GDC Converters
Tod Casasent
2023-10-03-1600

Introduction

This document purports to cover in somewhat technical terms the way converters take GDC Harmonized data and convert them to Standardized Data. By "technical", I generally mean specifying columns being converted and the like, rather than code descriptions. The target audience for this document is consumers of Standardized Data who have an interest in how we standardized the data, in order to confirm the data contains what they think it contains.

This document will cover individual converters. For a more general overview, the code for downloading and converting is available online here: https://github.com/MD-Anderson-Bioinformatics/BatchEffectsPackage/blob/master/apps/PyMBatch/scripts/convert_gdc.py

Data Groups

Different kinds of data are acquired from the GDC:

- Biospecimen data used to produce batch information
- (Public) Clinical data used to produce clinical information
- Workflow data data from a GDC workflow, used to produce matrix and mutation data as appropriate.

Output Files

The conversion process produces four kinds of files:

- matrix data data in a samples by features matrix format
- batch data data in a dataframe giving batch variables and values associated with samples
- clinical data data in a dataframe giving public clinical variables and values associated with patients (and samples)
- mutation data data in a sparse matrix-like TSV file describing the different mutation calls--basically a cleaned up version of the basic MAF.

Datasets

A dataset is defined by a unique set of categories specifying the platform, project, program, and other information that specifies a unique set of samples and data.

Overall Flow

The overall flow of the conversion process is:

1. From the GDC API, collect manifest information (files, samples, and patients) associated with a dataset. (Manifests exist for Biospecimen, Clinical, and Workflow data.)
2. Download the files associated with a manifest.
3. Convert the Workflow data to a matrix/mutation files, and use related Biospecimen data to generate batch files and Clinical data to general clinical files.

File Downloads

Files are downloaded by using the file UUID from the manifest and the /data endpoint. Files are named for the MD5SUM and the file name and stored in dataset directories.

Batch and Clinical Conversion

Biospecimen XML

Biospecimen files are XML files. Each file contains information for the same patient and share the same BCR, batch, project, disease, tissue source site (TSS), and sex. Data is used to create a batch file. "Unknown" is used when values are not provided.

Samples are related to either a "portion" or an "aliquot".

For Portion Samples, the following XML conversion is used:

Column Id	Parent Element	Tag
Project	admin:admin	admin:pro
Disease	admin:admin	admin:dis
Batch	admin:admin	admin:bat
Bcr	admin:admin	admin:bcr
Tss	document	shared:tis
Sex	document	shared:ge
For each sample element in document with tag bio:sample		
SampleTypeId	sample	bio:sample
SampleTypeName	sample	bio:sample
For each portion element in sample element with tag bio:shipment_portion		
Barcode	portion	bio:shipm
Uuid	portion	bio:bcr_s
PlateId	portion	bio:plate_

Column Id	Parent Element	Tag
ShipDate	portion	bio:day_o
		bio:month
		bio:year_
SourceCenter	portion	bio:center
IsFfpe	sample	bio:is_ffp
	portion	bio:is_ffp

For Aliquot Samples, the following XML conversion is used:

Column Id	Parent Element	Tag
Project	admin:admin	admin:project_code
Disease	admin:admin	admin:disease_code
Batch	admin:admin	admin:batch_numbe
Bcr	admin:admin	admin:bcr
Tss	document	shared:tissue_source
Sex	document	shared:gender
For each sample element in document with tag bio:sample		
SampleTypeId	sample	bio:sample_type_id
SampleTypeName	sample	bio:sample_type
For each analyte element in sample element with tag bio:analyte		
For each aliquot element in analyte element with tag bio:aliquot		
Barcode	aliquot	bio:bcr_aliquot_bar
Uuid	aliquot	bio:bcr_aliquot_uui
PlateId	aliquot	bio:plate_id
ShipDate	aliquot	bio:day_of_shipmen
		bio:month_of_shipm
		bio:year_of_shipmer
SourceCenter	aliquot	bio:center_id
AliquotCenter	aliquot	bio:source_center
AliquotConcentration	aliquot	bio:concentration
AliquotQuantity	aliquot	bio:quantity
AliquotVolume	aliquot	bio:volume
PlateRow	aliquot	bio:plate_row
PlateColumn	aliquot	bio:plate_column
IsFfpe	sample	bio:is_ffpe
	aliquot	bio:is_derived_from_

Clinical XML

Column Id	Parent Element	Tag
bcr_patient_barcode	document	shared:bcr_patient_b
bcr_patient_uuid	document	shared:bcr_patient_u
days_to_birth	document	clin_shared:days_to_
height	document	clin_shared:height
weight	document	clin_shared:weight
race	document	clin_shared:race
		concatenated pipe deli
ethnicity	document	clin_shared:ethnicity
vital_status	document	clin_shared:vital_stat
days_to_last_followup	document	clin_shared:days_to_
days_to_last_known_alive	document	clin_shared:days_to_
days_to_death	document	clin_shared:days_to_
relative_family_cancer_history	document	clin_shared:relative_f
cancer_first_degree_relative	document	clin_shared:cancer_fi
clinical_stage	shared_stage:stage_event	shared_stage:clinical_
pathologic_stage	shared_stage:stage_event	shared_stage:patholog
age_at_initial_pathologic_diagnosis	document	clin_shared:age_at_i
follow_up_vital_status	document	clin_shared:vital_stat
follow_up_days_to_last_followup	document	clin_shared:days_to_
follow_up_days_to_death	document	clin_shared:days_to_
follow_up_new_tumor_event_after_initial_treatment	document	nte:new_tumor_event

SNP6-Type Conversion

This conversion type is based on the historical handling of SNP6 data. The data types using this conversion are:

Genotyping Array -> Copy Number Segment -> DNAcopy -> Copy-Number-with-CNV

Genotyping Array -> Masked Copy Number Segment -> DNAcopy -> Copy-Number-no-CNV

This type of processing uses the "Segment_Mean" column as the value and the information from the "Chromosome", "Start", and "End" columns for information about the gene. This then uses the algorithm described in the StdData_03_Docs_GeneScoreAlgorithm document for calculating a generic "gene score value".

StarCount-Type Conversion

This conversion type is based on the handling of STAR count data. The data types using this conversion are:

RNA-Seq -> Gene Expression Quantification -> STAR - Counts -> RNASeq-TPM

RNA-Seq -> Gene Expression Quantification -> STAR - Counts -> RNASeq-FPKM

RNA-Seq -> Gene Expression Quantification -> STAR - Counts -> RNASeq-FPKM-UQ

This type of processing uses the "gene_name" column and the "gene_id" column values, separated by a pipe "|", as the feature id. The matrix value used depends on the dataset being processed and is one of the following: "tpm_unstranded", "fpkm_unstranded", "fpkm_uq_unstranded".

Sesame Methylation-Type Conversion

This conversion type is based on the handling of the SeSAMe Methylation data. (GDC recalculated Methylation 27 and 450 data.) The data types using this conversion are:

Methylation Array -> Methylation Beta Value -> SeSAMe Methylation Beta Estimation -> Methylation-With-Sex-Chromosomes

Methylation Array -> Methylation Beta Value -> SeSAMe Methylation Beta Estimation -> Methylation-No-Sex-Chromosomes

This type of processing uses the first column as the probe and the second column as a value. (Usually, these files do not have a header row.) For some datasets, we exclude x and y chromosome data.

RPPA-Type Conversion

This conversion type is based on the handling of RPPA (Reverse Phase Protein Array) data. The data types using this conversion are:

Reverse Phase Protein Array -> Protein Expression Quantification -> Protein Analysis -> RPPA

This type of processing builds the pipe ("|") delimited feature name using the "peptide_target" and "AGID" column values. The data comes from the "protein_expression" column.

miRNA-Type Conversion

This conversion type is based on miRNA-Seq data handling. The data types using this conversion are:

miRNA-Seq -> Isoform Expression Quantification -> BCGSC miRNA Profiling -> miRNA-Isoform

miRNA-Seq -> miRNA Expression Quantification -> BCGSC miRNA Profiling -> miRNA-Genes

This type of processing uses the "miRNA_ID" and "miRNA_region" columns for the feature id. If the region is empty, only the miRNA_ID is used. If miRNA_region starts with "mature", anything after the comma in the value is added to the miRNA_ID with a pipe "|" separating them. Otherwise, if region is not empty, the region is added to the miRNA_ID with a pipe "|" separating them. If the feature name ends in "_calculated", that substring is removed.

The value of the column "reads_per_million_miRNA_mapped" is used as the value. If there are multiple entries for the same feature, values are added.

Copy Number-Type Conversion

This conversion type is based on the handling of Copy Number data. The data types using this conversion are:

Genotyping Array -> Gene Level Copy Number -> ASCAT2 -> Copy-Number-With-Sex-Chromosomes

Genotyping Array -> Gene Level Copy Number -> ASCAT2 -> Copy-Number-No-Sex-Chromosomes

WGS -> Gene Level Copy Number -> AscatNGS -> Copy-Number-With-Sex-Chromosomes

WGS -> Gene Level Copy Number -> AscatNGS -> Copy-Number-No-Sex-Chromosomes

This type of processing uses the columns "gene_name" and "gene_id" to build the feature id. If the gene_id column value ends in _PAR_Y, that row is ignored. Otherwise, gene_name and gene_id are concatenated with a pipe "|" in between.

The column "chromosome" is used to sort our chrx and chry when requested.

The value in the column "copy_number" is used as the data value.

Mutation Calling-Type Conversion

This conversion type is for all "Mutation-Calling" data at the GDC.

This type of processing generates two different files. One is a matrix containing non-silent mutation counts. The other file uses a MAF-like format with columns standardized. See converter_mutationmaf.py for details.

scRNA-Type Conversion

This conversion type handles single-cell RNA Differential data. The data types using this conversion are:

scRNA-Seq -> Differential Gene Expression -> Seurat - 10x Chromium -> scRNA-Differential

This type of processing uses the "gene" column for the feature and the "avg_log2FC" as the value.