2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.55

GeneInitiative: A Graphic Tool for Automatic Analysis of Large-scale Sequences

Kaikuo Xu[+], Hongwei Zhang, Jia He, Surong Zou, Wei Wei

College of Computer Science & Technology, Chengdu University of Information Technology
Chengdu, China

Abstract. GeneInitiative is an integrated graphic tool for automatic analysis of large-scale sequences through web interface. It takes advantage of both BLAST interface provided by NCBI (National Center for Biotechnology Information) and GO annotation interface provided by UDGenome project. Results are fetched from PubMed and UDGenome through information extraction (IE) and are reorganized into hierarchy, which can be used for further text mining. In addition, a result viewer for the visualization of the results is integrated into this tool. Its three modules (blaster, gofigurer, result viewer) can work independently and can be integrated into other systems.

Keywords: BLAST; GO; information extraction; visulization

1. Introduction

The BLAST (Basic Local Alignment Search Tool) [1] program was developed to perform a sequence similarity search. A powerful computer system dedicated to running BLAST has been established at NCBI [2]. NCBI provides a web interface (http://www.ncbi.nlm.nih.gov/blast/Blast.cgi) to access the BLAST system. As a web interface, at most one file composed of many sequences can be taken as input once (megablast).

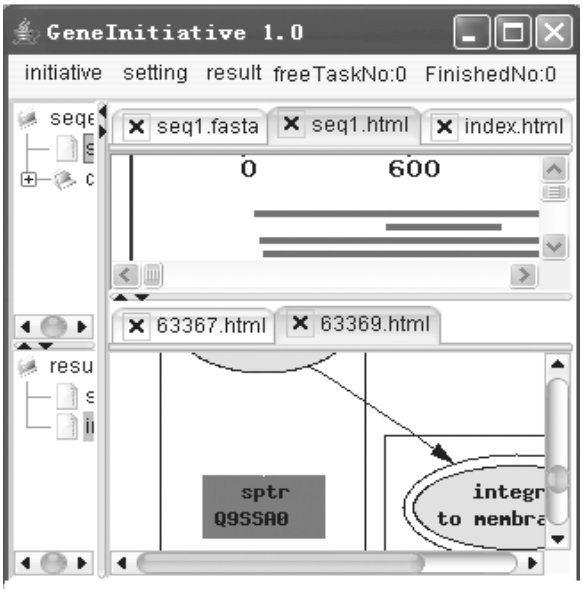

Fig.1: A screenshot of GeneInitiative GUI

[+] Corresponding author.
E-mail address: kaikuoxu@gmail.com

GO (Gene Ontology) [3] is one of the most important tools to assist in the task of representing and processing information about genes. It provides a controlled vocabulary for the description of cellular components, molecular function, and biological processed. GO annotation is used to provide descriptive terms from the Gene OntologyTM [4] controlled vocabulary. UDGenome [5] (A project designed to aid in the interpretation of unknown genomes) has left a web interface called GoFigure (http:// udgenome.ags.udel.edu/frm_go.html) to access their Go annotation system. As BLAST, only one sequence can be taken as input once.

In daily use of BLAST and GoFigure, researchers have to submit their sequences and get the results one by one manually. And when the scale increases, the time cost is unaffordable. To carry out automatic analysis of large scale sequences using both NCBI BLAST and GoFigure, store the results for further text mining, as well as provide a visual tool for result view, we developed a Java package, GeneInitiative. It can be used as a stand-alone GUI application, as shown in Figure 1, or its module may be integrated into other automated sequence analysis system.

2. Program overview

GeneInitiative integrates 3 modules (Java classes), a blaster, a gofigurer, a result viewer. Both blaster and gofigurer are composed of a querier, an extractor, and a fetcher. Before analysis, all parameters for this program should be configured. Then the program will run according to the configuration. The blaster reads input file, posts the sequence to the server, then extracts the urls of the result and fetches the result. The gofigurer roughly does the same thing as blaster, but the input is not from local files. Only the sequence of BLAST results, whose E value is the lowest, is taken as the input of GoFigure. Because in this program, the sequence, which is most similar to the input of BLAST, is considered to be the sequence that can most explain it. At last, the results can be browsed through the result viewer. The whole process is shown in Figure 2. The *italic* represents blaster, the **bold** represents gofigurer. In section III, the details of each module will be described one by one.

3. Module description

A. The Blaster

The blaster takes advantage of NCBI BLAST service (http://www.ncbi.nlm.nih.gov/blast/Blast.cgi).The querier utilizes “Protein” and “Translations” items. Other

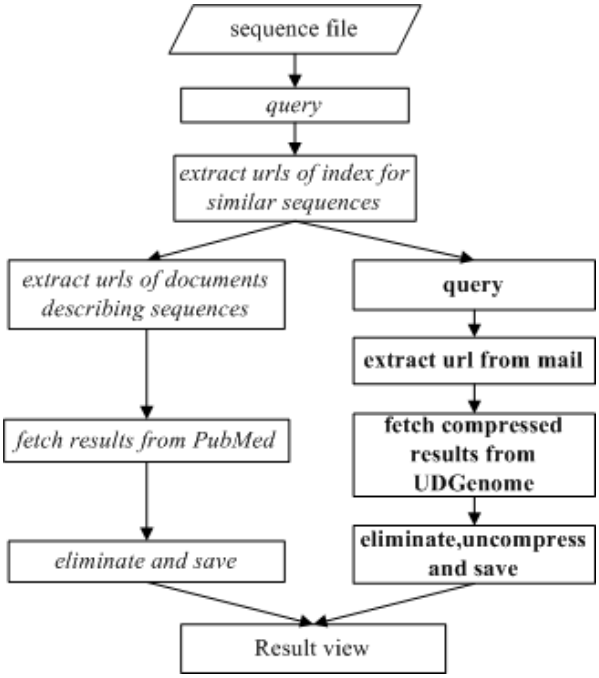

Fig.2: The process of sequence analysis

items can be added to this system, too. The extractor makes use of Information Extractor (IE) [6] technology, which is commonly used in text mining. It extracts urls from the html documents and hands them to the fetcher. Then the fetcher takes advantage of PubMed (http://www.ncbi.nih.gov/entrez/query.fcgi), downloading the results of PubMed queries to the local host. Before saving, the fetcher would eliminate redundant information. All the options for blaster are from [7][8]. This module provides an option called "result No": since many similar sequences will be found through blast, users can use this option to choose the top "N" sequences to be saved. Users could choose the blast according to their sequence type, as shown in table I.

TABLE I. THE RELATIONSHIP BETWEEN SEQUENCE TYPE AND BLAST

sequence type	BLAST
protein	blastp
nucleotide	blastx
protein	tblastn
nucleotide	tblastx

B. The gofigurer

The way gofigurer works is of little difference from blaster. But gofigurer needs to communicate with email server and eliminate redundant files from compressed results.

The email communication is implemented with [9]. All the options for gofigurer are from [5]. The blast item of GO annotation is determined by the type of BLAST in the program, as shown in table II.

TABLE II. THE RELATIONSHIP BETWEEN BLAST AND GO ANNOTATION

BLAST	GO annotation
blastp	blastp
blastx	blastp
tblastx	blastx
tblastn	blastx

C. The result viewer

The results of blaster and gofigurer are saved to the local file system automatically and reorganized into hierarchy as shown in figure 3. Note that figure 3(a) shows that the input files are also organized in hierarchy. A result directory corresponds to a single sequence. Let's take seq1.fasta as an example. The *italic* represents the result of seq1.fasta. "seq1.html" is the index of blast results under directory "seq1", which is created according to "seq1.fasta". "seq1" is composed of the gofigure results and the blast results. Each blast result is saved under "uid", which is a unique identification of the similar sequence and its gif. Dashed is used to refer to the "gif" because it may not exist for some sequences. "e" is the E value of the similar sequence. The results for other sequences are of the same architecture. This semi-structured file system is suitable for further text mining.

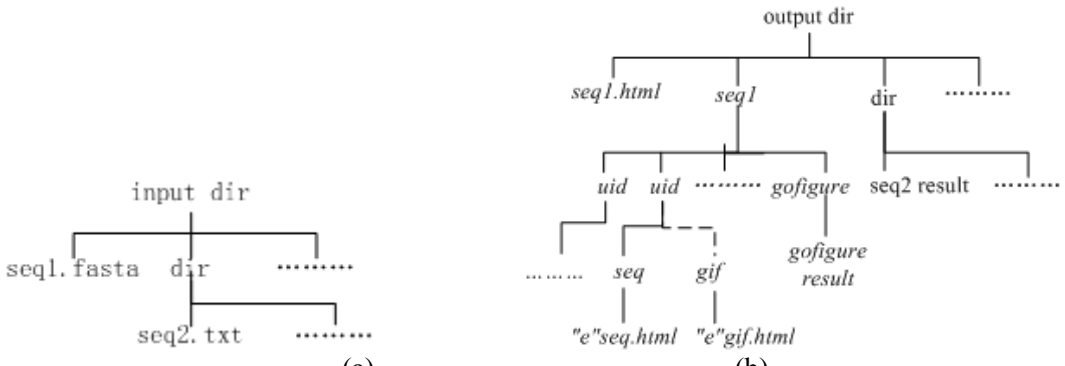

Fig.3: The hierarchy for the storage of results

Fig.4: The hierarchy for the storage of results

TABLE III. THE RELATIONSHIP BETWEEN BLAST AND GO ANNOTATION

Exception	Available	Reason
Network error	Y	
Power off	Y	
Wrong operation	Y	
Disk error	N	The file used for recovery may not be saved correctly
Changes made on BLAST server	N	GeneInitiative is based on permanent web interfaces

The result viewer utilizes the structure for the storage of the results. It visualizes the results through a rich-client program, as shown in figure 4. The results can be browsed through the result viewer as on the web, because we integrate the native browser into our program by JDIC [10]. Only Internet Explorer and Mozilla are supported. The panel in figure 4 is divided into 5 parts. ① shows the current status on finishing a given task. Since we only check the results, the status numbers are all "0". All the sequence files under the input directory could be seen through the "sequence" tree in ②. Users can check the gene initiative results for the sequences under the input directory like this: Click on a sequence file in ②. Then the content of the sequence file will show in ③. Both blast result index and gofigure result index could be seen through the "result" tree in ④. The file name of the blast result index is generated according to the sequence file name. In addition, a simple searcher is embedded into the viewer. Fig 4 shows an example on visualization of the results for seq2.txt.

- Blast result view

Click on the blast result index "seq2.html". The content will show in ③. You can browse the result as you did on the web.

Click on the hyperlink of the index. If the content of the URL is saved to the local disk, then it will show in ⑤; Else it will show on a new opened browser.

- Gene ontology result view

Click on the gofigure result index "index.html". The content will show in ③. You can browse the result as you did on the Internet.

Click on the hyperlink of the index. If the content of the URL was saved to the local disk, it would show in ⑤; Else it would show on a new opened browser.

D. The recovery system

GeneInitiative reads large scale sequences from input

files according to the configuration, and repeats the process in Figure 2. All the description above about the automatic analysis is under the normal condition: the network works well, the local disk has enough space, users' operation is right, i.e. never click on "exit" when system is running, and so on. But in real use, it won't be that ideal. Then problem arises: It's unavoidable that errors such as network error, power off may occur in the processing. Simply restarting the program will both put extra burden on the servers and is time-consuming. Therefore to conquer this problem, we developed a recovery system that can carry out "Analyzing Resuming", i.e. it can record the status of automatic analysis and restart the work from where it stops without any extra manual operation.

In this recovery system, a mechanism called "multilevel granularity" is adopted. This mechanism has been widely used in the field of granular computing and database application [11, 12, 13]. In GeneInitialtive, the automatic analysis results are partitioned into multiple levels of granularity as shown in figure 5. Granularities of different levels are adopted in normal result fetching and recovery. In normal result fetching, granularity of the second level is adopted, i.e., "Result for a sequence" is taken as the smallest unit. If any exception occurred when fetching any of its subunit, Geneinitiative would stop fetching all other subunits that have not been fetched yet, label the sequence as unfinished and ignore the labeled sequence this time. In recovery, granularity of the lowest level is adopted. For example, "Sequence html" is taken as the smallest unit instead of "Similar sequence". When Geneinitiative redoes the task for the sequence which is labeled as unfinished, it will fetch part of the results that have not been fetched instead of fetching the whole for the labeled sequence.

According to the test, GeneInitiative can recover from various exceptions. Table III shows the condition when it could recovery and the reason for why it can't.

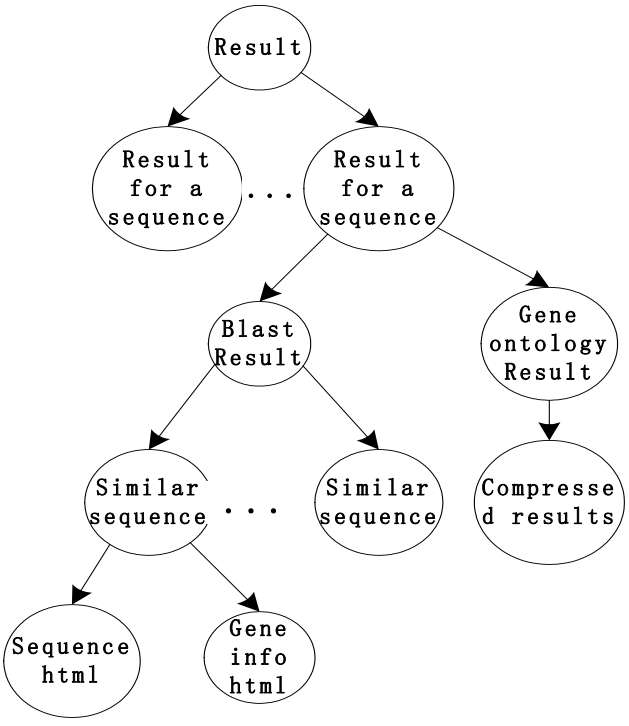

Fig.5: Multilevel granularity of the results

4. Conclusions

To perform automatic analysis on large-scale sequences,

we developed a software package called GeneInitiative. It is composed of three modules: blaster to perform the task of BLAST, gofigurer to perform the task of GO and result viewer to store the results in a hierarchy structure and visualize them in an embedded web browser. These three modules could be used as a

whole in our package or integrated into other packages. To keep the software's robustness, we adopt a mechanism called multilevel granularity. This mechanism guarantees that the system could recover from various exceptions.

5. Acknowledgment

This work was supported by NSFC Grant Number: 60773169, 11-th Five Years Key Programs for Sci. &Tech. Development of China under grant No. 2006BAI05A01 and the.Science Foundation of Chengdu University of Information Technology under Grand No. KYTZ200901.

6. References

[1] D.W. Mount (2004). "Bioinformatics: Sequence and Genome Analysis.". Cold Spring Harbor Press

[2] NCBI BLAST: http://www.ncbi.nlm.nih.gov/BLAST/

[3] Harris,M.A., Clark,J., Ireland,A., Lomax,J., Ashburner,M., Foulger,R., Eilbeck,K., Lewis,S., Marshall,B., Mungall,C. et al. (2004) The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res.,32, D258–D261.

[4] http://sourceforge.net/projects/geneontology/

[5] http://udgenome.ags.udel.edu:16080/gofigure/

[6] Ronen Feldman , Yonatan Aumann , Michal Finkelstein-Landau , Eyal Hurvitz , Yizhar Regev , Ariel Yaroshevich, A Comparative Study of Information Extraction Strategies, Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, p.349-359, February 17-23, 2002

[7] http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&LAYOUT=TwoWindows&AUTO_FORMAT=Semiauto&PAGE=Proteins&NCBI_GI=yes&HITLIST_SIZE=100&COMPOSITION_BASED_STATISTICS=yes&SHOW_OVERVIEW=yes&AUTO_FORMAT=yes&CDD_SEARCH=yes&FILTER=L&SHOW_LINKOUT=yes

[8] http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&LAYOUT=TwoWindows&AUTO_FORMAT=Semiauto&PAGE=Translations&NCBI_GI=yes&FILTER=L&HITLIST_SIZE=100&SHOW_OVERVIEW=yes&AUTO_FORMAT=yes&SHOW_LINKOUT=yes

[9] Sun Microsystems(2010). JavaMail: library providing a platform- independent and protocol- independent framework to build mail and messaging applications

[10] Sun Microsystems(20010). JDIC: library providing Java applications with access to functionalities and facilities provided by the native desktop

[11] Y.Y Yao, The art of granular computing, Proceedings of International Conference on Rough Sets and Emerging Intelligent System Paradigms (RSEISP'07), LNAI 4585, 101-112, 2007.

[12] Zeng, Y., Wang, Y., Huang, Z., Zhong, N.: Unifying web-scale search and reasoning from the viewpoint of granularity. In: Proceedings of the 2009 International Conference on Active Media Technology, Springer (October 2009) 418-429

[13] Yiyu Yao, Interpreting concept learning in cognitive informatics and granular computing, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, v.39 n.4, p.855-866, August 2009a