Intelligent Genomic Information Retrieval with Distributed System Resources

Wagner Arbex*
Leonardo Carvalho Nápolis Costa
Leonardo Mariano Gravina Fonseca
Brazilian Agricultural Research Corporation
Juiz de Fora, MG, Brazil

Camillo de Léllis Falcão da Silva
Federal University of Juiz de Fora
Juiz de Fora, MG, Brazil

Abstract

The National Center for Biotechnology Information (NCBI) website provides innumerable computational resources from several web services, which can be reached with some programming languages and/or using specific modules of theses programming languages. Nevertheless, the totality of these services or almost all services must be accessed from knowledge of metainformation about the data to be mined. Eventually, without knowing any information about what one wants to search, such as a single DNA sequence representing "string" by computational view, hindering access to NCBI's resources. This paper presents a set of bioinformatics resources to enable metainformation discovery of raw genomic data such as genetic sequences without any knowledge about them.

Keywords: Intelligent retrieval, genomic information, NCBI, BLAST, distributed system

1. Introduction

The methods of computerized information process are essential for conducting scientific research in all fields of knowledge, such as genomic or biomedical research, and the National Center for Biotechnology Information (NCBI) keeps the largest data repositories and bioinformatics computing resources worldwide. NCBI was established on November 4, 1988, as a division of the National Library of Medicine (NLM) at the National Institutes of Health (NIH) [1]. NCBI has experience in creating and maintaining biomedical databases, serving as a research program in computational molecular biology.

The NCBI aims at developing new information technologies to help and understand molecular and genetic processes that are connected to control health and disease. Moreover, the NCBI is coordinating efforts to gather biotechnology information, and create and maintain automated systems for storing and analyzing knowledge about molecular biology, biochemistry and genetics, thus facilitating the use of databases and software for searching the scientific community [1].

The issue featured in the Section 2 occurs when it is necessary perform searches without knowing any information about the data to be studied, because almost all NCBI's search services are based on some knowledge about the investigated data. This paper presents a solution for cases where the data investigated is raw nucleotide sequences raw and which are not known any information, even the organism of origin of sequences.

2. Problem Setting

The NCBI's responsibilities [1] are diverse, featuring several tools that allow you to use mathematical and computational methods, it helps maintain collaborations with several institutions, industry and other government agencies. Promoting scientific communication through meetings, workshops, lectures and training in support of basic and

*wagner.arbex@embrapa.br

applied research in bioinformatics. It has a variety of databases and software, in addition to developing and promoting standards for data storage and biological nomenclature.

The Entrez system [2] is an example of NCBI's resources, which support databases, named Entrez Databases. However, this resource does not only allows access to databases. Actually, the Entrez is a retrieval system designed for searching several linked databases. More specifically, Entrez is a search and retrieval system of NCBI that offers users integrated access to sequencing, mapping, taxonomy and structural data and provides graphical views of sequences, maps of chromosomes or other structures.

From of the NCBI unique identifier of data, can be reached innumerable information about the them. For instance, a NCBI unique identifier is the "accession number", which is "given to a sequence when it is submitted to one of the DNA repositories (GenBank, EMBL, DDBJ)" [3] and it is possible to access many information from Entrez resources.

Using Entrez resources can retrieve all information about a DNA sequence, or part of them. In the Figure 1, it can be seen which was recovered from the query [4]:

```
http://eutils.ncbi.nlm.nih.gov/entrez/
eutils/efetch.fcgi?db=nuccore&id=34577062,
24475906&rettype=fasta&retmode=text
```

```
>gi|34577062|ref|NM_001126.2| Homo sapiens (...), mRNA
GGAAGGGGCGTGGCCTCGGTCCGGGGTGGCGGCCGTTGCCGCCACCAGGGCCTCTTC
CTGCGGGCGGTGCTGCCGAGGCCGGCCTGCGCGGGGCAGTCATGGTACCCCCTTGAG
CGGGCTGTGGCGGAGAGCGGGGCGGGGACTGGCTGGAGGGTGGCGGCCCGGCGGGGC
GGGGGCGGGGCCGGCCTCTGGCTCCTTCTTCCTCTGCATGTGGCTGGCGGCCGCAGA
GCAGTTCAGTTCGCTCACTCCTCGCCGGCCGCCTCTCCTTCGGGCTCTCCTCGCGTC
ACTGGAGCCATGGCGTTCGCCGAGACCTACCCGGCGGCATCCTCCCTGCCCAACGGC
GATTGCGG
(...)
>gi|24475906|ref|NM_009417.2| Mus musculus (...), mRNA
CACAGAGAACACACCCAGCGGTGCACATCCTGCTTCTTCCGCTGCTGAGAAAAGGAA
AACGTTCAGCTCTAGAATGAGAACACTTGGAGCTATGGCAATAATGCTGGTGGTTAT
GGGAACTGTAATTTTCCTCTCTTTTATCCTGAGAAGCAGAGACATCTTGTGTGGGAA
GACCATGAAGTCCCATGTTATCAGTGCTGTGGAAACGAGCCAGCTCATGGTGGACCA
TGCAGTCTACAACACCATGAAAAGAAACCTCAAGAAAAGGGAAGTCCTTTCTCCAGC
CCAGCTTCTCTCTTTCTTTAAGCTGCCCGAGTCCACCAGTGGGGCTATTTCCCGAGC
AGCAGAGA
(...)
```

Figure 1. Retrieved information from Entrez query perform. The sequence of characters (...) indicates information edition to readability of the figure.

The Figure 1 shows all retrieved information NCBI's website, using **efetch.fcgi** function, performing a service provided by the Entrez system. It is possible see two FASTA[1] format sequences recovered by the query.

As can be seen, in the query you must inform identification parameters on what you want to look for, such as the part of the query

id=34577062,24475906

which determines two **GI** numbers.

The **GI** number is "Genbank identifier", that is another kind of NCBI unique identifier. In the specific case, only in this way it was possible to retrieve other information about the sequence, including the sequence itself.

2.1. Approaching the problem with BLAST program

If there was no knowledge of these sequences, the BLAST software could be used to find the first information about them.

BLAST is an acronym for Basic Local Alignment Search Tool and this software is available to use at the NCBI site in web version. Although BLAST originated at the NCBI, there are other bioinformatic websites, as well as the NCBI, provide bioinformatic computational resources, including BLAST software.

The BLAST software is able to sequence alignment and find similarity between them. Seting up similarity between sequences is a powerful tool for identifying the unknowns in the sequence world [5].

There are three ways to use the BLAST software: directly through NCBI's website, in the BLAST page, when it is possible submit a sequence or a list of sequence, "putting" one by one, or read all sequences from a FASTA file; the second way, it is necessary to get a version of BLAST program, e. g., available from NCBI website, and the particular BLAST file to the aligning process. To install the BLAST program on a local machine to run it using the BLAST file. In the third one, the BLAST process is running by remote call, like a remote procedure call (RPC) mode.

if the BLAST program was used directly from NCBI website, there are typical constraints of a website and web access. In the second case, if the choice is acquire the BLAST program and making

[1]The FASTA format is currently adopted as the standard format for genomic sequence data. The FASTA format consists of a *definition line* – which is a single line that begins with ">" – and the sequences lines, which are put immediately below to definition line. Can be used as many sequence lines as needed, but they should be limited numbers of characters [5].

download of the specific BLAST database necessary to the align, certainly a very good network, stable and with large bandwidth, it is essential, because the download of any BLAST database requires severe and large load data transmission.

2.2. Using Perl to access BLAST program from remote procedure call

Perl is the acronym to Practical Extraction Report Language and it is a programming language with many features to access databases, website developing, regular expression evaluation etc.. Originally, it was developed to text handling and today it keeps all of features, but is used for numerous kinds of different applications [6, 7].

One of the advantages of Perl is the CPAN (Comprehensive Perl Archive Network) website repository. The CPAN website stores a set of over 117 thousands Perl modules, for several distribuitions [8], organized, documented and free to be used by any developer.

In the CPAN repository it is possible to find several resources to build solutions for accessing remote database or others computational resources, including modules for distributed computing. For instance, the LWP [9], "The WWW library for Perl", enables the development of distributed computing procedures over HTTP protocol.

As it is known, to bioinformatics and computational biology the BioPerl module [10, 11] is the best resource provided from Perl. The BioPerl project "is an international open-source collaboration of biologists, bioinformaticians, and computer scientists (...)" [10] which developing a set of resources, as a "toolkit of Perl modules", to provide appropriate features to help bioinformatics application programmers to develop useful programming interface to access other resources, like database or web services.

The BioPerl module has several procedures to able access NCBI's services, and it is can run the BLAST program from remote call, among other procedures. Therefore, from Perl it is possible build a distributed system in the architecture client/server, executing remote calls, over HTTP protocol.

3. Genomic Information Mining and Retrieval

This paper shows how the Perl language can be used to get metainformation about a sequence, from NCBI website, without any a priori information on it. In particular, to this work, was used Perl with

```
(...)
# blastRemoteCall.pl
#
# Runs remote BLAST from NCBI, over HTTP protocol, using
# the methods of BioPerl module. Returns full information
# about searched sequence, if it was found. Currently
# works only with nucleotide sequences.
(...)
# perl blastRemoteCall.pl input_file_name output_base_file_name
(...)

use Bio::Tools::Run::RemoteBlast;
use Bio::SearchIO;
use Data::Dumper;
(...)

$prog = "blastn";
$db = "nr";
$e_val = "1e-10";

my @params = ( '-prog' => $prog,
               '-data' => $db,
               '-expect' => $e_val,
               '-readmethod' => 'SearchIO' );

my $remoteBlast = Bio::Tools::Run::RemoteBlast->new ( @params );
(...)

$remoteBlast->submit_blast ( $inpFile );

while ( my @rids = $remoteBlast->each_rid ) {

  foreach my $rid ( @rids ) {
    my $rc = $remoteBlast->retrieve_blast ( $rid );

    if ( !ref ( $rc ) ) {

      if ( $rc < 0 ) { $remoteBlast->remove_rid ( $rid ); }

      print STDERR ".";
    }
    else {
      my $result = $rc->next_result();

      my $outFile = $outBaseName.$result->query_name()."\.txt";
      $remoteBlast->save_output( $outFile );
      $remoteBlast->remove_rid( $rid );

      print "\nQuery name: ", $result->query_name(), "\n";

      while ( my $hit = $result->next_hit ) {
        print "\thit name is ", $hit->name;

        while( my $hsp = $hit->next_hsp ) {
          print " (score ", $hsp->score, ")";
        }

        print "\n";
      }
    }
  }
}
exit;
```

Figure 2. Listing of blastRemoteCall.pl script. The sequence of characters (...) indicates information edition to readability of the source code or to anonymity of authorship of the paper.

BioPerl module to call BLAST remote procedure, as previously explained (Sections 2.1 and 2.2).

As can be seen in the source code of blastRemoteCall.pl script, listing at Figure 2, were used the modules Bio::Tools::Run::RemoteBlast and Bio::SearchIO, which form part of the BioPerl project, and Data::Dumper.

The adopted strategy for blastRemoteCall.pl script approach was to get unknown sequences in a the FASTA format and submitted them to NCBI's BLAST program from remote call. The Bio::Tools::Run::RemoteBlast module, in this script, is the most important for script strategic role and its need to use, because the method which performs the remote BLAST is implemented in the Bio::Tools::Run::RemoteBlast module.

The necessary parameters to run the remote BLAST are setting to **@params**, which it is used to instance a new "object" nominated **remoteBlast**. These parameters assign, among others features, the "kind" of BLAST will be used and on which database will be done the search. In the blastRemoteCall.pl script, these parameters received **blastn** and **nr** values, specifying the BLAST program and database for nucleotides, respectively.

Next, the BLAST submission, actually, remote submission is done with the sequences to be investigated sent in FASTA files (Figure 3) as a submission parameter, and each the BLAST process for each sequence received a unique identification, known as **rid**, which are stored in **@rid**.

```
>
GGAAGGGGCGTGGCCTCGGTCCGGGGTGGCGGCCGTTGCCGCCACCAGGGCCTCTTC
CTGCGGGCGGTGCTGCCGAGGCCGGCCTGCGCGGGGCAGTCATGGTACCCCCTTGAG
CGGGCTGTGGCGGAGAGCGGGGCGGGGACTGGCTGGAGGGTGGCGGCCCGGCGGGGC
GGGGGCGGGGCCGGCCTCTGGCTCCTTCTTCCTCTGCATGTGGCTGGCGGCCGCAGA
GCAGTTCAGTTCGCTCACTCCTCGCCGGCCGCCTCTCCTTCGGGCTCTCCTCGCGTC
ACTGGAGCCATGGCGTTCGCCGAGACCTACCCGGCGGCATCCTCCCTGCCCAACGGC
GATTGCGG
(...)
```

Figure 3. FASTA sequence file sample for BLAST remote submission on blastRemoteCall.pl script. The sequence of characters (...) indicates information edition to readability of the figure.

One by one, the BLAST process are performed and their outputs are tested to check if is valid the returned content.

Could be observed in example (Figure 3), the definition line – marked with '>' – is empty, because its have no information about the sequence.

4. Approach Analysis

The subject of this article proposes to make remote access on NCBI's website to use their web services and perform BLAST program to get information about unknown sequences in FASTA format. Particularly, these sequences could be nucleotide sequences, as a DNA or RNA sequences.

To implement the remote procedure call was developed a Perl script, which uses modules of BioPerl project to set up the BLAST parameters; executes and submits the BLAST program to mining the database; and retrieve the answer.

The means adopted to implements this strategy was process remote calls as a RPC. The RPC are a way to implement distributed systems and use explicit calls to execute send and receive procedures [12, 13].

The interaction between send and receive procedures – or *message passing* between them, as it is technically called this kind of interaction in RPC systems – define the type of communication between the client, which performs the request, and the server, which replies to the request.

The client and server doing *synchronous communication* if the client waits for the server response, but if the client performs a request and continues processing, without waits for the reply, then they doing *asynchronous communication* [12, 13].

Clearly, the blastRemoteCall.pl script implements asynchronous communication mode. Therefore, as the BLAST server is able to receiver and handle each request independently of the others and is not necessary for it waits for the reply, then the entire process becomes more efficient.

The method of communication implemented with distributed system concepts creates a lightweight and efficient message passing between the Perl script, client application, and the server on NCBI's website.

5. Conclusions

This paper introduces and analyzes an intelligent retrieval for metadata of unknown sequences with a Perl script using Bioperl and other modules as well as resources of the NCBI website.

Search for metadata about sequence fragments can be necessary many times on bioinformatic and computational biology tasks and the NCBI website is an important place to do prospecting like these and it offers many efficient tools to do it.

However, almost all NCBI's tools need a kind identifier for the sequence which look up its metadata, usually one of NCBI's unique identifier, like the **GI** number.

Nevertheless, it is not uncommon encounter complete sequences or fragments of sequences without identifiers or knowing any information about them. Therefore, many NCBI's tools are not as useful in these searches, but the solution proposed shows another approach.

The search for metadata is done from remote execution of the BLAST alignment program, enabling the recovery all the information found by search of similarity among sequences.

Acknowledgments

The authors thanks to reviewers who gave useful comments, and would like to express thanks to the State of Minas Gerais Research Support Agency (FAPEMIF) for the partial support for the accomplishment of this paper.

References

[1] NCBI. (2013, Jan.) Our mission. [Online]. Available: http://www.ncbi.nlm.nih.gov/About/glance/ourmission.html

[2] ——. (2011, Dec.) Entrez help: NCBI bookshelf. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK3837/

[3] ——, *The NCBI Hanbook: Glossary*, J. McEntyr and J. Ostell, Eds. Bethesda: NCBI, 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK21106/

[4] E. Sayers, "E-utilities quick start," in *Entrez Programming Utilities Help*, J. McEntyr and J. Ostell, Eds. Bethesda: NCBI, 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK25500/

[5] I. Korf, M. Yandell, and J. Bedell, *BLAST*. Sebastopol: O'Reilly & Associates, Inc., 2003.

[6] T. Christiansen, B. Foy, and L. Wall, *Programming Perl*, 4th ed. Sebastopol: O'Reilly Media, Inc., Feb. 2012.

[7] Perl.org, "The perl programming language," Website, 2013. [Online]. Available: http://www.perl.org/

[8] ——, "The comprehensive Perl archive network," Website, 2013. [Online]. Available: http://www.cpan.org/

[9] G. Aas and M. Koster, "The WWW library for Perl," Website, 2009. [Online]. Available: http://search.cpan.org/perldoc?LWP

[10] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehvaslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney, "The Bioperl toolkit: Perl modules for the life sciences." *Genome Res*, vol. 12, no. 10, pp. 1611–1618, Oct 2002.

[11] BioPerl Core Developer, "BioPerl," Website, Aug. 2012. [Online]. Available: http://www.bioperl.org/

[12] J. Dollimore, T. Kindberg, and G. Coulouris, *Distributed systems: concepts and design*, 4th ed. Addison Wesley, 2005.

[13] A. S. Tanenbaum and M. Van Steen, *Distributed systems: principles and paradigms*, 2nd ed. Prentice Hall, 2006.