MW 03 MetaBatch and Metabolomics Workbench Architecture
Tod Casasent
2021-09-14-0835

Introduction

This document is to familiarize users and developers with the technologies, techniques, and architecture behind the MetaBatch Omic Browser and the MDACC Standardized Data Metabolomics Workbench Tool.

We would like to extend special thanks to Eoin Fahy and Shankar Subramaniam for assistance with this project. In particular, after discussions and viewing an early version of our results, Eoin added the allfactors RESTful HTTP endpoint to allow us to programmatically gather all possible batch types.

Definitions

Standardized Data Format consists generally of two files -- a data matrix file and a batch information file. These are found in the Data Archive download ZIP file in the original directory and are names original_matrix_data.tsv and original_batches.tsv.

Standardized Data "Data Matrix" Format

The Standardized Data "Data Matrix" format is a tab delimited file. The first line of the file begins with a tab and contains sample identifiers. For Standardized Data from the Metabolomics Workbench, the sample identifiers are Metabolomics Workbench Sample Identifiers, whose format is generally unique to each project. Each subsequent row begins with a Feature Identifier and is followed by numeric data. Feature Identifiers are also often specific to the project, but are generally standardized to Metabolomics Workbench metabolite names.

This extract from the Data Matrix format shows four sample ids and five feature ids. Note that the first blank cell indicates the starting tab for the sample identifiers line.

	S00008880	S00008886	S00008938	S0000
11BETA,21-DIHYDROXY-5BETA-PREGNANE-3,20-DIONE	126491.00	67897.00	0	0
11-BETA-HYDROXYANDROST-4-ENE-3,17-DIONE	0	246810.00	126695.00	40840
13(S)-HPODE	0	814496.00	914760.00	75951
15(S)-HETE	0	0	674112.00	33984
17BETA-HYDROXYANDROSTAN-3-ONE	0	0	0	0

The file ngchm_link_map.tsv provides a mapping between the feature name

from the matrix file and other metabolite identifiers. The rol_cols_types.tsv file gives NGCHM-compatible names for the different id origins. These identifiers are pulled from Metabolomics Workbench data and described later. Not all features have valid mappings. Current id origins are: RefMet, PubChem, Metabolomics Workbench regno, HMDB, KEGG, CheBi, and MetaCyc. Sampes are:

Standardized Data Batch File Format

The Standardized Data Batch File format is also a tab delimited file. The first line of the file contains the sample id column id and batch type identifiers, none of which should contain spaces. The first entry should be the "Sample" column, which contains sample ids. For TCGA data (from the DCC and the GDC), the other batch type identifiers are Type, BatchId, PlateId, ShipDate, and TSS.

Sample	AFTER_MEAL_TIME	POST_SURGERY_TIME	SURGERY	gender
S00008865	NO_WAIT	PRE_SURGERY	ROUX_EN_Y	female
S00008866	30_MIN	PRE_SURGERY	ROUX_EN_Y	female
S00008867	NO_WAIT	12MO	ROUX_EN_Y	female
S00008868	30_MIN	12MO	ROUX_EN_Y	female

Batch files have been filtered using rules-of-thumb described later to limit batch types (called factors in Metabolomics Workbench) to ones whose data forms useful batches. That is, factors with all or mostly unique or identical values per sample have been dropped.

Components

Public components are available on GitHub and Docker Hub. The MD Anderson Department of Bioinformatics and Computational Biology on GitHub https://github.com/MD-Anderson-Bioinformatics/ has Docker-Compose files for using the Docker images.

The MDACC Standardized Data Metabolomics Workbench Tool is a Java library and JavaScript/Java web user interface for retrieving datasets from the Metabolomics Workbench (https://metabolomicsworkbench.org) and converting them to Standardized Data Format for processing by the MetaBatch Pipeline.

The MetaBatch Pipeline is a Java pipeline for processing Standardized Data using the MBatch R package via the MetaBatch Analyzer.

The MBatch R package and MBatchUtils R package contain the base batch effects analysis and correction options (MBatch) and extended options such as running from reproducible configuration files and generating NGCHM output (MBatchUtils).

The MetaBatch Analyzer is a collection of Docker images running R, Java, and JavaScript that includes a web GUI for non-developers to use the MBatch packages and to view the results with an included MetaBatch Omic Browser.

The MetaBatch Omic Browser is a Java and JavaScript web application that reads the results archive ZIP files from MBatch and displays interactive results for the user. Interactive results support generating SVG PDFs for resizable print-quality graphics.

Overall Architecture

In general, data is retrieved by the pipeline from the Metabolomics Workbench using their HTTP API. Data is converted to Standardized Data Format. The Standardized Data is processed by MetaBatch Analyzer using MBatch. Then the MBatch Results are displayed by the MBatch Omic Browser.

Metabolomics Workbench

We would like to extend special thanks to Eoin Fahy and Shankar Subramaniam for assistance with this project. In particular, after discussions and viewing an early version of our results, Eoin added the allfactors RESTful HTTP endpoint to allow us to programmatically gather all possible batch types.

Generating the Cache

Our first step is to generate the Cache of known datasets.

Collecting Cache Study / Summary

First, we collect from the Study Summary endpoint, which lists all Studies and some information. Not all Studies and Analysis have data available via the API.

https://www.metabolomicsworkbench.org/rest/study/study_id/ST/summary

Collecting Cache Analysis

The next step for each Study is to fetch any Analysis associated with each Study. (As a reminder, Analysis is the level at which data exists in Metabolomics Workbench and each Study can have multiple Analysis.

https://www.metabolomicsworkbench.org/rest/study/study_id/<StudyId>/analysis

Collecting Cache Analysis Metabolites

We also collect all Metabolites associated with each Analysis, necessary since each Analysis uses its own version of metabolite names.

https://www.metabolomicsworkbench.org/rest/study/analysis_id/<AnalysisId>/metabolites

Read RefMet File

Stepping up the hierarchy, outside of Studies or Analysis, we read in the RefMet file from Metabolomics Workbench which has a list of known metabolites for mapping.

Collect Other Ids for Metabolites

Using PubChem Ids from RefMet and the Analysis Metabolite lists, we collect lists of other ids for metabolites, which we use to generate link outs in NGCHMs so researchers can look up metabolite information on the fly.

https://www.metabolomicsworkbench.org/rest/compound/pubchem_cid/<pubchemId>/all

Converting the Data

Using the cache and an index that tracks processes datasets (Analysis), for each new dataset in the cache, we download and convert the data.

Download the Matrix Data

Data is available in three different formats: Raw, Drop Class, and Merged Sample-Class. Data all comes from the datatable/txt endpoint, which is Raw Data with samples and features. But some datasets include an additional Class row, which is sometimes not needed, and can be dropped for Drop Class. And finally, some datasets need the Sample and Class to be merged to generate unique identifiers for the samples, Merged Sample-Class. We generate all three and test to see which generates a unique list of samples.

https://www.metabolomicsworkbench.org/rest/study/analysis_id/<AnalysisId>/datatable/txt

Download the Batch Data

As mentioned, what the more general Batch Effects community calls "batch types" is referred to in the Metabolomics community as Factors. Here we use the allfactors endpoint Eoin created for us to get all factors associated with a Study. (Note that factors are associated with the Study, not the Analysis.)

https://www.metabolomicsworkbench.org/rest/study/study_id/<theStudyId>/allfactors

We also filter the factors to generate useful for Batch Effects Analysis Batch Types. In general this means:

- Dropping batches that are "-" or "", indicating NA or no information.
- If a Batch Type has only one batch, we drop it as not useful for Batch Effects Analysis.
- If the number of batches is equal to or greater than 90% of samples count, then do not use the factor.

- If less than 60% of samples have batch values, then do not use that factor.

This does not mean the resulting Batch Types are actually useful or meaningful for Batch Effects Analysis, but it removes the greatest majority of elements like alternative ids or file names that are not useful.

Write the Analysis Metabolites

We also generate a list of metabolites and Other Ids for the Analysis.

Generate Final Files

From this data, we generate a matrix data file, batches file, and a file that maps the feature names to link out ids for metabolites.

Analysis

Data when processed is log transformed. The data matrix is converted to a vector and all zeros and NAs are removed. We then calculate the .1 quantile, add that to each value in the matrix and compute the base 2 log value for each element.

MBatch supports two types of analysis. One analysis called "continuous" uses parametric tests particularly useful for non-sparse matrices -- although some will work with sparse matrices. This includes PCA+, DSC (Dispersion Separability Criterion https://bioinformatics.mdanderson.org/public-software/tcga-batch-effects/#the-dsc-metric), Boxplots, UMAP, NGCHM, Supervised Clustering, and Hierarchical Clustering.

The second type of analysis is called "discrete" and uses non-parametric tests particularly useful for sparse matrices (such as mutation data and some metabolomics data) and is also useful for very large datasets. This analysis performs the Kruskal-Wallis Test by Rank, a non-parametric test ideal for the sparse nature of certain data. The Kruskal-Wallis test by rank gives p-values for significance, with significant deviations having a p-value less than a specified value. We use .00001 as a default. For datasets where Kruskal-Wallis indicates a significant deviation (that is, probably batch effects), we perform a Dunn's Test to determine which batches or elements are the likely source of the batch effect.