**Processing of gene2pubmed**

**Resource:** Download gene2pubmed.gz file from <ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/> and unzip.

**To retrieve gene2pubmed for human (taxon is 9606)**

*$ zcat gene2pubmed.gz | grep -w ^9606 > gene2pubmed\_Human*

Format:

|  |  |  |
| --- | --- | --- |
| **Taxon** | **Gene ID** | **PubMed ID** |
| 9606 | 1 | 2591067 |
| 9606 | 1 | 3458201 |
| 9606 | 1 | 8889549 |
| … | … | … |

**To retrieve PubMed articles with abstracts**

All PubMed articles do not include abstracts. Since, our mapping approach is carried out at abstracts or sentence level for various associations, we filtered PubMed articles with abstracts.

Approach:

For every article we checked for the presence of abstract by querying the local version of PubMed database (indexed with Apache Lucene). Please refer to our project ConceptMap:LocalPubmedDatabaseCompiler for generating a local version of PubMed database.

Java program PubMedWithAbstractsExtractor.java is meant to access the local version of PubMed. The following command is used to execute the program:

*$ javac PubMedWithAbstractsExtractor.java*

*$ java PubMedWithAbstractsExtractor INPUT\_FILE OUTPUT\_FILE*

The program needs external libraries for execution. We suggest creating a .jar to wrap the program and libraries together. The following command is used to execute the jar file:

*$ java -jar PubMedWithAbstractsExtractor.jar INPUT\_FILE OUTPUT\_FILE*

The input file is the list of PubMed IDs from gene2pubmed\_Human file.

**To retrieve gene2pubmed records with abstracts (based on PubMed ID collected from the previous step)**

Java program GeneDistributionTester.java – Method *gene2PubmedHumanWithAbstracts*() is for filtering gene2pubmed records where PubMed has abstract. The following command is used to execute the program:

*$ javac GeneDistributionTester.java*

*$ java GeneDistributionTester INPUT\_FILE\_1 INPUT\_FILE\_2 OUTPUT\_FILE*

The input files are list of PubMED IDs filtered from gene2pubmed (i.e. related to human and has abstract), gene2pubmed resource related to human (i.e. gene2pubmed\_Human file retrieved from gene2pubmed).

**To sort based on PubMed ID (File from the resource is sorted based on PubMed ID)**

*$ sort -k 3,3 INPUT\_FILE > OUTPUT\_FILE*

The input file is list of records from gene2pubmed, related to human and has abstract.

**To count number of genes mapping to each PubMed article**

Java program GeneDistributionTester.java - Method: *pubmed2geneCount*() is for counting number of genes mapping to each PubMed ID. The following command is used to execute the program:

*$ javac GeneDistributionTester.java*

*$ java GeneDistributionTester INPUT\_FILE OUTPUT\_FILE*

The input file is the output from the previous step (i.e. list of records from gene2pubmed, related to human, has abstract and sorted based on PubMed ID).

**To explore on distribution of genes in PubMed**

Java program DistributionRangePrinter.java is for counting number of PubMed\_ID falling within specific range of number of genes. This provided us knowledge on distribution of genes in PubMed. The program has to be executed for each range (by changing the condition in the if loop).

*$ javac DistributionRangePrinter.java*

*$ java DistributionRangePrinter INPUT\_FILE*

The input file is the output from the previous step (i.e. PubMed ID with number of gene mapping). The output is displayed on the console.

**Distribution (May 2016)**

|  |  |
| --- | --- |
| **Number of genes** | **Number of PubMed ID** |
| **1-5** | **469,995** |
| 6-10 | 8278 |
| 11-15 | 1639 |
| 16-20 | 735 |
| 21-25 | 386 |
| 26-30 | 184 |
| 31-35 | 193 |
| 36-40 | 119 |
| 41-45 | 88 |
| 46-50 | 56 |
| 51-55 | 39 |
| 56-60 | 51 |
| 61-65 | 42 |
| 66-70 | 37 |
| 71-75 | 30 |
| 76-80 | 37 |
| 81-85 | 23 |
| 86-90 | 19 |
| 91-95 | 23 |
| 95-100 | 16 |
| >100 | 394 |

**\*\*\* 1-5 genes per PubMed accounts for 97.4% (469,995 out of 482,380) of total PMIDs available gene2pubmed file related to human and has abstract.**

**To get PubMed IDs mapping to 1 to 5 genes**

Java program PMIDWith1to5Genes.java is for filtering PubMed IDs mapping to 1 to 5 genes. The following command is used to execute the program:

*$ javac PMIDWith1to5Genes.java*

*$ java PMIDWith1to5Genes INPUT\_FILE OUTPUT\_FILE*

The input file is list of gene2pubmed related to human, has abstract and PubMed ID is mapping up to five genes.

**To get gene2pubmed\_Human records for PMID with abstracts and mapped to 1-5 genes only**

Java program Gene2PubMedHumanWith1to5Genes.java is for collecting the records.

*$ javac Gene2PubMedHumanWith1to5Genes.java*

*$ java Gene2PubMedHumanWith1to5Genes INPUT\_FILE\_1 INPUT\_FILE\_2 OUTPUT\_FILE*

The input files are: a file with list of gene2pubmed related to human, has abstract and PubMed ID is mapping up to five genes and a second file gene2pubmed resource related to human (i.e. gene2pubmed\_Human file retrieved from gene2pubmed).

**Summary: (May 2016)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Resource (gene2pubmed)** | **Filter applied** | | |
| **Human** | **Human and PubMed with abstracts** | **Human, PubMed with abstracts, and PubMed mapping to 1 to 5 genes** |
| Total records (gene to PubMed) | 8,510,685 | 1,097,310 | 1,070,836 | 707,870 |
| Gene count | 4,513,310 | 34,079 | 33,995 | 20,086 |
| PubMed count | 985,933 | 498,259 | 482,380 | 469,995 |
| Organism count (Taxon) | 10,570 | - | - | - |

**Mapping Gene ID to Gene Name, Symbol, and Aliases**

The resource gene2pubmed from NCBI consists of Gene ID. However, we need Official Gene Symbol, Synonyms/Aliases and Name for mapping genes in the biomedical literature.

The resource **Homo\_sapiens.gene\_info.gz** from Entrez Gene contains of the information we are looking for. The file can be downloaded from:

<ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/>

**To extract Gene Name, Symbol, and Aliases for Gene ID**

Java program EntrezGeneExtractor.java is for extracting GeneID, Symbol, Synonyms and description (gene aliases) from **Homo\_sapiens.gene\_info** (unzipped file).

*$ javac EntrezGeneExtractor.java*

*$ java EntrezGeneExtractor INPUT\_FILE OUTPUT\_FIL*E

The input file is the unzipped resource file, **Homo\_sapiens.gene\_info.gz.** The output file contains Gene ID, Symbol, Synonyms and Description/Name (i.e. gene name).

**Preprocessing**

Java program GeneID2SymbolSynonymNamePreprocessor.java is for changing the format of the output file from the previous step.

*$ javac GeneID2SymbolSynonymNamePreprocessor.java*

*$ java GeneID2SymbolSynonymNamePreprocessor INPUT\_FILE OUTPUT\_FILE*

The input file is the output file from the previous step. The output file contains mapping of Gene ID to its Symbol, Aliases and Description/Name (i.e. gene name).

**To sort a file based on PubMed ID**

The output file from the Java program Gene2PubMedHumanWith1to5Genes.java is sorted.

*$ sort -k 3,3 INPUT\_FILE OUTPUT\_FILE*

**To group a list of genes mapping to each PubMed ID**

Java program PubMed2GeneCounterForPubMedWithAbstracts1to5GeneMapping.java is for grouping the genes mapping to each PubMed ID. The sorting in the previous step is mandatory for the execution of this Java program.

*$ javac PubMed2GeneCounterForPubMedWithAbstracts1to5GeneMapping.java*

*$ java PubMed2GeneCounterForPubMedWithAbstracts1to5GeneMapping INPUT\_FILE OUTPUT\_FILE*

The input file is the output from the previous step.

**To replace Gene ID with Gene Symbol, Aliases, and Description/Name (i.e. gene name)**

Java program GeneID2SymbolSynonymName.java replaces Gene ID with gene symbol, aliases and gene name.

*$ javac GeneID2SymbolSynonymName.java*

*$ java GeneID2SymbolSynonymName INPUT\_FILE1 INPUT\_FILE2 OUTPUT\_FILE*

The input files are the output file from preprocessing step and the output file from the previous step.