262a18b38671c3e337150ef535f?source=SwissProt
```

Retrieve the UniProt ID for a given sequence identifier.

Chapter 7

Example scripts using the MG-RAST REST API

.

7.1 Introduction

As part of the RESTful API (see chapter 6), we are providing a collection of example scripts.

Each script has comments in the source code as well as a help function. This document provides a brief overview of the available scripts and their intended purpose. Please see the help associated with all of the individual files for a complete list of options and more details.

We believe these scripts to be the best starting point for many users, he we attempt to provide a listing of the most important tools.

7.1.1 URLs

The Examples are located on github at:

`https://github.com/MG-RAST/MG-RAST-Tools`

This is the base directory for the rest of this chapter, go here to find the tools and examples described below:

`https://github.com/MG-RAST/MG-RAST-Tools/tree/master/tools/bin`

Each script has a verbose help option (–help) to list all options and explain their usage.

7.2 Download DNA sequence for a function – mg-get-sequences-for-function.py

This script will retrieve sequences and annotation for a given function or functional class.

The output is a tab-delimited list of: m5nr id, dna sequence, semicolon seperated list of annotations, sequence id.

Example:

```
mg-get-sequences-for-function.py --id "mgm4441680.3" --name "
   Central carbohydrate metabolism" --level level2 --source
   Subsystems --evalue 10
```

7.3 Download DNA sequences for a taxon or taxonomic group– mg-get-sequences-for-taxon.py

This script will retrieve sequences and annotation for a given taxon or taxonomic group.

The output is a tab-delimited list of: m5nr id, dna sequence, semicolon seperated list of annotations, sequence id

Example:

```
mg-get-sequences-for-taxon.py --id "mgm4441680.3" --name
   Lachnospiraceae --level family --source RefSeq --evalue 8
```

7.4 Download sequences annotated with function and taxonomy – mg-get-annotation-set.py

Retrieve functional annotations for given metagenome and organism.

The output is a tab-delimited list of annotations: feature list, function, abundance for function, avg evalue for function, organism.

Example:

```
mg-get-annotation-set.py --id "mgm4441680.3" --top 5 --level
   genus --source SEED
```

7.5 Download the n most abundant functions for a metagenome – mg-abundant-functions.py

Retrieve the top n abundant functions for metagenome.

The output is a tab-delimited list of function and abundance sorted by abundance (largest first). 'top' option controls number of rows returned.

Example:

```
mg-abundant-functions.py --id "mgm4441680.3" --level level3
   --source Subsystems --top 20 --evalue 8
```

7.6 Download and translate similarities into different namespaces e.g. SEED or GenBank – m5nr-tools.pl

MG-RAST computes similarities against a non-redundant database [41] and later translates them into any of the supported namespaces. As a result you can view your annotations (or indeed the similarity results) in each of these namespaces. Sometimes this can lead to new features and or differences becoming visible that would otherwise be obscured.

m5nr-tools can translate accession ids, md5 checksums, or protein sequence into annotations. One option for output is a blast m8 formatted file.

Example:

```
m5nr-tools.pl --api "http://api.metagenomics.anl.gov/1" --option
   annotation --source RefSeq --md5 0
   b95101ffea9396db4126e4656460ce5,068792
   e95e38032059ba7d9c26c1be78,0b96c92ce600d8b2427eedbc221642f1
```

7.7 Download multiple abundance profiles for comparison – mg-compare-functions

Retrieve matrix of functional abundance profiles for multiple metagenomes. The output is either tab-delimited table of functional abundance profiles, metagenomes in columns and functions in rows or BIOM format of functional abundance profiles.

Example:

```
mg-compare-functions.py --ids "mgm4441679.3,mgm4441680.3,
   mgm4441681.3,mgm4441682.3" --level level2 --source KO --
   format text --evalue 8
```

Chapter 8

FAQ – Frequently asked questions about MG-RAST

The answers to some of these Frequently Asked Questions can be found elsewhere in this manual, they are listed here for users who would like a quick answer to a simple question. Other sections of the manual will generally contain more detail than the answers in this chapter. Some answers are just links to relevant sections in other chapters.

8.1 General

What is MG-RAST?

The MG-RAST server is an open source system for annotation and comparative analysis of metagenomes. Users can upload raw sequence data in fasta format; the sequences will be normalized and processed and summaries automatically generated. The server provides several methods to access the different data types, including phylogenetic and metabolic reconstructions, and the ability to compare the metabolism and annotations of one or more metagenomes and genomes. In addition, the server offers a comprehensive search capability. Access to the data is password protected, and all data generated by the automated pipeline is available for download in a variety of common formats.

Contacting the MG-RAST team and help desk

The MG-RAST project uses a ticket system to manage interactions with users, please use the email address for the MG-RAST project shown in Figure 8.1.

We recommend including as much detail as possible into your emails to the help-desk, details like account names, MG-RAST identifiers will help us identify any issues and speed up resolving them.

Figure 8.1: The email address for the MG-RAST project. Note that it was inserted into this document as an image and can not be copied as text, you will have to type it.

Below are examples of the types of details we would like to receive:

- your name
- your account name for MG-RAST (please do NOT include your password or webkey)
- a clear text description of your problem
- any MG-RAST identifiers (those are the 444xxxx.3 numbers)
- any project numbers
- the browser and which version you are using, if the problem relates to the web site
- what platform your data was created on
- if your data was a failure in the web site, what time the failure occurred
- the URL and name of the page you were viewing
- screenshot(s) of the problem

What kinds of data sets does MG-RAST analyze?

MG-RAST is designed to annotate a large set of nucleotide sequences, not a complete genome and not amino acid sequences. The RAST server should be used if you want to annotate complete, or nearly complete prokaryotic genomes. Version 3.2 accepts reads of length 75bp and up, and is capable of handling sequences of several dozen kilobases. For whole metagenome shotgun data we use a gene prediction step that is not suitable for eukaryotes, for that reason do not expect MG-RAST v3.2 to work with eukaryotic data sets or for the eukaryotic subsets of your data.

How many metagenomes can I submit?

We do not restrict user submission of samples. However, the computation required is massive and samples are processed on a first-come, first-serve basis. MG-RAST v3 is over 200 times faster than the previous version. We will also provide a CLOUD client (shortly after the initial release) that connects to MG-RAST and will allow you to add processing power to your jobs in MG-RAST.

Can I use MG-RAST as a repository for my metagenomic data?

MG-RAST has become an unofficial repository for metagenomic data, providing a means to make your data public so that it is available for download and viewing of the analysis without registration, as well as a static link that you can use in publications. It also requires that you include experimental metadata about your sample when it is made public to increase the usefulness to the community. We undertake to maintain public datasets within MG-RAST and they are not subject to deletion.

Who should I contact with questions or problems with MG-RAST?

All questions, comments or problems regarding MG-RAST should be directed to our support team using either the letter symbol in the navigation toolbox or via email to: `mg-rast at mcs.anl.gov`.

How should I link to MG-RAST in a publication?

You can provide a stable link to an MG-RAST job or project using the following URLs:

```
http://metagenomics.anl.gov/linkin.cgi?metagenome=
http://metagenomics.anl.gov/linkin.cgi?project=
```

For example, for the metagenome ID 4440283.3 the URL is:

```
http://metagenomics.anl.gov/linkin.cgi?metagenome=4440283.3
```

This URL provides a stable method of linking to your data which does not require the viewer to have an MG-RAST account. Please do not use the URL you see when you are browsing the site. Note that by default your data is not visible to others, you will need to explicitly grant permission for it to be visible to anyone on the internet by making it public through the MG-RAST website.

Identifiers

MG-RAST automatically assigns a unique identifier to every dataset submitted. Upon completion of the automated pipeline, datasets can be viewed via the web interface by using the identifiers. The dataset identifiers are of the form `integer prefix.revision`. An example is `4440283.3`.

In addition to individual datasets, projects (groups of datasets) can be addressed with simple numerical project identifiers. An example is `128`.

Linking to MG-RAST

Because future versions of MG-RAST may change, we provide a link-in mechanism as a stable way of linking to MG-RAST. To link to datasets or projects in MG-RAST, users should always use the `linkin.cgi`, especially in publications.

Note: You must make the data set PUBLIC before you can publicly share the link. It will not work for others until you do.

Note: Do not use the URL that is displayed in the browser when browsing the site.

```
http://metagenomics.anl.gov/linkin.cgi?metagenome=
http://metagenomics.anl.gov/linkin.cgi?project=
```

Figure 8.2: Stable URLs provided by the `linkin.cgi` mechanism for linking to MG-RAST.

For example, for the public dataset with metagenome ID `4440283.3` the URL is: `http://metagenomics.anl.gov/linkin.cgi?metagenome=4440283.3`. For the public project with project ID `128` the URL is: `http://metagenomics.anl.gov/linkin.cgi?project=128`.

These URLs provides a stable method of linking to data that does not require the viewer to have an MG-RAST account.

Privacy

By default, a user's data is not visible to others; the user needs to explicitly grant permission for the data to be visible to anyone on the Internet, either by sharing with individuals or by making it public through the MG-RAST website. Only the owner of a dataset (the original submitter) can make a dataset public and this requires explicit action on their part, MG-RAST does not make data public without this action. Owners can grant anonymous access to manuscript reviewers (see Section 8.1).

The web interface allows sharing and publication of data, requiring the presence of minimal metadata (see Section 4.7) for data that is made public. Data can be shared or made public only after the computation has finished.

Sharing with individual users

Data and analyses can be shared with individual users. To share data, users simply enter their email address via clicking the `Sharing` link on the Metagenome Overview page. The dialogue shown in Figure 8.3 will allow entering email addresses.

Both individual jobs as well as entire projects containing one or more jobs can be shared using a similar mechanism from the Job Overview and Project pages respectively.

As shown in Figure 8.4, we tend to see dataset sharing between small groups of users.

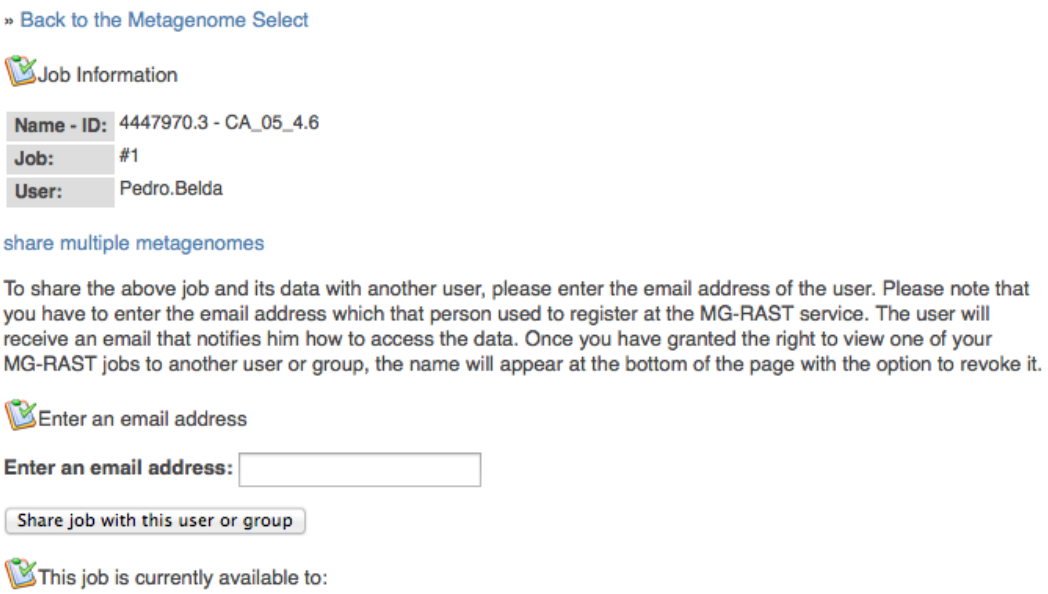


Figure 8.3: Dialogue showing the sharing mechanism. The mechanism requires a valid email address for the user with whom the data is to be shared. A list of users with access to the data is displayed at the bottom on the page.

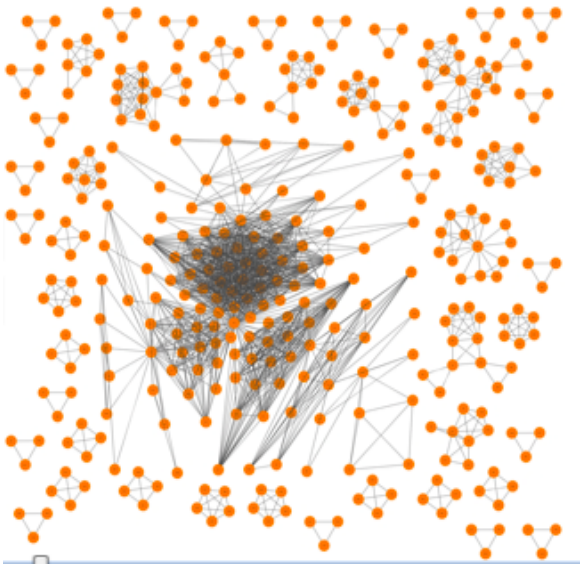


Figure 8.4: Data sets shared in MG-RAST by users (orange dots), shown as connecting edges.

Anonymous sharing with reviewers

To grant manuscript reviewers access to a project while preserving their anonymity click on the 'Create Reviewer Access Token' button on the project page. This button is visible only to the owner of a project by clicking on the 'Share Project' link. It will generate a token that can be sent to the publisher to pass on to reviewers. When a reviewer receives the token from the publisher they need to use the included link to access MG-RAST. If necessary the reviewer will need to register for an account and their account will have anonymous access to the project. The number of reviewers who have accessed the project is displayed to the owner in the list of users the project is shared with, but the identity of the reviewers is not disclosed. The owner of the project can revoke the token at any time to disable access.

Publishing

MG-RAST provides a mechanism to make data and analyses publicly accessible. All sequence data, metadata, analyses, and analyses files for a dataset will be freely available for download once it is made public. Only the submitting user can make data public on MG-RAST and once this is done it can not be reversed. Metadata is mandatory for dataset publication (see Section 4.7).

The following checklist describes the process of making MG-RAST datasets and projects public:

1. Ownership of the datasets: To make a dataset public your account needs to be labelled as the owner in MG-RAST.

2. Ownership of the project: Your account should be the owner of the project as well, this is usually just the account that was used to create the project.

3. Metadata: MG-RAST requires that you enter metadata for the project, samples and libraries before it is made public to increase its utility to the community. This is done through a pre-formatted excel spreadsheet which you fill in with the necessary metadata. If you have already entered metadata, e.g. during submission, and want to make changes, you can download this file with the existing metadata prefilled from the project page with the 'Export Metadata' link.

   If you have not entered metadata for your project, download the latest metadata template file from: `ftp://ftp.metagenomics.anl.gov/data/misc/metadata/` The first sheet is a README containing some important tips for entering the metadata. The second row in each sheet in the template contains some explanation and instructions for each column. The columns marked with red headers are required.

   You can enter your data directly into the template, a better route would be to use the tool we built to facilitate metadata entry – MetaZen: `http://metagenomics.anl.gov/`

metazen.cgi. MetaZen will step you through the data entry and then give you a pre-filled excel spreadsheet to download which you can then edit further if necessary.

Once you have the metadata file ready, upload with the 'Upload Metadata' link on the project page.

4. Release metagenomes: Make each dataset public, there is a 'Make public' link in the blue bar near the top of the Metagenome Overview page.

5. Project Data: Edit the project page information if you wish with the 'Edit Project Data' link. You can enter an abstract, links to publications, additional description, contacts etc. This page is the central point in MG-RAST from where people will access your data and analyses so add all information that may be useful.

6. Final step: Make the project public from the project page (project page blue bar, 'Make Public').

The link for a public project which should be used in a publication is listed near the top of the project page, e.g.: http://metagenomics.anl.gov/metagenomics.cgi?page=MetagenomeProject&project=128 where 128 is the MG-RAST project ID.

The link for individual public metagenomes which should be used in a publication is listed near the top of the metagenome overview page, e.g.: http://metagenomics.anl.gov/linkin.cgi?metagenome=4440283.3 where 4440283.3 is the MG-RAST metagenome ID.

The publication to cite for MG-RAST is at http://www.biomedcentral.com/1471-2105/9/386.

Who should I cite when I use this service?

See Section 1.4.

Is MG-RAST open source and can I install it locally?

MG-RAST is indeed open source. We make the current stable versions available on github: https://github.com/MG-RAST/ However MG-RAST is a complex system to install (note: we have not been funded to create a readily installable version) and even more complex to operate. We advise against attempting to create a private installation and can not provide any help installing MG-RAST locally.

If you are a biologist worried about runtime of your jobs, there is a way to run your jobs on computational resources provided by you that will significantly help. Please contact us at our usual address mg-rast at mcs.anl.gov to inquire about ways of setting this up.

If you are a bioinformatician and want to contribute code or test alternatives for individual steps, we are currently preparing a system that will make all components of MG-RAST easily accessible. This is not currently sea-worthy. Same as with the biologists, please contact us at mg-rast at mcs.anl.gov for details.

8.2 Accounts

The analyses of all public datasets in MG-RAST can be viewed in entirety without an MG-RAST account. An account is required to submit sequence data for analysis or view the analyses of datasets which have been shared with you.

Accounts are for individuals, not services or groups. In our experience account sharing (e.g. two or more users having access to the same username/password information) will always lead to problems, we **strongly** discourage account sharing.

As scientist typically will switch employers every few years we encourage users to provide two email addresses, the primary email address could be your work email, the secondary your private email. By providing a second email address you can avoid losing access to your account if and when you switch employers and your work email is no longer available.

Account registration

Use the "Register" link on the front page of the website to request an account with MG-RAST, you will need to enter a unique login name and email address along with other minimal information. Use an email address you use regularly as it will be used to communicate with you when necessary. After registering you will receive an automated email with a temporary password after your account has been authorized, usually within a day.

If you forget your password you can request a new password on the MG-RAST website using your login and registered email address, a new password will be generated and sent by email to this address.

Account webkey

The webkey is a unique string of text, e.g. "b8Dvg2d5DCp7KsWKBPzY2GS4i" associated with your account which is used by MG-RAST for identification purposes. Your webkey is valid for a limited time period after which it expires and will not work anymore. You can generate a new webkey at any time, even if your current webkey has not expired.

The MG-RAST website provides two locations where you can generate a new webkey:

1. Log in to MG-RAST and go to the Account Management page. Press the button under "Preferences" to go the the Manage Preferences page where the Web Services section dis-

plays your current webkey with its termination date. Click on the “generate new key” button to generate a new key and then click the “set preferences” button.

2. Log in to MG-RAST and go to the Upload page and click on the “generate webkey” button in the “upload files” tab and then click on the “generate new key” button.

Note that generating a new webkey will invalidate your old webkey and your new webkey will be valid until the termination date displayed on the page.

Why do I need to register for this service?

If you do not plan to submit data for analysis to MG-RAST and only want to browse data which is publicly available there is no need to register. Otherwise we request that users register, with a valid email address, so we can contact you once the computation is finished or in case user intervention is required.

I have forgotten my password, what should I do?

In the navigation toolbox (top right corner of the webpage) there is a ’Forgot?’ link displayed. Click on this and enter your login and the email address you registered with MG-RAST. A changed password will be sent by email to this address. For security purposes you should login and change this new password as soon as you receive the email.

Can I change my account information?

Yes, you can change or modify your password, email address, name and funding source for your account. Login and make the changes on the account management page.

8.3 Upload and Submission

MG-RAST was designed to allow users to upload sequence data directly from next-generation sequencing machines. Data can be in FASTA, FASTQ, or SFF format.

We suggest uploading raw data (in FASTQ or SFF format) and letting MG-RAST perform the quality control step because this approach will allow us to identify any issues with the sequencing run. Frequently, local quality control will identify some issues but mask others.

Compressing large files will reduce the upload time and the chances of a failed upload. Users can upload gzip (.gz) and bzip2 (.bz2) or Zip (.zip) files, as well as tar archives compressed with gzip (.tar.gz) or bzip2 (.tar.bz2).

It is not necessary to assemble data prior to upload to MG-RAST. The system has been optimized for short reads and can handle uploads of many hundreds of gigabytes.

Figure 8.5: The flow for MG-RAST submissions via the web interface

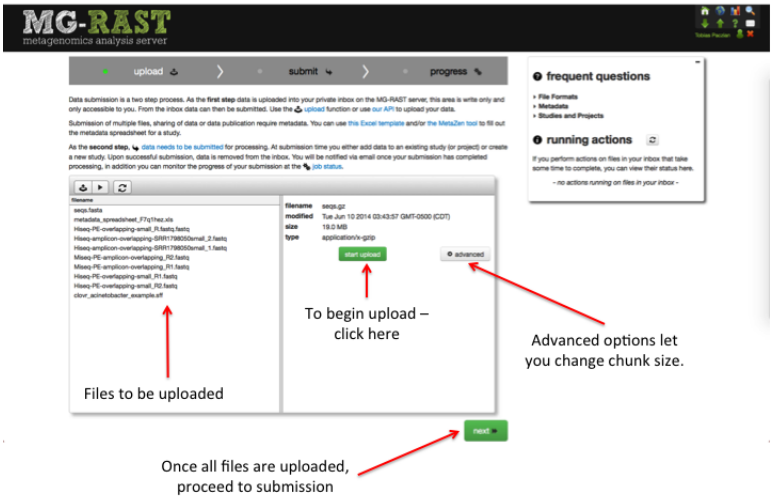


Figure 8.6: The MG-RAST upload page with its three main stages

Assembled data can be uploaded to MG-RAST and read abundance information for contigs can be imported as well from FASTA files. The “assembled” option for the pipeline will attempt to retrieve read abundance information from the FASTA sequence files.

Data submission via the web interface

To start uploading data to MG-RAST through the website, click on the green up arrow. Doing so opens the Upload page. On this page you can upload files, modify the files where needed, add metadata, and submit files for analysis.

The page has three stages (see Figure 8.3). The first “Upload” to upload, manipulate, and collect all the files required for a submission, and “Submit,” to create the MG-RAST job(s), set analysis parameters, and start the analysis. The last is “Progress”, where you can monitor your job status.

Starting with version 3.6 of MG-RAST, the web upload page will provide significantly more user guidance and help with ensuring the files uploaded are both compliant with the required naming scheme and are transferred intact.

Data requirements for upload

Files larger than 50 MB should be compressed before upload, using gzip (preferable), bzip2 or Zip (less than 4 GB in size). Compression will reduce the time taken for the upload of the file, which in turn reduces the chance that the upload will fail. The requirements for submission are sequence information (required), metadata (strongly recommended) and barcode information (for multiplexed datasets only).

We note that priority will be giving to data that has compete GSC metadata and has been marked for eventual release to the public. The data release is under user control, MG-RAST staff will not release the data for the user.

To ensure files are uploaded properly, MG-RAST performs automatic MD5[1] checking on client and server side (for most files) to ensure that files are received correctly by MG-RAST. This is an important part of data hygiene as files may get corrupted in flight. The new interface (from version 3.6 onwards), will check the integrity and will give you immediate feedback about whether your upload was successful. If not detected at upload time, a damaged file will lead to errors later in the pipeline, wasting both valuable compute cycles and, even more importantly, your time.

All files uploaded to MG-RAST should be named using only alphanumeric and ._ characters without spaces. As of version 3.6, the upload system ensures that files are compliant with the mandatory naming scheme, using only alphanumeric and .-characters without spaces. In addition, there is no need to extract/uncompress files after upload. MG-RAST does this automatically along with checking metadata and sequence file format and nomenclature compliance.

Advanced options provides the option to change chunk size. Chunked uploading allows us to break a large file into small chunks, and send these pieces to the upload server one-by-one. If an upload fails, we need only resume from the last successful chunk and allows for resuming uploads. As a rule, the larger the file and the faster your connection, the larger the chunk size should be. Set the size lower if your connection is slow. We have a default setting that works well for most data sets and connection speeds. If you are encountering upload failure (outside of formatting issues), try a smaller chunk size.

The following three kinds of files can be uploaded:

- **Sequence files**

  Sequence files must be in either FASTA, FASTQ, or SFF formats

  Sequence file names must have one of the following extensions – '.fasta', '.fna', '.fastq', '.fq', or '.sff'.

  FASTA and FASTQ files should be in plain text ASCII.

[1] An MD5 checksum is a widely used way to create a digital fingerprint for a file. Think of it as a kind of checksum, if the fingerprint changed, so did the file. The fingerprints are easy to compare. There are many tools out there for creating MD5 checksums, google is your friend.

FASTA files (and all sequence data submitted to MG-RAST) should not contain protein sequences.

Assembled data with read abundance information must be in FASTA format and the coverage included in the sequence ID using the following simple format:

```
>sequence_number_1_[cov=2]
CTAGCGCACATAGCATTCAGCGTAGCAGTCACTAGTACGTAGTACGTACC
>sequence_number_2_[cov=4]
ACGTAGCTCACTCCAGTAGCAGGTACGTCGAGAAGACGTCTAGTCATCAT
....
```

The abundance information must be appended without spaces to the end of the sequence name (also without whitespace) in the format "_[cov=n]", where n is the coverage or abundance of each contig. Sequence files in this format should be submitted with the "assembled" option selected and the pipeline will retrieve read abundance information from the sequence file.

- **Metadata file**

  We provide a spreadsheet template that can be filled out with all the available metadata information for a dataset, there is a link to the template on the upload page. Download the template and edit to include as much information as is available. While the number of fields in the template is large, the number of required fields, colored in red in the template, is small. The template file can be used to upload metadata for one or multiple samples and submit them to MG-RAST as a single project. The metadata can be modified at any time after submission to add information or to correct errors. See Section 8.3 for more details.

  We note that a good strategy is to copy an existing metadata file and modify the values appropriately. Our experience has also shown that editing the metadata file with tools other than Microsoft Excel will corrupt the files.

- **Barcode file**

  Barcoding reads allows multiplexing multiple samples into a single sequence file. Barcode files allow demultiplexing those files. Consequently, Barcode files are required only for sequence data which will be demultiplexed on the MG-RAST website. In many cases (typically for shotgun metagenomes) the demultiplexing will have already been done by the sequencing center. If you have demultiplexed sequence data, you do not need to enter the barcodes associated with your samples in a Barcode file. While suitable for all kinds of barcodes and sequence data, we expect the built-in demultiplexing to be used mostly for custom barcoded amplicon sequences.

The barcode file should be in plain text ASCII.

If the sequencing facility generated the libraries and did not demultiplex them for you, make sure to get the barcodes corresponding to each of your samples. The barcode file should be in plain text ASCII, a downloadable example can be found at: `ftp://ftp.metagenomics.anl.gov/data/manual/example/`.

Each line of the file should contain a single barcode sequence followed by a tab and then a unique filename, with as many lines as necessary for the barcodes in the sequence file you are submitting. Additional columns are ignored.

```
Example:
ACTCTCGTG    sample_1
CAGACATCT    sample_2
GTAGATCAC    sample_3
```

The barcode file typically will be provided by whoever created the amplicons, in many cases that is the sequencing center.

Uploading data

In this first step, data is uploaded into your private inbox on the MG-RAST server, this area is write-only and only accessible to you. Data in the inbox cannot be read or re-exported, its sole purpose is to serve as a starting point for the pipeline.

When an upload is started it can be aborted or paused. Pausing will cause the current chunk to complete and then pause the upload. Abort will interrupt the upload immediately. A paused upload can be resumed by clicking the resume button in the upload dialog. Aborted uploads can be resumed or deleted just like other incomplete uploads by clicking the resume button in the top bar.

When an upload completes (that is not an archived file), an automatic md5 check will be calculated and the result presented to the user. In the case of an archive file uploaded, the user has to produce the checksum of the local file themselves and can paste it into a check field for validation. A note will be displayed to the user to calculate the md5sum on the uncompressed file. Archived files will be decompressed automatically.

At upload:

- Sequence files will automatically trigger sequence stats calculation
- Sequence files with calculated stats will display those stats upon selection
- Sequence files will show buttons for demultiplexing and joining of paired ends
- Barcode files will automatically show a button for demultiplexing

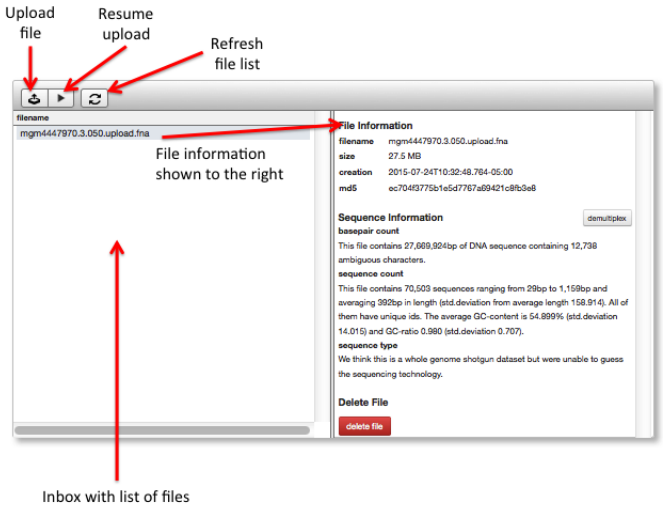


Figure 8.7: The main elements of the file browser explained. The left side pane shows a list of uploaded files. The top bar provides available actions. Users can select files to view information and whether the file passes formatting check.

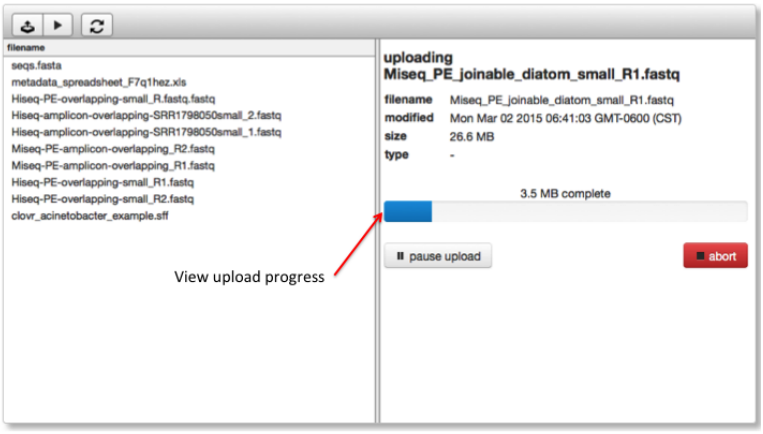


Figure 8.8: Once selected from the file browser you can start the upload and observe progress in the right side pane.

Table 8.1: Summary of upload times

Technology	Rate (bit/s)	Time for 1GB Upload
Modem 14.4 (2400 baud)	14.4 kbit/s	154 hours
ADSL Lite	1.5 Mbit/s	1.5 hours
Ethernet	10 Mbit/s	13.33 minutes
T3	44.736 Mbit/s	3 minutes
Fast Ethernet	100 Mbit/s	1.33 minutes

From the inbox data needs to be submitted to the annotation pipeline. Once files are uploaded, the inbox allows a set of operations to be performed.

Expected upload speeds

Based on observed values, upload times per 1 GB (10^9 bytes) vary from 2 minutes to over an hour, with typical times being 10 to 15 minutes. Your experience will vary depending on the speed of your connection to the internet and the quality of service in your region.

Table 1 summarizes observed upload times that might help users estimate how long the upload should take.

Frequent issues with data uploading

- Old browser version will not provide good throughput with the upload and may fail to execute the Javascript for the uploader properly. Update to the latest version of Firefox for optimal performance.
- Browser-add-ons have in several occasions blocked uploads or led to aborted uploads in the past. Disable those add-ons temporarily for the duration of the upload.
- In rare cases network devices have been presenting problems for the upload. Some institutions have not anticipated the use of the http protocol to transfer large data sets. In these cases the best option is to find another network location for the transfer.

File filters in place for uploaded files.

Since MG-RAST has been designed to work with metagenomic and metatranscriptomic datasets, there is a filter in place trying to identify datasets not suitable for MG-RAST. Those datasets will be colored red in the inbox listing and cannot be submitted. Following are the criteria for rejection:

- Protein sequences – MG-RAST is optimized to perform translation from DNA to proteins.

- Reads shorter than 75 basepairs – The gene prediction stage performance deteriorates significantly with shorter reads.
- genomes – Submissions with complete genomes or a small number of contigs are rejected as well. Here our sister service RAST at `http://rast.nmpdr.org` should be used instead of MG-RAST.
- Files that are too small (sequence data less than 1 Mbp) – Files that are too small for MG-RAST to properly function are rejected at the submission stage. The minimal size requirement is 1 megabasepair.
- Corrupted files – FASTA and FASTQ files which do not conform to the format standard, e.g. if the number of unique identifiers does not match the number of sequence records in a file, the file is considered corrupt.
- Alignments – We cannot identify proteins from sequences containing alignment information.
- Colorspace – The tool chain does not function for ABIsolid sequences in colorspace. Please translate to standard FASTA.
- rar compressed files and Zip files over 4 GB – We cannot decompress these files.

In addition we will filter at the upload stage any Word documents, Rich Text Format files, and all files without the extension .fna, .fasta, .fq, .fastq, or .sff in their name.

Note: We recommend computing an MD5 checksum and verifying that the checksum computed by MG-RAST is identical to the locally computed checksum. This is the best way to ensure data integrity.

Please note: After the actual upload is complete, the system will compute the statistics shown in Figure 8.3. Computing this information takes some time, so the statistics for your sequence files will not be visible immediately after you uploaded it. If the statistics are not displayed in a reasonable time refresh your browser page to trigger the statistics computation.

Submit data for processing

In the second step, data needs to be submitted for processing. At submission time you either add data to an existing study (or project) or create a new study. Upon successful submission, data is removed from the inbox. You will be notified via email once your submission has completed processing. In addition you can monitor the progress of your submission at the job status.

- All submitted data will stay private until the owner makes it public or shares it with another user.

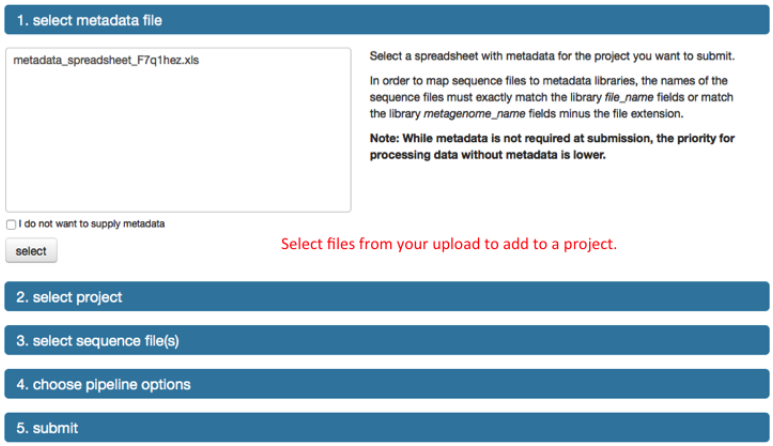


Figure 8.9: The submit page with none of the fields filled out.

- Providing metadata is required to make your data public and will increase your priority in the queue.
- The sooner you choose to make your data public, the higher your priority in the queue will be.

The submission step provides a visual aid to identify completed tasks (the bars on the page are turning from blue (open) to green (done), see Figures 8.3 and 8.3).

Progress monitoring

Once data is submitted, you can monitor its progress.

Depending on your priority (assigned based on available metadata and how public your data is) your jobs will progress through the system. Jobs that fail due to technical reasons (component failure etc.) will be restarted by MG-RAST staff.

You will receive an email once a given data set has finished processing.

Cmd-line uploader

The following upload instructions are for all file types supported by MG-RAST.

The mg-inbox command line tool allows upload of sequence and metadata files and management of the user's upload area, the inbox. In order to operate on the inbox the user has to authenticate with an MG-RAST token. The token can be retrieved from the "Account Management" –> "Manage personal preferences" –> "Web Services" –> "authentication key" page via MG-RAST Web site.

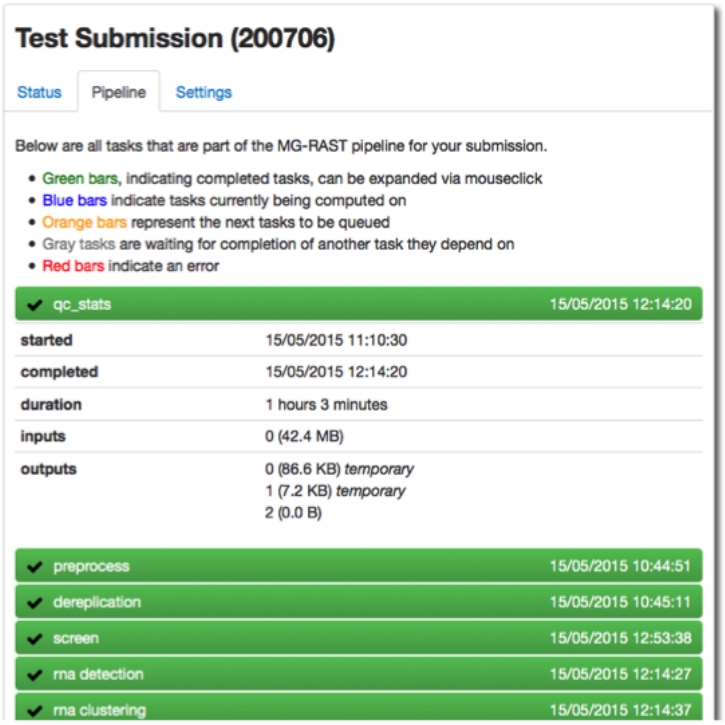


Figure 8.10: The submit page with all bars in green indicating that the respective sections have been filled out.

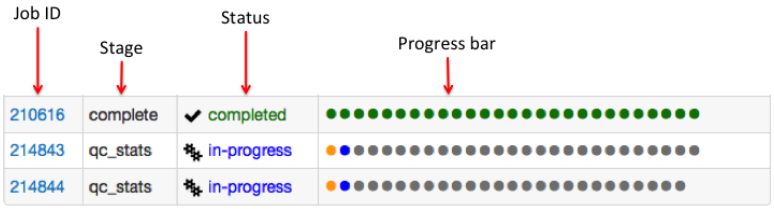


Figure 8.11: The jobs you have submitted are listed with their current status. A green dot indicates the stage has completed successfully, blue indicates that the current stage is in progress. Queued stages will produce an orange dot, green indicates a completed stage and red indicates error state. Gray dots will show for all stages waiting for other stages to complete.

1. Make sure you have python installed on your system.

   ```
   https://pip.pypa.io/en/latest/installing.html
   ```

2. Go to the directory where you have your files to upload.

3. Download the upload script

   ```
   ftp://ftp.metagenomics.anl.gov/tools/upload/mg-inbox.py
   ```

4. Checkout the help options for "mg-inbox.py". If you received an error message that you are missing certain python libraries, you will need to install them before you can run the script. To install python libraries use pip install <lib-name> to install the missing libraries. Note: The error message from running "python mg-inbox.py –help" will provide the "lib-name" you are missing. You may have a few libraries to install.

   ```
   python mg-inbox.py --help
   ```

5. First you need to login. The login option takes a user token and writes a login file after successful login. For example:

   ```
   python mg-inbox.py login --token <myToken>
   python mg-inbox.py view all
   ```

6. You can upload a file into your inbox with

   ```
   python mg-inbox.py upload <path_to_file>/<file_name>
   ```

7. If you have a compressed file to upload, supports gzip or bzip2

   ```
   python mg-inbox.py --gzip upload <path_to_file>/<gzip_file>
   python mg-inbox.py --bzip2 upload <path_to_file>/<bzip2_file>
   ```

8. If you have an archive file containing multiple files to upload, supports: .zip, .tar, .tar.gz, .tar.bz2

   ```
   python mg-inbox.py upload-archive <path_to_file>/<archive_file>
   ```

9. You can examine the content of your inbox with

   ```
   python mg-inbox.py view all
   ```

10. You can submit your sequence files from the Upload page on the MG-RAST web site (cmd-line option coming soon).

REST API uploader

The following upload instructions are for using the MG-RAST REST API with the curl program. In order to operate the API the user has to authenticate with an MG-RAST token. The token can be retrieved from the "Account Management" –> "Manage personal preferences" –> "Web Services" –> "authentication key" page via MG-RAST Web site.

We strongly suggest that you use the scripts we provide, instead of the native REST API.

1.

2. You can upload a file into your inbox with

```
curl -X POST -H "auth: <myToken>" -F "upload=@<path_to_file>/<
   file_name>" "http://api.metagenomics.anl.gov/inbox"
```

3. If you have a compressed file to upload, supports gzip or bzip2

```
curl -X POST -H "auth: <myToken>" -F "upload=@<path_to_file>/<
   gzip_file>" -F "compression=gzip" "http://api.metagenomics.
   anl.gov/inbox"
curl -X POST -H "auth: <myToken>" -F "upload=@<path_to_file>/<
   gzip_file>" -F "compression=bzip2" "http://api.metagenomics.
   anl.gov/inbox"
```

4. If you have an archive file containing multiple files to upload do the following two steps, supports: .zip, .tar, .tar.gz, .tar.bz2

   1. `curl -X POST -H "auth: <myToken>" -F "upload=@<path_to_file>/<archive_file>" "http://api.metagenomics.anl.gov/inbox"`
   2. `curl -X POST -H "auth: <myToken>" -F "format=<one of: zip, tar, tar.gz, tar.bz2>" "http://api.metagenomics.anl.gov/inbox/unpack/<uploaded_file_id>"`

Generating metadata for the submission

MG-RAST uses questionnaires to capture metadata for each project with one or more samples. Users have two options, they can download and fill out the questionnaire and then submit it or use our online editor, MetaZen `http://v3-web.metagenomics.anl.gov/Html/mgmainv3.html?mgpage=metazen`. Questionnaires are validated automatically by MG-RAST for completeness and compliance with the controlled vocabularies for certain fields.

MG-RAST has implemented the use of Minimum Information about any (X) Sequence (MIxS) [45] developed by the Genomic Standards Consortium. In addition to the minimal checklists, more detailed data can be captured in optional environmental packages.

We use simple spreadsheets to capture metadata, with a minimal number of required fields (in red in the spreadsheets) and a number of optional fields. The spreadsheet is separated into multiple tabs representing the different metadata categories. The MG-RAST metadata spreadsheet template is available on the MG-RAST upload page or at `ftp://ftp.metagenomics.anl.gov/data/misc/metadata/MGRAST_MetaData_template_1.3.xlsx`.

A filled-out version of the spreadsheet is available at `ftp://ftp.metagenomics.anl.gov/data/misc/metadata/MGRAST_MetaData_template_example.xlsx`.

In Figure 8.12 we show the template tab for project and the required field labels (in red) (in essence, your contact information). Figure 8.13 shows the various tabs in the spreadsheet.

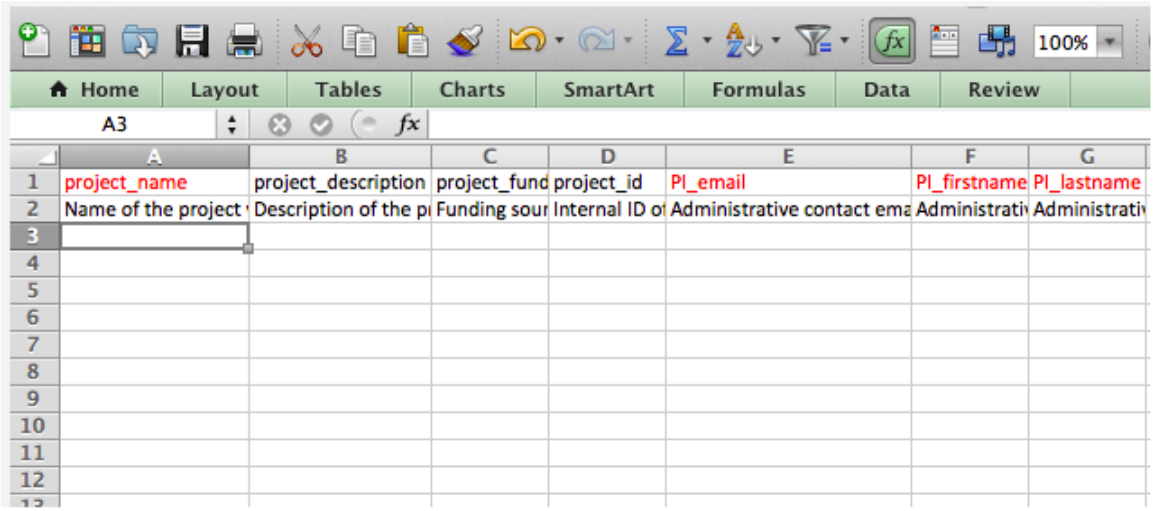
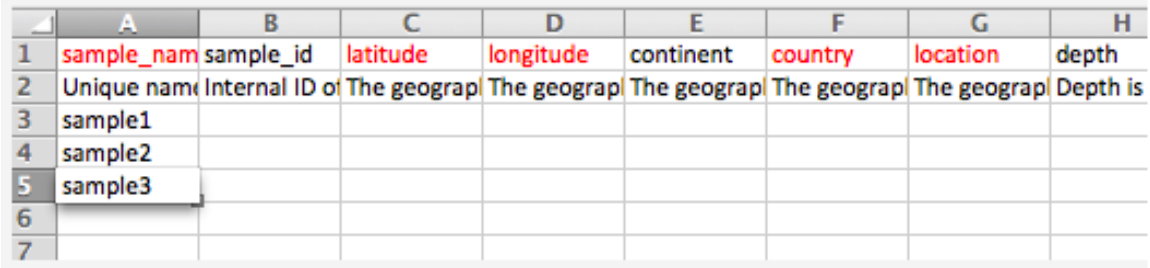


Figure 8.12: Project spreadsheet. In red are required fields. Note that the 2nd row contains information on how to fill out the form.

Figure 8.13: The various tabs in the spreadsheet. Project, sample and one of library metagenome or library mimarks survey are required.

Note: Use the third line in the spreadsheet and as shown in Figure 8.14 to enter your data. Do not attempt to alter the first two lines or delete them; they are read only. The first line contains the field labels, and the second line contains descriptions that can help explain how to fill out the fields, along with what unit to use (e.g., temperature in Celsius and distance in meters), URL for the bioportal ontology site etc..



	A	B	C	D	E	F	G	H
1	sample_nam	sample_id	latitude	longitude	continent	country	location	depth
2	Unique nam	Internal ID o	The geograp	The geograp	The geograp	The geograp	The geograp	Depth is
3	sample1							
4	sample2							
5	sample3							
6								
7								

Figure 8.14: Sample tab with 3 new samples (sample1, sample2, and sample3) added. Again red text in the first row indicates required fields. Rows 1 and 2 cannot be altered.

Required sheets You need to fill out four sheets to describe your metadata:

1. Project – This sheet has only one row, and describes a set of samples uploaded together; the other sheets have one row per sample.
2. Sample – This sheet includes either the filename or metagenome name used for matching.
3. Library – This sheet includes either the metagenome (for WGS and WXS), mimarks-survey (for 16s and amplicon) or metatranscriptome.
4. Environmental package (ep) – Several packages of suggested standard metadata are available. Choose the package that best describes your dataset (e.g., water, human-skin, soil).

Sample sheet The sample sheet requires minimal information (including the sample name) about where and when the sample was taken. Note that some fields in the spreadsheet must be filled out with terms from a controlled vocabulary or in a certain way. Country and environment (biome, feature, material) fields require entries from curated ontologies, gazetteer and environmental ontology, respectively.

Figure 8.14 shows the sample tab with three new samples (sample1, sample2, and sample3) added. Again red text in the first row indicates required fields.

Mandatory fields Five fields must be completed.

- Country – e.g. United States of America, Netherlands, Australia, Uruguay
- Latitude and longitude – e.g. [106.84517, -104.60667], [28°42.306′N, 88°24.099′W], [45.30 N, 73.35 W]
- Biome – e.g. small lake biome, urban biome, mangrove biome. This term must be one of the terms from the bioportal ontology (`http://bioportal.bioontology.org/`

ontologies/1069?p=terms&conceptid=ENVO%3A00000428). Terms that are not listed on this site are not valid.

- Feature – e.g. city, fish farm, livestock-associated habitat, marine habitat, ocean basin, microbial mat. This term must be one of the terms from the bioportal ontology. Terms that are not listed on this site are not valid.

- Material – e.g. air, dust, volcanic soil, saliva, blood, dairy product, surface water, piece of gravel. This term must be one of the terms from the bioportal ontology. Terms that are not listed on this site are not valid.

Library section The library section captures technical data on the preparation and sequencing done. You should choose the library tab to fill out ("metagenome" for shotgun sequencing, "mimarks-survey" for amplicon or "metatranscriptome") based on the type of sequencing done. These are separated as different sequencing techniques involving different metadata fields. Each row describes one library for one sample. The required fields are colored red.

The **sample_name** value in the library sheet must exactly match one of the values used in the sample sheet.

The **file_name** field holds the filename of the sequence file uploaded, or the filename to use for creating the demultiplexed file if you uploaded a multiplexed sequence file and have barcode sequences in the spreadsheet. This is used for mapping sequence files to metadata.

The **metagenome_name** field holds the name of the metagenome you are submitting. If the file_name field is empty, the metagenome_name will be used to map metadata to sequence files, in this case it would need to match the uploaded sequence filename without the file extension, e.g. a sequence file "test-sequence.fasta" would be mapped to the metadata in the row which has the metagenome_name value "test-sequence".

The **investigation_type** field is required to be "metagenome" for shotgun metagenome samples, "mimarks-survey" for amplicon studies or "metatrascriptome", reflecting which tab was filled out.

The type of sequencing instrument used is another required field. Values are, for example, Sanger, pyrosequencing, ABI-solid, 454, Illumina, assembled, other.

Again, only a limited number of fields are required. However, the more information you provide, the easier it is for you and others to understand any potential uses of your data and to understand why results appear in a particular way. It might, for example, allow understanding of specific biases caused by technology choices or sampled environments.

Environmental Package (ep) sheet You can fill out one or more environmental metadata packages. Currently we provide support for the following GSC environmental packages:

- Air
- Built Environment

- Host-associated
- Human-associated
- Human-oral
- Human-skin
- Human-gut
- Human-vaginal
- Microbial mat/biofilm
- Miscellaneous natural or artificial environment
- Plant-associated
- Sediment
- Soil
- Wastewater sludge
- Water

We strongly encourage users to submit rich metadata, but we understand the effort required in providing it. Using the environmental packages (which were designed and are used by practitioners in the respective field) should make it reasonably simple to report the essential metadata required to analyze the data. If there is no environmental package to report metadata for your specific sample, please contact MG-RAST staff: we will work with the GSC [11] to create the required questionnaire.

Using MetaZen

MG-RAST uses a simple spreadsheet with 12 mandatory terms. MetaZen designed to help you fill out your metadata spreadsheet. The metadata you provide, helps us to analyze your data more accurately and helps make MG-RAST a more useful analysis resource for everyone.

This tool will help you get started on completing your metadata spreadsheet by filling in any information that is common across all of your samples and/or libraries. This tool currently only allows users to enter one environmental package for your samples and all samples must have been sequenced by the same number of sequencing technologies with the same number of replicates. This information is entered in tab 2.

Note: If your project deviates from this convention, you must either produce multiple separate metadata spreadsheets or generate your spreadsheet and then edit the appropriate fields manually.

Metazen's online form allows users to either use an existing project, or add in new information to start a new project (Figure 8.15). Users will expand each tab and fill in their metadata information. One of the benefits to using this form is that it provides compliant ENVO terms to select from to describe your sample, without the cumbersome task of looking them up outside of MG-RAST. Figure 8.16 shows an example of this for entering in environmental information.

The first tab is for project information where you enter the project name and description as well as PI information, information for the technical contact and cross-references to different analysis tools so that your dataset can be linked across these resources.

What you enter in the second tab (sample set information) will dictate what the next tabs will be. Note: You must submit the information here before proceeding with the rest of the form. Enter the information about your set of samples. First, indicate the total number of samples in your set. Second, tell us which environmental package your samples belong to. Then, indicate how many times each of your samples was sequenced by each sequencing method. Each entry of more than zero for number of shotgun, metatranscriptome or amplicon libraries will produce an additional tab to fill out about your sample (Figure 8.17. Once you add or change information into this form you will need to press the button "show library input forms" to update subsequent tabs.

Note: It is allowable to indicate here if your samples were sequenced using more than one sequencing method.

Once the data has been entered, click on "download excel spreadsheet" to download your filled sheet. You can now use this for upload and submission to MG-RAST.

Can I upload files to my inbox through the MG-RAST API?

Yes. You can upload files to your user inbox using the MG-RAST API with the command-line tool cURL, invoked as:

```
curl -H "auth: webkey" -X POST -F "upload=@/path_to_file/metagenome.fasta"
          "http://api.metagenomics.anl.gov/1/inbox/" > curl_output.txt
```

where you need to substitute "webkey" with the unique string of text generated by MG-RAST for your account. Your webkey is valid for a limited time period and ensures that the uploads you perform from the command line are recognized as belonging to your MG-RAST account and placed in the correct inbox.

How do I handle the metadata for paired end reads?

With paired reads (e.g. R1 and R2) the reads can be merged prior to submission, in this case the metadata should only refer to the new merged reads.

You only need to include metadata for the R1 and R2 reads separately if you choose to treat the second read (R2) as a technical replicate. The mate pair merging can be handled by the Web UI by the submission script we provide in the MG-RAST tools repository.

What type of sequence files should I upload?

Your sequence data can be in FASTA, FASTQ or SFF format. These are recognized by the file name extension with valid extensions for the appropriate formats .fasta, .fna, .fastq, .fq, and .sff

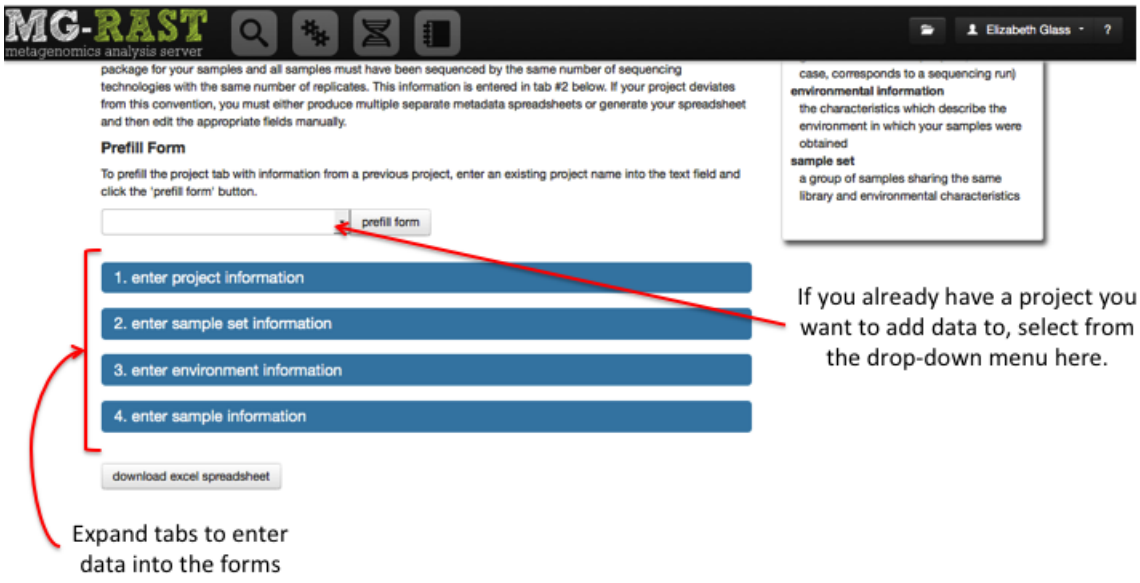


Figure 8.15: The Metazen form for filling out metadata allows users to fill in data online and add data to existing projects or start new ones. Tabs are expandable and reveal forms for the various required metadata sections.

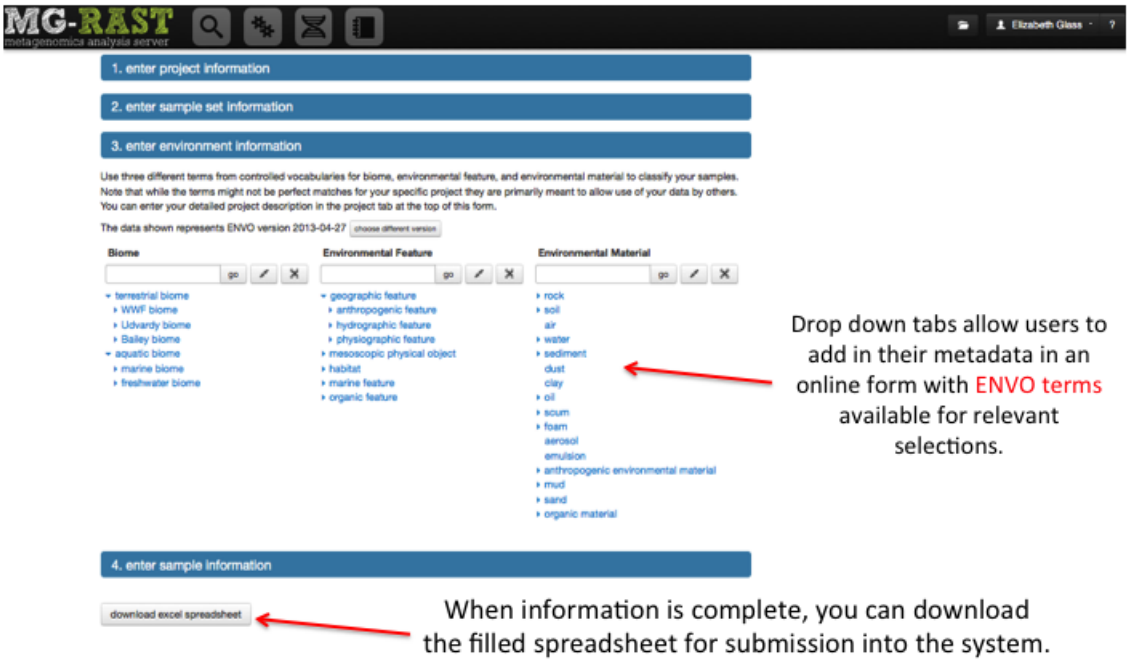


Figure 8.16: The Metazen form for filling out metadata allows users to fill in data using standard nomenclature.

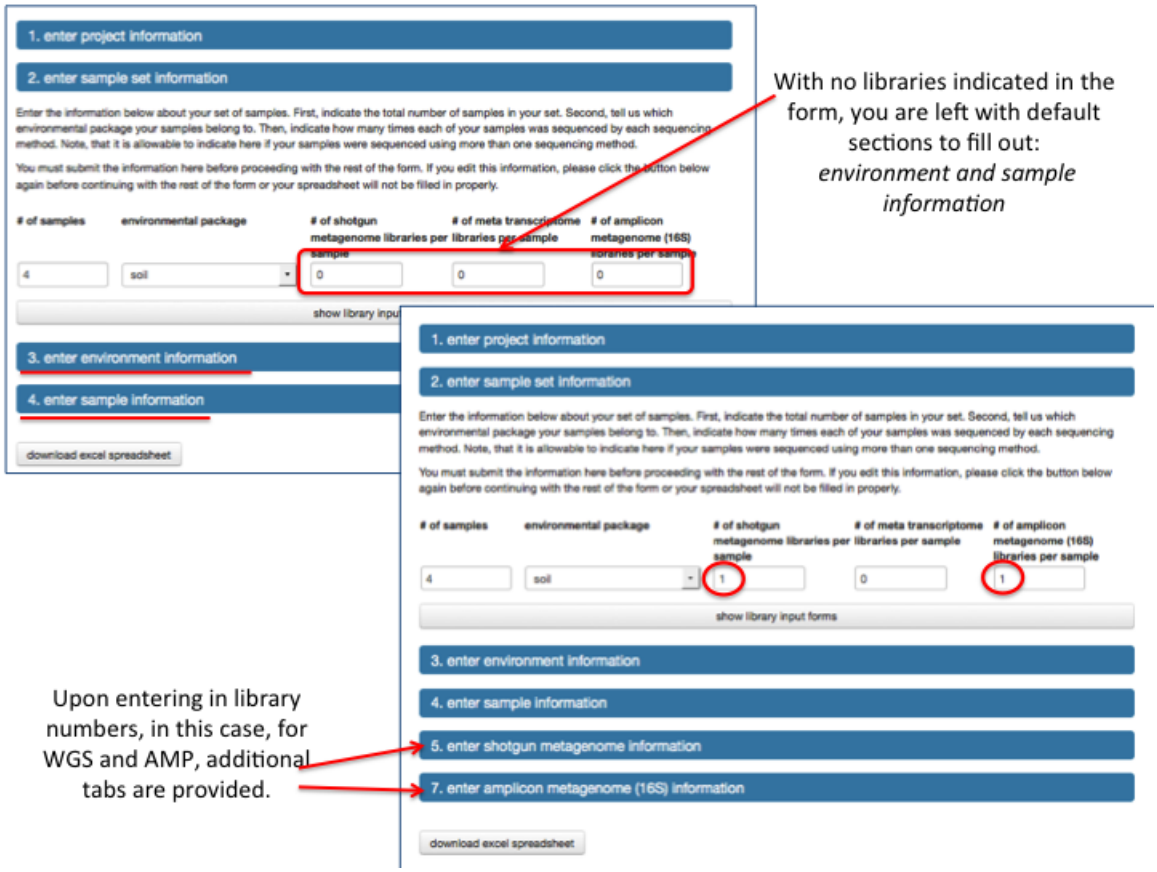


Figure 8.17: The second tab in the Metazen form must be filled out before moving further down the forms. Selecting the number of libraries (other than zero) adds forms for those libraries. Click on the "show library input forms" button to display them. If no libraries are entered, then only the default tabs for environment and sample information are provided.

and FASTA and FASTQ files need to be in plain text ASCII. Compressing large files will reduce the upload time and the chances of a failed upload, you can use gzip (.gz), bzip2 (.bz2) Zip (.zip less than 4 GB in size) as well as tar archives compressed with gzip (.tar.gz) or bzip2 (.tar.bz2), rar files are not accepted. We suggest you upload raw data (in FASTQ or SFF format) and let MG-RAST perform the quality control step, see Section 3 for details.

What type of sequence files should I NOT upload?

MG-RAST will not analyze the following:

- protein sequences,
- WGS reads <75bp,
- complete genomes,
- sequence data less than 1Mbp,
- sequences containing alignment information,
- ABIsolid sequences in colorspace,
- rar compressed files,
- Zip files over 4GB,
- Word documents,
- Rich Text Format files, and
- files without the extension .fna, .fasta, .fq, .fastq or .sff in their name.

How do I prepare my metadata for upload?

You can submit metadata for your samples during the upload/submission process. The metadata is transferred to MG-RAST in a spreadsheet in which you can enter metadata for one or more samples along with information about the project the samples should be placed in. Step one in the first section, 'Prepare data', has the empty metadata spreadsheet template available for download with the required fields labeled in red. The metadata is hierarchical with three levels, project, sample and library. There has to be a sequence file corresponding to each library entry and the sequence filename must match the library file_name fields or match the library metagenome_name fields minus extension. Once you have filled out the spreadsheet with metadata you can upload it along with the sequence files to your inbox with the MG-RAST uploader.

Table 8.2: Upload speeds for different technologies

Technology	Rate	Time for 1GB Upload
Modem 14.4 (2400 baud)	14.4 kbit/s	154 hours
ADSL Lite	1.5 Mbit/s	1.5 hours
Ethernet	10 Mbit/s	13.33 minutes
T3	44.74 Mbit/s	3 minutes
Fast Ethernet	100 Mbit/s	1.33 minutes

Will my metadata file in .xls format work OK?

Yes, the site is designed to handle .xls metadata files and we have successfully tested uploading and validating .xls files. The metadata template file we provide is a .xlsx file and that is the preferred format If you do experience problems with a .xls file being recognized, Microsoft provides a convertor to the .xslx format:

- for Mac: `http://www.microsoft.com/mac/downloads?pid=Mactopia_AddTools&fid=6B9238E1-CF69-48C4-BF2D-C4A8ACEEE520`
- for Windows: `http://www.microsoft.com/en-us/download/details.aspx?id=3`

How are the projects listed on the upload page during submission selected?

During the submission process, you can choose to place the new datasets in an existing project. All the projects you have write access to will be listed for selection, this includes all the projects you own as well as projects owned by other users for which you have been granted write access. You can also specify a particular project from this list in the metadata template file or create a new project for your dataset(s) by typing in the name.

How much time will it take to upload my data to MG-RAST?

Based on observed values, upload times per 1GB (10^9 bytes) vary from 2 minutes to over an hour with typical times being 10 to 15 minutes. Your experience will vary depending on the speed of your connection to the internet and the quality of service in your region. The fastest times that could be expected for the technology you are using is listed in table 8.2. In practice the time taken will be more than indicated in the table.

Do I need to compress my files before uploading to MG-RAST?

It is not required that you compress your files before uploading to MG-RAST, but it is highly recommended.

Compressing your sequence data using Zip or gzip before it is uploaded will reduce the time required for the upload. The compression rate depends on the nature of the sequences, typical compression rates for uploaded sequence data that we have observed is between 30-35%. This means the time taken for the upload may be reduced by a third or even more. On a slow connection where uploading 1GB takes over an hour this could be a considerable reduction in time. In addition, the shortened time will also reduce the chance of a failed upload if something goes wrong.

What does the "Join paired-ends" function do?

The 'Join paired-ends' function on the Upload page allows users to merge two fastq files which represent paired end reads from the same sequencing run. The fastq-join utility (`http://code.google.com/p/ea-utils/wiki/FastqJoin`) is used to merge mate-pairs with a minimum overlap setting of 8bp and a maximum difference of 10% (parameters: -m 8 -p 10). There is an option to retain or remove the pairs which do not overlap—the 'remove' option drops paired reads for which no overlap is found and the 'retain' option will keep non-overlapping paired reads in your output file as separate individual (non-joined) sequences. There is also an option to include an index file (if you have one) that contains the barcode for each mate-pair. If this file is included, the barcodes will be reverse-complemented and then prepended to the output sequences.

What does the "assembled" pipeline option do?

The "assembled" pipeline option allows users to submit sequence data under a slightly altered analysis pipeline that is more appropriate for assembled sequences. Your assembled contigs should be uploaded in FASTA format and should include the abundance of each contig in your dataset with the following format:

```
>sequence_number_1_[cov=2]
CTAGCGCACATAGCATTCAGCGTAGCAGTCACTAGTACGTAGTACGTACC...
>sequence_number_2_[cov=4]
ACGTAGCTCACTCCAGTAGCAGGTACGTCGAGAAGACGTCTAGTCATCAT...
```

The abundance information must be appended without spaces to the end of the sequence name (also without whitespace) in the format

```
_[cov=n]
```

where 'n' is the coverage or abundance of each contig.

Can I use the coverage information in my Velvet sequence file?

Yes, coverage information can be included in the header lines of FASTA-formatted files, for the exact format see the FAQ entry on the assembled pipeline.
The following unix command:

```
cat contigs.fa |
   sed 's/_cov_\([0-9]*\).[0-9]*/_[cov=\1]/;' > Assembly-formatted-for-MGRAST.fa
```

should transform Velvet's default FASTA output into MG-RAST's preferred output.
Adding one more term:

```
cat contigs.fa |
     sed 's/_cov_\([0-9]*\).[0-9]*/_[cov=\1]/;
          s/NODE/Assembly-and-sample-name/' > Assembly-formatted-for-MGRAST.fa
```

will give the contigs better names than NODE_4_etc., substitute your information for 'Assembly-and-sample-name'.

8.4 Job processing

How long does it take to analyze a metagenome?

The answer depends on three factors:

- the priority assigned to your dataset,
- the size of your dataset, and
- the current server load.

In practice the time taken will range between a few hours and a week.

How is the job processing priority assigned?

MG-RAST assigns a priority to each dataset which will influence the order in which datasets are selected for processing as well as the processing speed for individual stages in the analysis pipeline. The priority of processing a dataset is based on its usefulness to the scientific community and is estimated using a combination of the amount of metadata supplied and the length of time before the dataset will be made public. The highest priority is given to datasets with complete metadata that will be made public immediately.

8.5 Analysis pipeline

How is the dereplication step performed?

The dereplication step is performed to remove replicates which can be produced during sequencing. MG-RAST identifies two reads as replicates if they have 100% identity over the first 50 basepairs. This step is optional and you should skip it for amplicon data.

What does the "demultiplex" function do?

The 'demultiplex' function on the Upload page gives users the ability to demultiplex a multiplexed sequence file. The user enters the multiplexed sequence file and a bar codes file. A process is then run that separates out sequences, based upon bar codes, into separate sequence files. The separate sequence files are then turned into separate jobs in MG-RAST upon submission..

How is the job processing priority assigned?

MG-RAST assigns a priority to each dataset which will influence the order in which datasets are selected for processing as well as the processing speed for individual stages in the analysis pipeline. The priority of processing a dataset is based on its usefulness to the scientific community and is estimated using a combination of the amount of metadata supplied and the length of time before the dataset will be made public. The highest priority is given to datasets with complete metadata that will be made public immediately.

8.6 Analysis results

What annotations does MG-RAST display?

At the moment, the annotations provided by MG-RAST are annotations produced by the MG-RAST v3.2 analysis pipeline. Different pipelines (and different pipeline strategies) may produce different results, and the results of different annotation strategies are notoriously different to reconcile. Some users have reported and published using annotations that differ from those produced by MG-RAST; we provide the MG-RAST annotations. While in theory the various annotation tools and approaches do similar things (annotating reads based on similarity to sequences in the public databases), the various approaches can provide significantly different descriptions, particularly at the species level.

Why don't the numbers of annotations add up to the number of reads?

See Section 4.6.

Is the alignment length in amino acids or in nucleotides?

For the protein similarities against the protein databases, alignment length is in amino acids. For the nucleic acid similarities against the RNA databases, the alignment length is in nucleotides.

Why am I seeing RNA similarities in my shotgun dataset?

MG-RAST identifies sequences similar to known RNA sequences in shotgun data and annotates them in addition to providing protein function annotation and protein-derived taxonomic annotation. Your mileage may vary.

Why am I seeing protein similarities in my RNA dataset?

These are called "false positives". We fall back on human judgment when computers give results that don't make sense.

Why don't you suppress the false positives?

If we suppress protein similarities when we think a dataset is RNA, we will sometimes make mistakes, and suppress protein similarities on a dataset that is, say, a metatranscriptome, for which the protein similarities are the principal objective. These might be called "false negatives", and our users don't want that.

What do all those symbols in the similarities table mean?

The MG-RAST system was designed to annotate large datasets; the similarities output is designed for the convenience of the MG-RAST system and not the end user. MG-RAST uses 32-character symbols like this `28614b98db4f4efc13b8b20b21ee9b95` (md5 protein identifiers) as the labels for protein sequences, regardless of database.

Can I run a BLAST search against all public metagenomes?

No. Such a search is too computationally expensive. But you can find public metagenomes that contain proteins that hit your favorite sequence from the Search page.

8.7 Download

Where is the table of reads with the annotation for each read?

MG-RAST versions 1 and 2 had this type of output, but MG-RAST v3 does not. MG-RAST version 3 has been optimized for large (Gbase+) datasets, and per-read annotation for large datasets is extremely bulky and difficult to interpret. The per-read annotations are not stored in a file on the server, but can be downloaded using the MG-RAST API.

Where can I download the results of the metagenome analysis?

Every completed MG-RAST dataset has a page where you can download the files produced by the different stages of the analysis, click on the link on the metagenome overview page. Datasets which have been made public have links to an ftp site at the top of this download page where you can access additional information.

How do I download everything?

As of April 2014 we have over 7 x 10^{12} bases of public sequence data, so you might want to consider if all the data is really what you need to answer your research question.

Public datasets, including sequence data and annotation data products, are available from our API.

8.8 Privacy

Who can access my uploaded data?

Your uploaded data will remain confidential as long as you do not share it with other users. You will have the ability to share the data with individuals or publish it to the MG-RAST community.

Will my private jobs ever be deleted?

Currently MG-RAST policy is that private jobs will not be deleted for 120 days after submission as mentioned in the Terms of Service. We do not enforce the 120 days as a strict deadline and your private jobs theoretically can remain in the system indefinitely, we will not delete your job without giving you ample warning. You are strongly encouraged to make your data public once it has been published to ensure it will never be considered for deletion.

How do I make a job public?

There is a 'make public' button on the metagenome overview page accessed by clicking on the MG-RAST ID on the metagenome browse page. Making a dataset public requires entering the relevant metadata without which the dataset is of limited use. The website will lead you through the process of entering metadata (if you have not done so earlier) and making the dataset public.

Will my public jobs ever be deleted?

No, we will not delete MG-RAST jobs which have been made public.

8.9 Webkey

What is an MG-RAST webkey?

See Section 8.2.

How do I generate a webkey?

See Section 8.2.

Chapter 9

Putting It All in Perspective

We have described MG-RAST, a community resource for the analysis of metagenomic sequence data. We have developed a new pipeline and environment for automated analysis of shotgun metagenomic data, as well as a series of interactive tools for comparative analysis. The pipeline is also being used for analyzing metatranscriptome data as well as amplicon data of various kinds. This service is being used by thousands of users worldwide, many contributing their data and analysis results to the community. We believe that community resources such as MG-RAST will fill a vital role in the bioinformatics ecosystem in the years to come.

9.1 MG-RAST, a community resource

MG-RAST has become a community clearinghouse for metagenomic data and analysis, with over 12,000 public datasets that can be freely used. Because analysis was performed in a uniform way, these datasets can serve as building blocks for new comparative analysis; so long as new datasets are analyzed similarly, results are robustly comparable between new and old dataset analysis. These datasets (and the resulting analysis data products) are made available for download and reuse as well.

Community resources like MG-RAST provide a clear value proposition to the metagenomics community. First, it enables low-cost meta-analysis. Users utilize the data products in MG-RAST as a basis for comparison without the need to reanalyze every dataset used in their studies. The high computational cost of analysis [44] makes precomputation a prerequisite for large-scale meta-analyses. In 2001, Angiuli et al. [1] determined the real currency cost of reanalysis for the over 12,000 datasets openly available on MG-RAST to be in excess of $30 million if Amazon's EC2 platform is used. This figure does not consider the 66,000 private datasets that have been analyzed with MG-RAST.

Second, it provides incentives to the community to adopt standards, in terms of both metadata and analysis approaches. Without this standardization, data products are not readily reusable, and

computational costs quickly become unsustainable. We are not arguing that a single analysis is necessarily suitable for all users; rather, we are pointing out that if one particular type of analysis is run for all datasets, the results can be efficiently reused, amortizing costs. Open access to data and analyses foster community interactions that make it easier for researchers' efforts to achieve consensus with respect to establishing best practices as well as identifying methods and analyses that could provide misleading results.

Third, community resources drive increased efficiency and computational performance. Community resources consolidate the demand for analysis resources sufficiently to drive innovation in algorithms and approaches. Because of this demand, the MG-RAST team has needed to scale the efficiency of their pipeline by a factor of nearly 1,000 over the past four years. This drive has caused improvements in gene calling, clustering, and sequence quality analysis, as well as many other areas. In less specialized groups with less extreme computational needs, this sort of efficiency gain would be difficult to achieve. Moreover, the large quantities of datasets that flow through the system have forced the hardening of the pipeline against a large variety of sequence pathology types that would not be readily observed in smaller systems.

We believe that our experiences in the design and operation of MG-RAST are representative of bioinformatics as a whole. The community resource model is critical if we are to benefit from the exponential growth in sequence data. This data has the potential to enable new insights into the world around us, but only if we can analyze it effectively. Only because of this approach have we been able to scale to the demands of our users effectively, analyzing over 200 billion sequences thus far.

We note that scaling to the required throughput by adding hardware to the system or simply renting time using an unoptimized pipeline on. For example, Amazon's EC2 machine would not be economically feasible. The real currency cost on EC2 for the data currently analyzed in MG-RAST (26 terabasepairs) would be in excess of $100 million using an unoptimized workflow such as CLOVR [1].

All of MG-RAST is open source and available on `https://github.com/MG-RAST`.

9.2 Future Work

While MG-RAST v3 is a substantial improvement over prior systems, much work remains to be done. Dataset sizes continue to increase at an exponential pace. Keeping up with this change remains a top priority, as metagenomics users continue to benefit from increased resolution of microbial communities. Upcoming versions of MG-RAST will include (1) mechanisms for speeding pipeline up using data reduction strategies that are biologically motivated; (2) opening up the data ecosystem via an API that will enable third-party development and enhancements; (3) providing distributed compute capabilities using user-provided resources; and (4) providing virtual integration of local datasets to allow comparison between local data and shared data without requiring

full integration.

9.2.1 Roadmap

We maintain a rough roadmap for future version of MG-RAST.

version 3.5

- provide a web services API
- develop an R client
- provide alpha version of MG-RAST remote compute client (using VMs)

4.0

- provide reviewer access tokens
- consolidate all SQL onto PostGRES
- provide beta version of MG-RAST remote compute client (using VMs)
- include IPython-based notebooks for analysis
- use AWE for all computations and SHOCK for all pipeline storage
- provide multi-metagenome recruitment plot
- convert all file access to SHOCK

version 4.x

- rewrite web interface to support many browsers
- provide BAM upload support
- provide BAM download support
- provide variation study support

version 5.0

- provide federated SHOCK system
- provide an assembly based pipeline

Acknowledgments

This project is funded by the NIH grant R01AI123037 and by NSF grant 1645609

This work used the Magellan machine (U.S.Department of Energy, Office of Science, Advanced Scientific Computing Research, under contract DE-AC02-06CH11357) at Argonne National Laboratory, and the PADS resource (National Science Foundation grant OCI-0821678) at the Argonne National Laboratory/University of Chicago Computation Institute.

In the past the following sources contributed to MG-RAST development:

- U.S. Dept. of Energy under Contract DE-AC02-06CH11357
- Sloan Foundation (SLOAN #2010-12),
- NIH NIAID (HHSN272200900040C),
- NIH Roadmap HMP program (1UH2DK083993-01).

Appendices

Appendix A

The downloadable files for each data set

Uploaded File(s) DNA (4465825.3.25422.fna)

Uploaded nucleotide sequence data in FASTA format. Preprocessing

Depending on the options chosen, the preprocessing step filters sequences based on length, number of ambiguous bases and quality values if available.

passed, DNA (4465825.3.100.preprocess.passed.fna)

A FASTA formatted file containing the sequences which were accepted and will be passed on to the next stage of the analysis pipeline.

removed, DNA (4465825.3.100.preprocess.removed.fna)

A FASTA formatted file containing the sequences which were rejected and will not be passed on to the next stage of the analysis pipeline. Dereplication

The optional dereplication step removes redundant "technical replicate" sequences from the metagenomic sample. Technical replicates are identified by binning reads with identical first 50 base-pairs. One copy of each 50-base-pair identical bin is retained.

passed, DNA (4465825.3.150.dereplication.passed.fna)

A FASTA formatted file containing one sequence from each bin which will be passed on to the next stage of the analysis pipeline.

removed, DNA (4465825.3.150.dereplication.removed.fna)

A FASTA formatted file containing the sequences which were identified as technical replicates and will not be passed on to the next stage of the analysis pipeline. Screening

The optional screening step screens reads against model organisms using bowtie to remove reads which are similar to the genome of the selected species.

passed, DNA (4465825.3.299.screen.passed.fna)

A FASTA formatted file containing the reads which which had no similarity to the selected genome and will be passed on to the next stage of the analysis pipeline. Prediction of protein coding sequences

Coding regions within the sequences are predicted using FragGeneScan, an ab-initio prokary-

otic gene calling algorithm. Using a hidden Markov model for coding regions and non-coding regions, this step identifies the most likely reading frame and translates nucleotide sequences into amino acids sequences. The predicted coding regions, possibly more than one per fragment, are called features.

coding, Protein (4465825.3.350.genecalling.coding.faa)

A amino-acid sequence FASTA formatted file containing the translations of the predicted coding regions.

coding, DNA (4465825.3.350.genecalling.coding.fna)

A nucleotide sequence FASTA formatted file containing the predicted coding regions. RNA Clustering

Sequences from step 2 (before dereplication) are pre-screened for at least 60% identity to ribosomal sequences and then clustered at 97% identity using UCLUST. These clusters are checked for similarity against the ribosomal RNA databases (Greengenes [9], LSU and SSU from [29], and RDP [7]).

rna97, DNA (4465825.3.440.cluster.rna97.fna)

A FASTA formatted file containing sequences that have at least 60% identity to ribosomal sequences and are checked for RNA similarity.

rna97, Cluster (4465825.3.440.cluster.rna97.mapping)

A tab-delimited file that identifies the sequence clusters and the sequences that comprise them.

The columns making up each line in this file are:

Cluster ID, e.g. rna97_998

Representative read ID, e.g. 11909294

List of IDs for other reads in the cluster, e.g. 11898451,11944918

List of percentage identities to the representative read sequence, e.g. 97.5%,100.0%

RNA similarities

The two files labelled "expand" are comma- and semicolon- delimited files that provide the mappings from md5s to function and md5s to taxonomy:

annotated, Sims (4465825.3.450.rna.expand.lca)

annotated, Sims (4465825.3.450.rna.expand.rna)

Packaged results of the blat search against all the DNA databases with MD5 value of the database sequence hit followed by sequence or cluster ID, similarity information, annotation, organism, database name.

raw, Sims (4465825.3.450.rna.sims)

This is the similarity output from BLAT. This includes the identifier for the query which is either the FASTA id or the cluster ID, and the internal identifier for the sequence that it hits.

The fields are in BLAST m8 format:

Query id (either fasta ID or cluster ID), e.g. 11847922

Hit id, e.g. lcl—501336051b4d5d412fb84afe8b7fdd87
percentage identity, e.g. 100.00
alignment length, e.g. 107
number of mismatches, e.g. 0
number of gap openings, e.g. 0
q.start, e.g. 1
q.end, e.g. 107
s.start, e.g. 1262
s.end, e.g. 1156
e-value, e.g. 1.7e-54
score in bits, e.g. 210.0

filtered, Sims (15:04 4465825.3.450.rna.sims.filter)

This is a filtered version of the raw Sims file above that removes all but the best hit for each data source. Gene Clustering

Protein coding sequences are clustered at 80% identity with UCLUST. This process does not remove any sequences but instead makes the similarity search step easier. Following the search, the original reads are loaded into MG-RAST for retrieval on-demand.

aa90, Protein (4465825.3.550.cluster.aa90.faa)

An amino acid sequence FASTA formatted file containing the translations of one sequence from each cluster (by cluster ids starting with aa90_) and all the unclustered (singleton) sequences with the original sequence ID.

aa90, Cluster (4465825.3.550.cluster.aa90.mapping)

A tab-separated file in which each line describes a single cluster.

The fields are:

Cluster ID, e.g. aa90_3270
protein coding sequence ID including hit location and strand, e.g. 11954908_1_121_+
additional sequence ids including hit location and strand, e.g. 11898451_1_119_+,11944918_19_121_+
sequence % identities, e.g. 94.9%,97.0%

Protein similarities

annotated, Sims (4465825.3.650.superblat.expand.lca)

The expand.lca file decodes the MD5 to the taxonomic classification it is annotated with.

The format is:

md5(s), e.g. cf036dfa9cdde3a8a4c09d7fabfd9ba5;1e538305b8319dab322b8f28da82e0a1
feature id (for singletons) or cluster id of hit including hit location and strand, e.g. 11857921_1_101_-

alignment %, e.g. 70.97;70.97
alignment length, e.g. 31;31
E-value, e.g. 7.5e-05;7.5e-05

Taxonomic string, e.g. Bacteria;Actinobacteria;Actinobacteria (class);Coriobacteriales;Coriobacteriaceae;S
exigua;-

annotated, Sims (4465825.3.650.superblat.expand.protein)

Packaged results of the blat search against all the protein databases with MD5 value of the database sequence hit followed by sequence or cluster ID, similarity information, functional annotation, organism, database name.

Format is:

md5 (identifier for the database hit), e.g. 88848aa7224ca2f3ac117e7953edd2d9

feature id (for singletons) or cluster ID for the query, e.g. aa90_22837

alignment % identity, e.g. 76.47

alignment length, e.g. 34

E-value, e.g. 1.3e-06

protein functional label, e.g. SsrA-binding protein

Species name associated with best protein hit, e.g. Prevotella bergensis DSM 17361 RefSeq 585502

raw, Sims (4465825.3.650.superblat.sims)

Blat output with sequence or cluster ID, md5 value for the sequence in the database and similarity information.

filtered, Sims (4465825.3.650.superblat.sims.filter)

Blat output filtered to take only the best hit from each data source.

Appendix B

Terms of Service

- MG-RAST is a web-based computational metagenome analysis service provided on a best-effort basis. We strive to provide correct analysis, privacy, but can not guarantee correctness of results, integrity of data or privacy. That being said, we are not responsible for any HIPPA regulations regarding human samples uploaded by users. We will try to provide as much speed as possible and will try to inform users about wait times. We will inform users about changes to the system and the underlying data.
- We reserve the right to delete non public data sets after 120 days.
- We reserve the right to reject data set that are not complying with the purpose of MG-RAST.
- We reserve the right to perform additional data analysis (e.g. search for novel sequence errors to improve our sequence quality detection, clustering to improve sequence similarity searches etc.) AND in certain cases utilize the results. We will NOT release user provided data without consent and or publish on user data before the user.
- User acknowledges the restrictions stated about and will cite MG-RAST when reporting on their work.
- User acknowledges the fact that data sharing on MG-RAST is meant as a pre-publication mechanism and we strongly encourage users to make data publicly accessible in MG-RAST once published in a journal (or after 120 days).
- User acknowledges that data (including metadata) provided is a) correct and b) user either owns the data or has the permission of the owner to upload data and or publish data on MG-RAST.
- We reserve the right to curate and update public meta data.

- We reserve the right at any time to modify this agreement. Such modifications and additional terms and conditions will be effective immediately and incorporated into this agreement. MG-RAST will make a reasonable effort to contact users via email of any changes and your continued use of MG-RAST will be deemed acceptance thereof.

Appendix C

Tools and data used by MG-RAST

The MG-RAST team is happy to acknowledge the use of the following great software and data products: Databases

MG-RAST uses a number of protein and ribosomal RNA databases integrated into the M5nr [41] (Wilke et al, BMC Bioinformatics 2012. Vol 13, No. 151) non-redundant database using the M5nr tools.

C.1 Databases

C.1.1 Protein databases

- The SEED [28] (Overbeek et al., NAR, 2005, Vol. 33, Issue 17)
- GenBank [3] (Benson et al., NAR, 2011, Vol. 39, Database issue)
- RefSeq [30] (Pruitt et al., NAR, 2009, Vol. 37, Database issue)
- IMG/M (Markowitz et al., NAR, 2008, Vol. 36, Database issue)
- UniProt [23] (Apweiler et al., NAR, 2011, Vol. 39, Database issue)
- eggNOGG [17] (Muller et al., NAR, 2010, Vol. 38, Database issue)
- KEGG [18] (Kanehisa et al., NAR, 2008, Vol. 36, Database issue)
- PATRIC [35] (Gillespie et al., Infect. Immun., 2011, Vol. 79, no. 11)

C.1.2 Ribosomal RNA databases

- greengenes [9] (DeSantis et al., Appl Environ Microbiol., 2006, Vol. 72, no. 7)

- SILVA [29] (Pruesse et al., NAR, 2007, Vol. 35, issue 21)
- RDP [7] (Cole et al., NAR, 2009, Vol. 37, Database issue)

C.2 Software

C.2.1 Bioinformatics codes

- FragGeneScan [33] (Rho et al, NAR, 2010, Vol. 38, issue 20)
- BLAT [20] (J. Kent, Genome Res, 2002, Vol. 12, No. 4)
- QIIME [6] (Caporaso et al, Nature Methods, 2010, Vol. 7, No. 5) (we also use uclust that is part of QIIME)
- Biopython
- Bowtie [21] (Langmead et al., Genome Biol. 2009, Vol 10, issue 3)
- sff_extract, Jose Blanca and Joaquin Cañizares
- Dynamic Trim, part of SolexaQA, [8] (Cox et al., BMC Bioinformatics, 2011, Vol. 11, 485)
- FastqJoin

C.2.2 Web/UI tools

- Krona [27] (Ondov et. al. BMC Bioinformatics, 2011, Vol. 12, 385)
- Raphael JavaScript Library (Dmitry Baranovskiy)
- jQuery
- Circos (Krzywinski et al., Genome Res. 2009, Vol. 19)
- cURL

C.2.3 Behind the scenes

- Perl
- Python
- R

- Go
- Google's V8 JavaScript engine
- Node.js
- nginx
- OpenStack

List of Figures

Bibliography

[1] S.V. Angiuoli, M. Matalka, A. Gussman, K. Galens, M. Vangala, D.R. Riley, C. Arze, J.R. White, O. White, and W.F. Fricke. CloVR: a virtual machine for automated and portable sequence analysis from the desktop using cloud computing. *BMC Bioinformatics*, 12:356, 2011.

[2] Ramy Aziz, Daniela Bartels, Aaron Best, Matthew DeJongh, Terrence Disz, Robert Edwards, Kevin Formsma, Svetlana Gerdes, Elizabeth Glass, Michael Kubal, Folker Meyer, Gary Olsen, Robert Olson, Andrei Osterman, Ross Overbeek, Leslie McNeil, Daniel Paarmann, Tobias Paczian, Bruce Parrello, Gordon Pusch, Claudia Reich, Rick Stevens, Olga Vassieva, Veronika Vonstein, Andreas Wilke, and Olga Zagnitko. The RAST Server: Rapid Annotations using Subsystems Technology. *BMC Genomics*, 9(1):75, 2008.

[3] D.A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and E.W. Sayers. GenBank. *Nucleic Acids Res*, 41(Database issue):D36–42, 2013.

[4] OpenMP Architecture Review Board. OpenMP Application Program Interface Version 3.1. This document is available as a PDF from `http://www.openmp.org/mp-documents/OpenMP3.1.pdf`, July 2011.

[5] A. Bolotin, B. Quinquis, A. Sorokin, and S.D. Ehrlich. Clustered regularly interspaced short palindrome repeats (CRISPRs) have spacers of extrachromosomal origin. *Microbiology*, 151(Pt 8):2551–61, 2005.

[6] J.G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F.D. Bushman, E.K. Costello, N. Fierer, A.G. Pena, J.K. Goodrich, J.I. Gordon, G.A. Huttley, S.T. Kelley, D. Knights, J.E. Koenig, R.E. Ley, C.A. Lozupone, D. McDonald, B.D. Muegge, M. Pirrung, J. Reeder, J.R. Sevinsky, P.J. Turnbaugh, W.A. Walters, J. Widmann, T. Yatsunenko, J. Zaneveld, and R. Knight. QIIME allows analysis of high-throughput community sequencing data. *Nat Methods*, 7(5):335–6, 2010.

[7] J.R. Cole, B. Chai, T.L. Marsh, R.J. Farris, Q. Wang, S.A. Kulam, S. Chandra, D.M. McGarrell, T.M. Schmidt, G.M. Garrity, J.M. Tiedje, and Ribosomal Database Project. The

Ribosomal Database Project (RDP-II): previewing a new autoaligner that allows regular updates and the new prokaryotic taxonomy. *Nucleic acids research*, 31(1):442–443, January 2003.

[8] M.P. Cox, D. A. Peterson, and P.J. Biggs. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics*, 11:485, 2010.

[9] T.Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E.L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G.L. Andersen. Greengenes, a Chimera-Checked16S rRNA Gene Database and Workbench Compatible with ARB. *Appl. Environ. Microbiol.*, 72(7):5069–5072, July 2006.

[10] R.C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–1, 2010.

[11] D. Field, L. Amaral-Zettler, G. Cochrane, J.R. Cole, P. Dawyndt, G.M. Garrity, J. Gilbert, F.O. Glöckner, L. Hirschman, and I. Karsch-Mizrachi. The Genomic Standards Consortium. *PLOS Biology*, 9(6):e1001088, 2011.

[12] Wolfgang Gerlach, Wei Tang, Kevin Keegan, Travis Harrison, Andreas Wilke, Jared Bischof, Mark D'Souza, Scott Devoid, Daniel Murphy-Olson, Narayan Desai, and Folker Meyer. Skyport: Container-based execution environment management for multi-cloud scientific workflows. In *Proceedings of the 5th International Workshop on Data-Intensive Computing in the Clouds*, DataCloud '14, pages 25–32, Piscataway, NJ, USA, 2014. IEEE Press.

[13] V. Gomez-Alvarez, T.K. Teal, and T.M. Schmidt. Systematic artifacts in metagenomes from complex microbial communities. *ISME J*, 3(11):1314–7, 2009.

[14] S.M. Huse, J.A. Huber, H.G. Morrison, M.L. Sogin, and D.M. Welch. Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol*, 8(7):R143, 2007.

[15] D.H. Huson, A.F. Auch, J. Qi, and S.C. Schuster. MEGAN analysis of metagenomic data. *Genome Res*, 17(3):377–86, 2007.

[16] National Human Genome Research Institute. Cost per raw megabase of dna sequence, 2012. This document is available on the NHGRI website at `http://www.genome.gov/sequencingcosts/`.

[17] L.J. Jensen, P. Julien, M. Kuhn, C. von Mering, J. Muller, T. Doerks, and P. Bork. eggNOG: automated construction and annotation of orthologous groups of genes. *Nucleic Acids Res*, 36(Database issue):D250–4, 2008.

[18] M. Kanehisa. The KEGG database. *Novartis Found Symp*, 247:91–101; discussion 101–3, 119–28, 244–52, 2002.

[19] K.P. Keegan, W.L. Trimble, J. Wilkening, A. Wilke, T. Harrison, M. D'Souza, and F. Meyer. A platform-independent method for detecting errors in metagenomic sequencing data: DRISEE. *PLOS Comput Biol*, 8(6):e1002541, 2012.

[20] W.J. Kent. BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–64, 2002.

[21] B. Langmead, C. Trapnell, M. Pop, and S.L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[22] Nicholas J. Loman, Raju V. Misra, Timothy J. Dallman, Chrystala Constantinidou, Saheer E Gharbia, John Wain, and Mark J. Pallen. Performance comparison of benchtop high-throughput sequencing platforms. *Nature Biotechnology*, 30(5):434–439, 2012.

[23] Michele Magrane and UniProt Consortium. UniProt Knowledgebase: a hub of integrated protein data. *Database: the journal of biological databases and curation*, 2011, January 2011.

[24] V.M. Markowitz, N.N. Ivanova, E. Szeto, K. Palaniappan, K. Chu, D. Dalevi, I. M. Chen, Y. Grechkin, I. Dubchak, I. Anderson, A. Lykidis, K. Mavromatis, P. Hugenholtz, and N.C. Kyrpides. IMG/M: a data management and analysis system for metagenomes. *Nucleic Acids Res*, 36(Database issue):D534–8, 2008.

[25] D. McDonald, J.C. Clemente, J. Kuczynski, J. Rideout, J. Stombaugh, D. Wendel, A. Wilke, S. Huse, J. Hufnagle, and F. Meyer. The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *Gigascience*, 2012.

[26] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E.M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, J. Wilkening, and R.A. Edwards. The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9(1):386, 2008.

[27] B.D. Ondov, N.H. Bergman, and A.M. Phillippy. Interactive metagenomic visualization in a web browser. *BMC Bioinformatics*, 12:385, 2011.

[28] R. Overbeek, T. Begley, R.M. Butler, J.V. Choudhuri, N. Diaz, H.-Y. Chuang, M. Cohoon, V. de Crécy-Lagard, T. Disz, R Edwards, M Fonstein, E.D. Frank, S. Gerdes, E.M. Glass, A. Goesmann, L. Krause, B. Linke, A.C. McHardy, F. Meyer, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, M. Kubal, N. Larsen, H. Neuweger, C. Rückert, G.J. Olsen, R. Olson, A. Osterman, V. Portnoy, G.D. Pusch, D.A. Rodionov, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein. The Subsystems Approach to Genome Annotation and its Use in the Project to Annotate 1000 Genomes. *Nucleic Acids Res*, 33(17), 2005.

[29] Elmar Pruesse, Christian Quast, Katrin Knittel, Bernhard M. Fuchs, Wolfgang Ludwig, Jörg Peplies, and Frank Oliver O. Glöckner. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic acids research*, 35(21):7188–7196, December 2007.

[30] K.D. Pruitt, T. Tatusova, and D.R. Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 35(Database issue), January 2007.

[31] Kottmann R., Gray T., Murphy S., Kagan L., Kravitz S., Lombardot T., Field D., and Glöckner FO; Genomic Standards Consortium. A standard MIGS/MIMS compliant XML Schema: toward the development of the Genomic Contextual Data Markup Language (GCDML). *OMICS*, 12(2):115–21, 2008.

[32] J. Reeder and R. Knight. The 'rare biosphere': a reality check. *Nat Methods*, 6(9):636–7, 2009.

[33] Mina Rho, Haixu Tang, and Yuzhen Ye. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic acids research*, 38(20):e191–e191, 2010.

[34] C.S. Riesenfeld, P.D. Schloss, and J. Handelsman. Metagenomics: genomic analysis of microbial communities. *Annu Rev Genet*, 38:525–52, 2004.

[35] E.E. Snyder, N. Kampanya, J. Lu, E.K. Nordberg, H.R. Karur, M. Shukla, J. Soneja, Y. Tian, T. Xue, H. Yoo, F. Zhang, C. Dharmanolla, N.V. Dongre, J.J. Gillespie, J. Hamelius, M. Hance, K.I. Huntington, D. Jukneliene, J. Koziski, L. Mackasmiel, S.P. Mane, V. Nguyen, A. Purkayastha, J. Shallom, G. Yu, Y. Guo, J. Gabbard, D. Hix, A.F. Azad, S.C. Baker, S.M. Boyle, Y. Khudyakov, X.J. Meng, C. Rupprecht, J. Vinje, O.R. Crasta, M.J. Czar, A. Dickerman, J.D. Eckart, R. Kenyon, R. Will, J.C. Setubal, and B.W. Sobral. PATRIC: the VBI PathoSystems Resource Integration Center. *Nucleic Acids Res*, 35(Database issue), January 2007.

[36] Terry Speed. *Statistical Analysis of Gene Expression Microarray Data*. Chapman and Hall/CRC, 2003.

[37] R.L. Tatusov, N.D. Fedorova, J.D. Jackson, A.R. Jacobs, B. Kiryutin, E.V. Koonin, D.M. Krylov, R. Mazumder, S.L. Mekhedov, A.N. Nikolskaya, B.S. Rao, S. Smirnov, A.V. Sverdlov, S. Vasudevan, Y.I. Wolf, J.J. Yin, and D.A. Natale. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4:41, 2003.

[38] Torsten Thomas, Jack Gilbert, and Folker Meyer. Metagenomics - a guide from sampling to data analysis. *Microbial Informatics and Experimentation*, 2(1):3, 2012.

[39] W.L. Trimble, K.P. Keegan, M. D'Souza, A. Wilke, J. Wilkening, J. Gilbert, and F. Meyer. Short-read reading-frame predictors are not created equal: sequence error causes loss of signal. *BMC Bioinformatics*, 13(1):183, 2012.

[40] P.J. Turnbaugh, R.E. Ley, M.A. Mahowald, V. Magrini, E.R. Mardis, and J.I. Gordon. An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature*, 444(7122):1027–31, 2006.

[41] A. Wilke, T. Harrison, J. Wilkening, D. Field, E.M. Glass, N. Kyrpides, K. Mavrommatis, and F. Meyer. The M5nr: a novel non-redundant database containing protein sequences and annotations from multiple sources and associated tools. *BMC Bioinformatics*, 13:141, 2012.

[42] A. Wilke, J. Wilkening, E.M. Glass, N. Desai, and F. Meyer. An experience report: porting the MG-RAST rapid metagenomics analysis pipeline to the cloud. *Concurrency and Computation: Practice and Experience*, 23(17):2250–2257, 2011.

[43] Andreas Wilke, Wolfgang Gerlach, Travis Harrison, Tobias Paczian, Wei Tang, William L. Trimble, Jared Wilkening, Narayan Desai, and Folker Meyer. Shock: Active storage for multicloud streaming data analysis. In Ioan Raicu, Omer F. Rana, and Rajkumar Buyya, editors, *2nd IEEE/ACM International Symposium on Big Data Computing, BDC 2015, Limassol, Cyprus, December 7-10, 2015*, pages 68–72. IEEE, 2015.

[44] J. Wilkening, A. Wilke, N. Desai, and F. Meyer. Using clouds for Metagenomics: A case study. In *IEEE Cluster 2009*, 2009.

[45] Pelin Yilmaz, Renzo Kottmann, Dawn Field, Rob Knight, James Cole, Linda Amaral-Zettler, Jack Gilbert, Ilene Karsch-Mizrachi, Anjanette Johnston, Guy Cochrane, Robert Vaughan, Christopher Hunter, Joonhong Park, Norman Morrison, Phillipe Rocca-Serra, Peter Sterk, Mani Arumugam, Laura Baumgartner, Bruce Birren, Martin Blaser, Vivien Bonazzi, Peer Bork, Pier Luigi Buttigieg, Patrick Chain, Elizabeth Costello, Heather Huot-Creasy, Peter Dawyndt, Todd DeSantis, Noah Fierer, Jed Fuhrman, Rachel Gallery, Richard Gibbs, Michelle Gwinn Giglio, Inigo San Gil, Elizabeth Glass, Antonio Gonzalez, Jeffrey Gordon, Robert Guralnick, Wolfgang Hankeln, Sarah Highlander, Philip Hugenholtz, Janet Jansson, Jerry Kennedy, Dan Knights, Omry Koren, Justin Kuczynski, Nikos Kyrpides, Robert Larsen, Christian Lauber, Teresa Legg, Ruth Ley, Catherine Lozupone, Wolfgang Ludwig, Donna Lyons, Eamonn Maguire, Barbara Methé, Folker Meyer, Sara Nakielny, Karen Nelson, Diana Nemergut, Josh Neufeld, Norman Pace, Giriprakash Palanisamy, Jörg Peplies, Jane Peterson, Joseph Petrosino, Lita Proctor, Jeroen Raes, Sujeevan Ratnasingham, Jacques Ravel, David Relman, Susanna Assunta-Sansone, Lynn Schriml, Erica Sodergren, Aymé Spor, Jesse Stombaugh, James Tiedje, Doyle Ward, George Weinstock, Doug Wendel, Owen White, Andreas Wilke, Jennifer Wortmann, and Frank Oliver Glöckner. The "Minimum Information about an ENvironmental Sequence" (MIENS) specification. *Nature Biotechnology*, 2010.