***Winter Internship 2017***
*Spring 2017 Semester*
*Supervisor: Asst. Prof. Dr. Tunahan ÇAKIR*

INTERNSHIP REPORT #1

# *Playing with Gene Expression Data:* GEO Database and Python-based GEO Datasets Retriever

***Muhamad Haries Ramdhani***
*Department of Molecular Biology and Genetics, Gebze Technical University, Turkey*

The advancement of science in 21st century, especially data science has pushed biology to the point where we didn't imagine it would be today. At the moment there are so many biological data spread all over the internet, one big problem is the insufficiency of our knowledge to manipulate and take the meaning out of all the data at once. One practical way to organize data is to create a database out of it thus it can be easily found then used for research purpose. There are thousands of different biological databases on the internet, range from the one which stores nucleotide data to the one which stores gene ontology data. Gene Expression Omnibus is one of them, a database which stores gene expression data from different experiments. On the first week of the internship I learned about GEO database deeper, the fundamentals of it, what can we do with the data on the database and different types of gene expression analysis experiments (DNA array, RNA-seq etc).

## INTRODUCTION

Molecular biological experiments to study gene expression utilizing high-throughput hybridization array- and sequencing-based techniques have become extremely popular in recent years [1]. Gene Expression Omnibus was created because of the high demand of gene expression datasets to be made public and its main goal was to attempt to cover the broadest spectrum of high-throughput experimental methods possible and remain flexible and responsive to future trends, rather than setting rigid requirements and standards for entry [2]. On GEO Database data sets are distinguished into four big categories, Sample, Platforms, Series and Dataset. Sample contains each sample data, Platforms are the data of the platform that were used to perform gene expression analysis, Series are built of uncurated data while Dataset is composed of curated data sets done by GEO staff.

GEO stores data range from DNA array experiment data to high-throughput sequencing experiment data like RNA-seq and the user is able to download the data sets in several different formats like SOFT, MiniMl, TXT or sometimes the user can also download the raw data if it's made available by the uploader. To make the work easier, GEO also permits the user to access the data through programming access thus the user can also benefit from NCBI eutils and direct download using users preferable programming language. Library like Biopython in Python and GEOquery in R were built for this purpose[3-4].

On the first week of the internship I tried to reproduce how Biopython works in a more specific (GEO-only) and user-friendly way by writing the code in Python. In this way I can improve my coding skill, learn deeper about GEO database and learn how the programmatic access of GEO NCBI works. For this purpose I use Python 3.4 as the programming language, code and example are available on the GitHub account (check materials section).

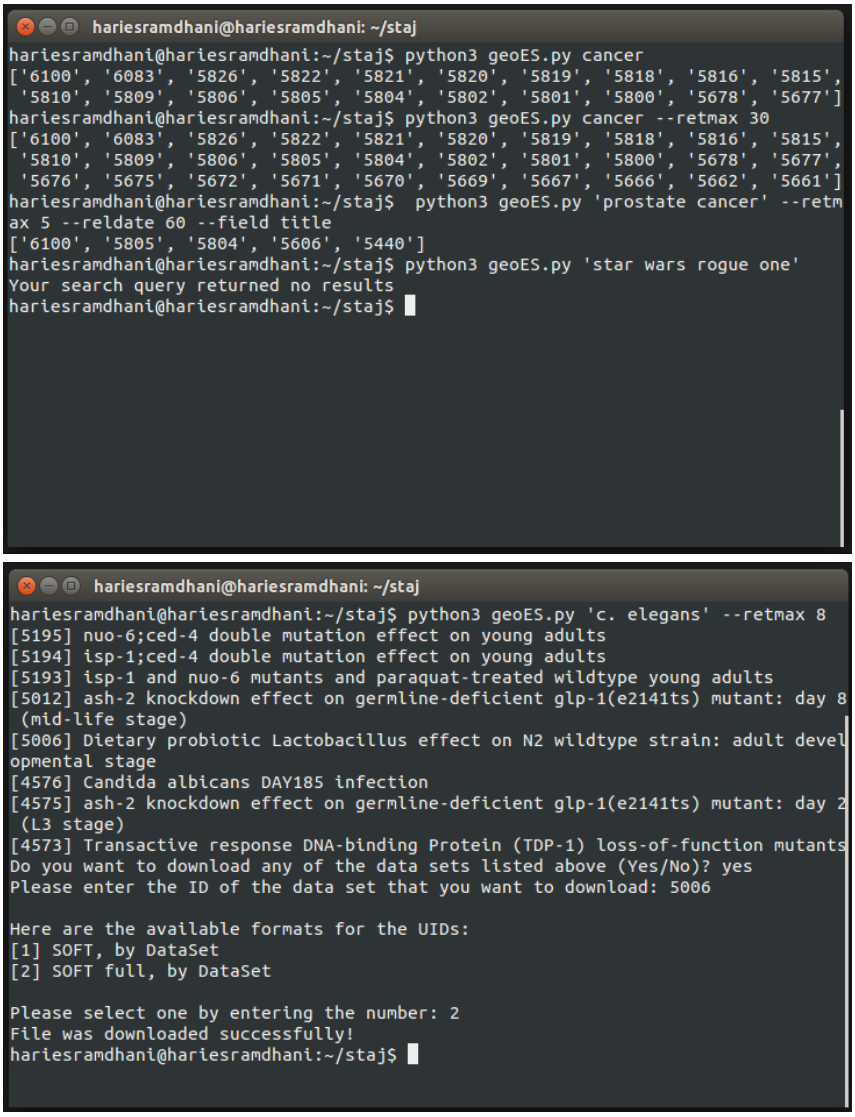


**Figure 1.**

Upper figure on the left is the first implementation of the code where the program can only return the unique IDs as the search result while lower figure on the left is the final version of the program where the program gets its bugs fixed and more user-friendly compared to the first one.

## RESULTS AND DISCUSSION

### 1. BUILDING THE CODE

The code was written in Python 3.4 because of its popularity and easy-to-learn syntax. In order to build this program, four libraries were also used before one of them ended up to be thrown away because it is not a Python built-in (*xml.dom*) module thus the user has to install it by themselves. *Re* module was used for works involving regular expression part, so it'll be easier to use than building a search tree from scratch, *argparse* was used to allow user input the argument from terminal and *urllib* was used to download the data sets from the FTP.

### 2. RETRIEVING DATA AND NCBI EUTILS

To retrieve the data from GEO server, NCBI already has the feature called Eutils[5]. Eutils allow us to search, retrieve the summary or even fetch the data available on NCBI Database, in this case GEO. What was done to the program is the url that was written Eutils was manipulated in several ways thus it can be used to search and download the data. For this purpose only three services were used, *Esearch*, *Esummary* and *Efetch*.

### 3. SIMPLE GDS-ID RETURNING PROGRAM

As the first step to build a better program, a simple program that only returns the GDS (GEO Datasets) ID as the search result was built (see Figure 1). This simple GDS-ID returning program was only able to use four positional arguments as this input, they are --*field* for limiting the search to specific NCBI field, --*retmax* to define the maximum number of

search results that want to be returned, --*datetype* to define the types of date, it can be either publication date or modification date

**4. FINAL VERSION OF THE PROGRAM**

The final version of the program is more user-friendly compared to the first one and also able to return the summary of the search, so instead of only returning the unique IDs it will also return the title of the study related to the search query **(see Figure 1)**. Another new implementation is the final version of the program is now able to download the datasets and --*reldate* as the to returns only those items within *–n* days away from the day the query was searched.
which is desired by the user either in SOFT or full SOFT format.

**5. DISCUSSION**

Since this is only trial to understand how GEO database works, the code was created to be as simple as it is. There are also similar projects on the GitHub which the author contributed too.

## MATERIALS

Code and documentation of the program can be downloaded through https://github.com/hariesramdhani . Code is available on the *src* folder while documentation of how the program works is available on the repository's wiki Day 0 and Day 1.

## REFERENCE


[1] Schena,M., Shalon,D., Davis,R.W. and Brown,P.O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science, 270, 467–470.

[2] Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository Nucleic Acids Res. 2002 Jan 1;30(1):207-10.

[3] Cock PJA, Antao T, Chang JT, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422-1423. doi:10.1093/bioinformatics/btp163.

[4] Davis S and Meltzer P (2007). "GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor." Bioinformatics, 14, pp. 1846–1847.

[5] Sayers E. E-utilities Quick Start. 2008 Dec 12 [Updated 2013 Aug 9]. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-.