Department of Computer Science, UTSA
Technical Report: CS-TR-2008-008

Mapping microarray chip feature IDs to Gene IDs for microarray platforms in NCBI GEO

Cory Burkhardt and Kay A. Robbins
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX 78249
{cburk, krobbins}@cs.utsa.edu

Abstract

The Gene Expression Omnibus or NCBI GEO (http://www.ncbi.nlm.nih.gov/projects/geo/) is one of the most comprehensive public repositories of microarray data. The data in NCBI GEO is organized into samples, series, datasets and platforms. Platform files specify the characteristics of a particular type of microarray chip, including a mapping of chip features to genetic features. These mappings are not standardized. This technical note describes a program GPLGeneMap that reads a platform specification in XML format and produces a mapping of chip features to NCBI Gene IDs. The mapping strategy is based on a user specification given in an XML file. The program produces a mapping file and a list of chip features that it was unable to map.

Introduction

The Gene Expression Omnibus or NCBI GEO (http://www.ncbi.nlm.nih.gov/projects/geo/) is a comprehensive public repository of microarray data. The data in NCBI GEO is organized into samples, series, datasets, and platforms. The data is linked to other NCBI resources.

A *sample* represents the expression measurements from one microarray chip. Samples are identified by accession IDs that begin with the GSM prefix. Sample files contain expression measurements presented as a table of ID_REF and VALUE pairs. The value in the ID_REF column identifies the chip feature associated with the corresponding expression value given in the VALUE column.

A *series* represents a collection of samples from the same experiment. Series are identified by accession IDs that begin with the GSE prefix. A *dataset* represents a collection of samples (usually a subset from the same series) that have been curated and annotated with common control variables. Datasets are identified by accession IDs that begin with the GDS prefix. A sample may appear in multiple series and datasets. Series may have samples from different platforms.

A *platform* represents a specific type of microarray chip. Platforms are identified by accession IDs that begin with the GPL prefix. The platform file, which defines how the IDs for the features on the chip map to genetic features, may contain multiple ways of associating chip feature IDs with these genetic features. Many downstream analyses require expression values to be associated with Gene IDs, so that measurements of expression can be identified with genes.

Unfortunately the association of chip features with genetic features varies considerably across platform specifications. For Affymetrix in situ oligonucleotide microarrays, the chip features are probesets identified by probeset IDs. Some platforms have multiple chip features to interrogate the same gene, and most platforms dedicate some features for housekeeping and scaling purposes. Furthermore, each platform can specify alternative ways of identifying genetic features. For example platform GPL96 (Affymetrix GeneChip Human Genome U133 Array Set HG-U133A) provides the Entrez gene ID, the GenBank accession number, the gene symbol and reference transcript ID to identify a genetic feature. GPL4557 (Affymetrix GeneChip Human Genome U133 Array Set HG-U133A), on the other hand, only provides the GenBack accession number.

Configuration

GPLGeneMap uses a tiered approach for translating chip features to Gene IDs. The user specifies the order in which the identifier lookups are performed in an XML configuration file. The types of identifiers accepted by GPLGeneMap are: the Gene ID, the Nucleotide

Accession, the UniGene Cluster ID, and the Gene Name/Symbol and its aliases. Table 1 summarizes the different types of identifying columns that can be parsed.

Table 1: ID type specifiers indicating type of IDs that can be parsed.

Genetic Feature	XML type attribute
Gene ID	GENE_ID
Nucleotide	NUCLEOTIDE_ACCESSION
Unigene Cluster ID	UNIGENE_CLUSTER_ID
Gene Name/Symbol	GENE_ALIAS

The following is an example of the configuration used for the GPL96 platform:

```
<platform name="GPL96">
        <geneColumn name="ENTREZ_GENE_ID" type="GENE_ID">
                <split>///</split>
                <group name="ID" default="true" location="1" />
                <group name="All" location="*" />
        </geneColumn>
        <geneColumn name="Representative Public ID" type="NUCLEOTIDE_ACCESSION">
                <group name="ID" default="true" />
        </geneColumn>
</platform>
```

Using this configuration, GPLGeneMap first attempts to use the Gene ID listed directly in the platform table under the column named "ENTREZ_GENE_ID". If this field is missing or is invalid, then the Nucleotide Accession identifier listed under the column named "Representative Public ID" is used as a secondary lookup.

Sometimes multiple values of an identifier are listed for a chip feature. In GPL96, the Gene ID field may list multiple values separated by sequences of "///" characters. The directives within the <geneColumn> element inform GPLGeneMap how to parse these values out of the platform specification file. The <split> element tells the parser to split the field using "///" as the delimiter. The <group> element with the name attribute ID indicates which value within the group should be used as the primary identifier. In the example, the first value in the field (location="1") will be used. The <group> element with the "All" name attribute indicates which values in the field represent valid IDs. In the example, all values

(location="*") specify valid Gene IDs. The attribute location="1" specifies that only the first field represents a valid ID. If a spot has more than one Gene ID listed, the primary Gene ID is written to the output map file, and the complete list of Gene IDs is written to a supplementary file. In the example, the secondary lookup using Nucleotide Accession identifiers does not perform any special parsing because it contains no <split> directives.

Identifier Resolution

GPLGeneMap uses a different algorithm to resolve each of the identifier types. The Gene IDs listed directly in the platform table are used verbatim, but the resolution process for the other identifiers is more involved. GPLGeneMap makes use of the NCBI E-Utilities (http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html) to search for these identifiers in the appropriate NCBI databases and follows links from these entries to their associated entries in the Entrez Gene database. The E-Utilities are a collection of XML Web services hosted by NCBI's web servers. GPLGeneMap uses the ESearch, ELink, and ESummary utilities.

The detailed resolution algorithms for each of the identifiers are described below. It is possible for any of the following algorithms to find multiple gene results for a single identifier. The algorithm for choosing a best candidate among these results is described later.

Gene Name/Symbol and Gene Aliases

Platforms such as GPL198 and GPL513 contain identifiers that are resolved as Gene Aliases in the NCBI Gene database. Resolving Gene Alias identifiers involves directly searching the NCBI Gene database. Each record in the Gene database has a name and a list of aliases. For instance, the record with Gene ID 242109 is named Zfp697 and is also known by the aliases AI467503 and 9430053K19. GPLGeneMap matches Gene Alias identifiers against these values when resolving them to Gene IDs.

First, GPLGeneMap formulates a search query. The search query contains two clauses. The first clause in the query consists of the taxonomy name. This prevents genes from other organisms from appearing in the results. The second clause consists of all the identifiers. An example search query for the organism *Mus musculus* and the two identifiers Zfp697 and Scn5a would be: "Mus musculus[ORGN] AND (Zfp697[GENE] OR Scn5a[GENE])". GPLGeneMap performs this search using ESearch and inspects the results using ESummary, parsing out each record's Gene ID. GPLGeneMap performs the searches in batches with up to 100 identifiers in each.

It is possible for a single identifier in a search to return multiple results. Some of these results may be current and some may be discontinued. For instance, the search "Arabidopsis

thaliana[ORGN] AND (at2g14230[GENE])" yields two results for the identifier At2g14230. However, one of the two records has been discontinued. If any of the found gene records for an identifier are current, only those records are used, and the discontinued records are discarded. If the only record(s) found are discontinued, then those records are used. If at the end of the process, an identifier has mapped to multiple gene records, the best candidate is determined using the algorithm described later. However, if more than twenty candidate gene records are found, an error is generated to prevent GPLGeneMap from overloading the NCBI servers with requests for large amounts of data. It then continues with the next identifier. In the search for At2g14230, the record with Gene ID 3767832 is the only current record, so this is the record that would be chosen. GPLGeneMap uses the Gene IDs of the chosen gene records when outputting the chip feature to the Gene ID mapping file.

Nucleotide Accession

Many platforms list Nucleotide Accession identifiers somewhere in the platform table. Some platforms, such as GPL4557 and GPL1820, use these as the primary identifier. GPLGeneMap resolves Nucleotide Accession identifiers by following a trail of links from the Nucleotide databases to the Gene database. GPLGeneMap first generates a query to search the Nucleotide database (which is actually composed of the CoreNucleotide, EST, and GSS databases). The query consists of each identifier, qualified with the field constraint "[ACCN]". An example query for the accession values AI849067, AE000665, and NM_025313 is: "AI849067[ACCN] OR AE000665[ACCN] OR NM_025313[ACCN]". GPLGeneMap searches up to 100 identifiers per query. After generating the query, GPLGeneMap then executes the search on the Nucleotide database using ESearch. It then retrieves the results of the search using ESummary and parses out the integer Nucleotide unique identifiers (UIDs). The UIDs retrieved for the search above are 5492973, 2358100, and 166851827.

In the next step, GPLGeneMap finds links from the records that were found in the Nucleotide databases to records in the UniGene database. However, the Nucleotide records that were returned can exist in either the CoreNucleotide or the EST databases. Links from these records can only be queried from one of these databases at a time. First, GPLGeneMap requests links from the EST database to UniGene using ELink, and then it requests the links from the CoreNucleotide records. Using the results from the query in the previous paragraph, a link is retrieved from the AI849067 record in EST to the record titled Mm.270382 in UniGene. A link from the NM_025313 record in CoreNucleotide to the record titled Mm.278560 in UniGene is also retrieved. No links are found to UniGene for the AE000665 record. Once GPLGeneMap has retrieved the links to the UniGene records, it then requests links from these UniGene records to records in the Gene database. The Gene IDs of these records are then used for the Gene ID mapping. If an identifier has more than twenty links to UniGene or the found UniGene identifiers have more than twenty total links

to Gene, GPLGeneMap generates an error and does not follow the links. It then continues with the next identifier. The Mm.270382 and Mm.278560 records found previously link to the gene records titled Yipf5 and Atp5d with Gene IDs 67180 and 66043, respectively.

If any records did not resolve to any gene records using the previous steps, GPLGeneMap requests links directly from these records in CoreNucleotide and Nucleotide EST to the Gene database. It then combines these results with the previous results. In the previous example, record AE000665 of CoreNucleotide did not have any links to UniGene. Therefore, GPLGeneMap would then search for links directly from CoreNucleotide to Gene. This search yields 6 results in Gene titled EG436523, EG386551, Tcrb-V13, Prss1, Prss2, and Prss3 with Gene IDs 436523, 386551, 269846, 114228, 22072, and 22073 respectively. Because this search yielded multiple Gene IDs for this identifier, the best of these candidate Gene IDs must be resolved. The algorithm for handling this situation is described below. If an identifier has more than twenty links to Gene, GPLGeneMap generates an error and continues with the next identifier.

UniGene Cluster ID

GPLGeneMap resolves UniGene Cluster identifiers by following links from the UniGene database to the gene database. First, it generates a query to search UniGene for records matching the identifiers in the platform table. This query consists of each identifier qualified with the field constraint "[CID]". An example query for the cluster IDs Hs.112873 and Hs.494173 is: "Hs.112873[CID] OR Hs.494173[CID]". GPLGeneMap searches up to 100 identifiers per query. After generating the query, GPLGeneMap executes the search on the UniGene database using ESearch and retrieves and parses the results using ESummary. Finally, GPLGeneMap passes these results to ELink to find links from the UniGene records to records in the Gene database and uses these records' Gene IDs when outputting the spot to Gene ID mapping file. If a UniGene record has more than twenty links to Gene, GPLGeneMap generates an error and continues with the next identifier.

Handling Multiple Gene Candidates

Any of the three previous identifier categories can yield results that contain multiple gene records for a single identifier. When this occurs, GPLGeneMap attempts to use the information available about these gene records to choose the best available record to associate with the spot in the mapping table. It uses EFetch to retrieve each gene's available XML data and looks at the RefSeq Status field to determine each gene's currently considered significance. The status values accepted by GPLGeneMap in order of preference are: reviewed, validated, provisional, predicted, and model. If a single candidate gene has a status that is preferred more than all the others, then that gene is used when outputting the chip feature to Gene ID map file. However, if more than one gene shares the same preferred

status, then an error is generated, and processing of this spot continues with the next available identifier, if one is available.

Using the example from the section describing Nucleotide Accession resolution, one of the identifiers, AE000665, resolved to multiple Gene IDs: 436523, 386551, 269846, 114228, 22072, and 22073. GPLGeneMap retrieves the XML for each of these gene records and examines their RefSeq Status fields. The value of this field is validated for all records except for the record titled Tcrb-V13 with Gene ID 269846. This record has a status of reviewed. Because the reviewed status is preferred over the validated status, the Tcrb-V13 record is chosen as the best candidate, and the Gene ID 269846 is output to the map file.

Downloading Extra Gene Information

GPLGeneMap can be configured to download additional information about a gene after the output map file has been generated. This extra information includes the gene's official symbol, official name, type, and status and all of the gene's UniGene identifiers. GPLGeneMap uses the EFetch E-Utility to download the gene records in XML format and then parses the document to retrieve this information. The gene type is read from the "Entrezgene_type" element located near the beginning of the XML record. The rest of the properties are read from the "Gene-commentary" elements.

Manual ID Resolution

In some cases, you may want to resolve an ID manually using the same process as GPLGeneMap. Visit http://www.ncbi.nlm.nih.gov/sites/entrez. Before you search, you need to select the appropriate database from the drop-down. For NUCLEOTIDE_ACCESSION identifiers, you would select Nucleotide. Enter the search query (e.g. "AI849067[ACCN] OR AE000665[ACCN] OR NM_025313[ACCN]") into the text box and click search. After searching Nucleotide, there will be links at the top of the page to "Nucleotide" (which means Core Nucleotide) and/or EST records. Clicking these brings up the records in each of these databases. There is no way to get both of the lists at the same time. With either the Core Nucleotide or EST records displayed, you must follow links from these records to UniGene. The "Display" drop-down should have "Summary" selected. Change this selection to "UniGene Links". This will bring up matching UniGene records. These records must then be followed to the Gene database. Change the "Display" drop-down to "Gene Links". Any matching gene records will be displayed with their Gene IDs. You can go back in your web browser to resolve the records in the other Nucleotide database.

Caching Downloaded NCBI Data

Searching for records and fetching information from NCBI through the E-Utilities is a time-consuming activity. NCBI has set strict guidelines on when and how often programs can make requests through the E-Utilities. Once GPLGeneMap begins making E-Utilities

requests, it can take hours to complete a single platform. Therefore, GPLGeneMap attempts to reduce the number of requests it makes where it is possible.

There is considerable overlap of the gene identifiers in the tables of platforms of the same organism. Thus, it makes sense to store the results of the E-Utilities requests to resolve the identifiers so that the information is immediately available when it is encountered in other platforms. When GPLGeneMap resolves an identifier to a Gene ID or finds that it cannot resolve the gene for a particular reason, it caches this information to disk so that it is available if the identifier is encountered in the future. In addition, when GPLGeneMap downloads the XML records of genes in the NCBI Gene database, it caches the records so that it does not need to download any record more than once.

By default, GPLGeneMap caches the data to a temporary directory. Before exiting, GPLGeneMap empties the directory and removes it from the file system. However, GPLGeneMap can be configured through command line parameters to use a persistent directory for its cache. A persistent cache directory keeps the cached data available if GPLGeneMap is run again in the future.

Generating Geo Series Data Files

NCBI maintains an archive for each Geo platform that may be split into multiple files due to its large size. The archives include an XML platform file, a tab-delimited platform file mapping, and a data file for every sample submitted to NCBI GEO that is associated with the platform. The XML file defines properties of the platform and describes every series and sample associated with the platform. The XML file defines the samples that belong to each series and provides the file name of each sample's data file. The tab-delimited platform file maps microarray chip feature IDs to genetic features.

All of the experimental data for the platform is kept in individual sample data files within the archive. This arrangement can make it difficult to perform series-wide data analysis because each series' data is spread out among many data files. Once the platform map file has been generated, GPLGeneMap has the capability to reorganize this sample data into individual series data files so that each series' data is stored within a single file. The rows in these data files are mapped to the platforms features (and associated genes), and the columns are mapped to the individual samples in the series.

Because each row in the series data files defines a single feature, the data in the files must be written one feature at a time. A single row contains a data value from each sample in the series. This poses a challenge in generating these files because each value in the row is recorded in a separate sample data file. Therefore, GPLGeneMap cannot process each sample file independently when generating a series file. Moreover, it is impractical for GPLGeneMap to open every sample file at once and read a single value from each file for

every row in the series file because a single series may be composed of hundreds or thousands of samples.

To address this problem, GPLGeneMap employs a divide-and-conquer strategy to interlace the sample data into the series data file. It systematically processes a small number of sample files at a time, interweaving the sample data into a smaller number of data files. When all of the samples have been processed, this smaller set of files are then merged together in the same fashion. This process continues until a single data file has been created.

Running GPLGeneMap

GPLGeneMap is available as a Java .jar file and can be run from the command line on any system running Java. The following is the usage statement:

```
Usage: java -jar GPLGeneMap.jar [-aAcCdeFGhiIopPS]
```

The command line options are given the following table.

Table 2: Command line options for GPLGeneMap.

Option	Description
[-a <directory>]	Directory containing manually downloaded archive for each platform.
[-A <directory>]	Directory containing the extracted files from the platform archives, in subdirectories by platform name.
[-c <config file>]	Configuration file specifying the strategies to use to resolve genes in each platform. Required parameter.
[-C <cache directory>]	Persistent cache directory (not deleted when the program exits). The data cached in this directory may be reused when GPLGeneMap is run again, eliminating the need to redownload the data from NCBI. If this directory does not exist it is created.
[-d <directory>]	Directory where the platform family archive files will be downloaded and saved.
-e	Download extra gene information from NCBI and write it to the ID map file.
-F	Flush the cache directory and start with an empty cache.
-G	Do NOT resolve genes and generate ID map, only generate series data files. ID map(s) must have been generated previously (and stored within the directory specified by -o).

-h	Perform NCBI lookups during peak hours (weekdays, 5am-9pm). If not specified, GPLGeneMap will sleep until nonpeak hours before invoking the E-Utilities.
-i	Input platform name through standard input. Useful when you want to process platforms individually, but do not want to modify the command line everytime GPLGeneMap is invoked.
-I (uppercase i)	Do NOT include unresolved features in the ID map. Unless specified, GPLGeneMap will include unresolved features in the ID map, leaving the gene ID and other fields empty.
[-o <output dir>]	Output directory. The ID map and its associated files and the series data files will be output to platform subdirectories. If this directory does not exist, it is created. Required parameter.
[-p platform name]	Names a platform to be processed. GPLGeneMap will generate the ID map files and/or series data files for this platform. -p may be used multiple times, once for each platform to process.
[-P <platform list file>]	File containing a list of platforms to be processed, one per line. This parameter may be specified as an alternative to naming each platform on the command line using -p.
-S	Do NOT generate series data, only generate the ID map and associated files.

The only required parameters are -o to specify an output directory and -c to specify the XML configuration file. If none of -d, -a, or -A are used on the command line, the platform archive files are downloaded from NCBI's FTP server and extracted to a temporary directory. All of these files are deleted before the GPLGeneMap exits. If -A is used, the given directory must contain a subdirectory for every platform, using the platform names as the directory names, and each subdirectory must contain the manually downloaded and extracted contents of the platform's archive.

By default, GPLGeneMap generates the ID map and its associated files and generates the series data files for each platform. Either of these operations can be suppressed with the -S and -G parameters. The -S parameter tells GPLGeneMap to skip generating the series data files and only generate the ID map files. -G tells GPLGeneMap to skip the generation of the ID map files. If -G is given, the -o parameter must point to an existing directory where the platforms' ID maps were previously generated.

-e instructs GPLGeneMap to download extra gene information from NCBI and write this information to the ID map file(s). This extra information includes each gene's official symbol, official name, type, and status and each gene's set of UniGene identifiers.

-C can be used to specify a persistent cache directory, enabling the cache to be shared between successive runs of the program. If this parameter is not specified, the cache is written to a temporary directory and is deleted before GPLGeneMap exits. If -F is specified along with -C, the cache directory is cleared before processing begins.

GPLGeneMap is provided as a Java archive (jar) executable. The jar must be executed using the java command. Along with the parameters described above, any of the Java Virtual Machine (VM) parameters accepted by the Java command may also be specified. One of the more useful VM parameters is used to increase the maximum amount of memory that can be allocated to the program. This is done using the -Xmx VM parameter. A memory value, such as 1G or 512M is specified after the Xmx to indicate the desired size. For example,

```
java -jar GPLGeneMap.jar -f GPL198_family.xml.tgz -f GPL198_family.xml.part2.tgz -c
platform_config.xml -o output/GPL198
```

reads the two files GPL198_family.xml.tgz and GPL198_family.xml.part2.tgz that describes platform GPL198. The platform_config.xml specifies how to interpret many different platforms, including GPL198. The output is generated in the directory output/GPL198. GPLGeneMap generates up to eight ID-map related files as well as series and sample data and information files. These files are described in the following table.

Table 3: Description of the output files produced by GPLGeneMap.

File	Description
GPLXX_idmap.csv	The ID map, with optional extra gene information.
GPLXX_idmap_errors.csv	List of chip features that could not be resolved and the errors that occurred
GPLXX_idmap_error_codes.csv	List of descriptions of the error codes used in the previous file
GPLXX_multi_genes.csv	List of all the Gene IDs of spots that had more than one gene field listed
GPLXX_spec.csv	List of all the information in the platform table file in a comma-delimited format
GPLXX_map_files.tar.gz	Archive containing all of the above files.
GPLXX_idmap_preview.csv	Lists a small number of the top rows in the ID map. Used by the GPLGeneMap web page to construct the preview table.
GPLXX_idmap.htm	HTML view of the ID map. Used by the GPLGeneMap web page to provide the

	visitor with a formatted view of the entire map.
GPLXX_series.txt	List of each series that was processed successfully.
GPLXX_series_info.csv	List of every processed series and its metadata from the platform XML file.
GPLXX_series_errors.csv	List of the series which could not be processed and the errors that occurred.
GPLXX_GSEXX_data.csv	The series data file.
GPLXX_GSEXX_sample_info.csv	List of the samples in a series and their metadata from the platform XML file.
GPLXX_GSEXX_missing.csv	List of samples in a series that were missing their data files in the platform archive.
GPLXX_multiref_samples.csv	List of samples that are a part of multiple series and each of these series.