# Biological Data Analysis using DDBJ Web services

**Hideaki Sugawara[1], Satoru Miyazaki[2], Takashi Abe[1] and Yasumasa Shigemoto[3]**
[1]*Center for Information Biology and DDBJ, National Institute of Genetics and SOKEN-DAI*
[2] *Faculty of Pharmaceutical Science, Tokyo University of Science*
[3]*Fujitsu Inc.*
*E-mail: hsugawar@genes.nig.ac.jp*

**ABSTRACT**: We demonstrate workflows in biological data retrieval and analysis using the DDBJ Web Service; specifically introduce a workflow for the analysis of proteins or proteomics data sets. The workflow mechanically extracts the gene whose protein structure and function are known from all the genes of a human genome in Ensembl (http://www.ensembl.org/) based on cross-references among Ensembl, Swiss-Prot (http://www.ebi.ac.uk/swissprot/) and PDB (Protein Data Bank; http://www.wwpdb.org/). The workflow discovered "hidden" linkages among databases. We will be able to integrate distributed and heterogeneous data systems into workflows, if they are provided based on standards for Web services.

## 1 INTRODUCION

A number of diverse data resources (namely, databases and data analytical tools) for molecular biology are now available from Web sites in the Internet [1]. They cover data of tiered biological objects from genes and genomes, amino acid sequences, tertiary structure of proteins, pathway and up to phenotypes.

We can now easily access multiple Web sites with Web browsers, e.g.:

a) Connect to a journal database to find accession numbers of the International Nucleotide Sequence Databases (INSD) of DNA Data Bank of Japan (DDBJ), EMBL/EBI (http://www.ebi.ac.uk/) and GenBank/NCBI (http://www.ncbi.nlm.nih.gov/)
b) Move to INSD to search amino acid sequences by the accession numbers found in the step a)
c) Move further to the Protein Data Bank (PDB) to get the 3D structure.

The three steps in the above are accomplished only by click, copy and paste by mouse. However, this mode of operation is not scalable and programmatic interfaces for mechanical processing of data resources have been demanded very much these years. In other words, many users want to send queries and process their results by their computer programs in successively connecting to multiple data resources. The solution to this issue will be given by Web services [2]. The comparison of conventional Web sites and Web services is summarized in Table 1. The conventional Web sites are suitable for manual surfing and the Web services for the utilization of data resources by users' program.

| | Conventional Web sites | Web services in DDBJ |
|---|---|---|
| Interface for: | manual operation | computer program |
| Access with: | browser | SOAP library |
| Locator: | URL | WSDL |
| Search by: | search engine | UDDI |

Table 1: Comparison of conventional Web sites and WebServices

Therefore, we at DDBJ have introduced XML (eXtensible Markup Language) technology、SOAP (Simple Object Access Protocol)、WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration) [3,4] to prepare programmatic interfaces to DDBJ data resources. The Web services are freely available at http://www.xml.nig.ac.jp/index.html. In the meantime, we have also developed workflows composed of methods of multiple Web services in DDBJ.

## 2 WEB SERVICES AVAILABLE FROM DDBJ

The Web services in DDBJ are prepared for: Blast and Fasta for homology search programs, ClustalW for the multiple alignment program, Ensembl and UniProt databases from EBI, Getentry for the retrieval of INSD entries by accession numbers, GIB (Genome Information Broker) of a complete microbial genome database, GTOP (the Genes TO Protein) database, NCBI Genome Annotation and RefSeq from NCBI, PML (Polymorphisms Markup Language), SPS of the splicing profile analysis, SRS of a key word search program, TxSearch for searching lineage of organisms, and VecScreen for vector screening.

Web services have multiple methods. For example, the Blast Web services includes three methods as introduced in the following list:

a) searchSimple(program, database, query)
Function: execute BLAST specified with program, database and query.
Parameters
program - specify blastn, blastp, blastx, tblastn or tblastx
database - specify database
query - query sequence
Example
*searchSimple("blastp", "SWISS",*

*"MSSRIARALALVVTLLHLTRLALSTCPAACHCPL EAPKCAPGVGLVRDGCGCCKVCAKQL")*

b) searchParam(program, database, query, param)

Funstion: Execute BLAST specified with program, database, query and parameters.

Parameters

program - specify blastn, blastp, blastx, tblastn or tblastx
database - specify database
query - query sequence
param - parameter at blast execution

Example

*searchParam("blastp", "SWISS", "MSSRIARALALVVTLLHLTRLALSTCPAACHCPLE APKCAPGVGLVRDGCGCCKVCAKQL", "-b 5 –m 7")*

c) extractPosition(result)

Function: parse a BLAST result and extract the position of the matched sequence.

Parameters

result - a result of BLAST

Example

*extractPosition($result)*

# 3 INFORMATION ENVIRONMENT FOR THE UTILIZATION OF WEB SERVICES

The DDBJ Web services utilize a SOAP server to wrap the data resources in DDBJ as illustrated in Figure 1.

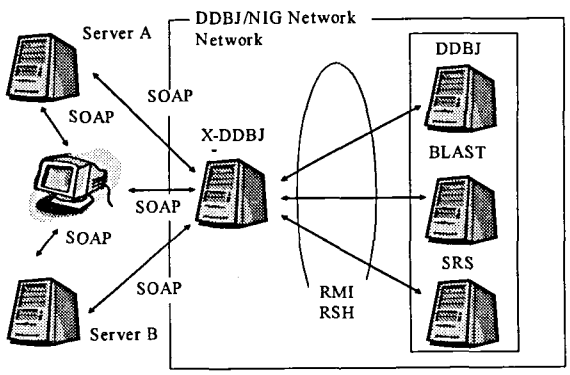


Figure 1: Structure of Web services at DDBJ

The DDBJ data resources such as DDBJ, Blast and SRS at the right hand side in Figure 1 are accessible through SOAP server of X-DDBJ in the middle from user programs in the client on the left.

The client computer has to have libraries to access the SOAP server. In the case of calling Web services from Perl program in WINDOWS PC, it is simple to prepare the environment by just downloading ActivePerl for WINDOWS from http://www.activestate.com/. Once ActivePerl is installed, users are requested essentially to specify a WDSL file and call a method. In the following example, the method of getXML_DDBJEntry(accession) ofthe Getentry Web services is called from a Perl program:

```
#!/usr/bin/perl
# include SOAP Lite
use SOAP::Lite;
# specifies WSDL file
$service = SOAP::Lite ->
service('http://xml.nig.ac.jp/wsdl/GetEntry.wsdl');
# call SOAP service
$result = $service->getXML_DDBJEntry("AB000003");
```

You can easily expand this sample program to get multiple entries from AB000002 to AB000005:

```
$result .=
$service->getFASTA_DDBJEntry("AB000002");
$result .=
$service->getFASTA_DDBJEntry("AB000003");
$result .=
$service->getFASTA_DDBJEntry("AB000004");
$result .=
$service->getFASTA_DDBJEntry("AB000005");
```

# 4 CONSTRUCTION OF WORKFLOWS BASED ON METHODS IN DDBJ WEB SERVICES

We will introduce workflows that we have developed by use of the DDBJ Web services in this section

## 4.1 DDBJ-Swiss-prot workflow

This workflow finds linkages between nucleotide sequences and proteins in the amino acid sequences database of Swiss-Prot in the following steps.

a) acquire accession numbers from the INSD in DDBJ by key word search with the SRS Web services
b) capture amino acid sequences of coding regions of the INSD entries form the step a) in FASTA format by the Getentry Web services
c) carry out homology search of the sequences in the step b) against Swiss-prot database by the Blast Web services
d) retrieve Swiss-prot entries that were hit in the step c)
e) prepare a table of accession numbers of the INSD entries, protein IDs, Swiss-prot IDs, protein symbols and definition of proteins. The source code in Perl is introduced in the following:

**Step a)** calls the SRS Web services to retrieve entries that have key words of 'prion',Human Division, Molecule kind'mRNA'

```
use SOAP::Lite;
# specifies WSDL file
$service = SOAP::Lite ->
service('http://xml.nig.ac.jp/wsdl/SRS.wsdl');
# call SOAP Service
$result = $service->searchSimple("[ddbj-AllText:prion*]
& [ddbj-Division:hum] & [ddbj-Molecule:mrna]");
```

**Step b)**

```
# make arrangement divided by the line feed code
@id = split "¥n", $result;
$getentry = SOAP::Lite ->
service('http://xml.nig.ac.jp/wsdl/GetEntry.wsdl');
```

```
$result = "";
for($i = 0; $i < $#id; $i++) {
#get accession number
$acc = substr($id[$i], 5);
#get DAD Entry by FASTA Format.
$result .=
$getentry->getFASTA_DADEntry($acc);
}
```

**Step c)** performs Blastp against Swiss-prot database

```
$blast = SOAP::Lite ->
service('http://xml.nig.ac.jp/wsdl/Blast.wsdl');
#call BLAST service
$result = $blast -> searchParam("blastp", "SWISS",
$result, "-m 8");
```

**Step d and e)** use GetEntry to get annotations in entries hit by Blastp in the step c) and output accession number, protein ID, Swiss-Prot ID, protein symbol and protein definition by tabular.

```
# make arrangement divided by the line feed code
@blastline = split(/¥n/,$result);
for($i = 0; $i < $#blastline; $i++) {
$swissid=&get_SwissId($blastline[$i]);
$swissentry = $getentry ->
getSWISSEntry($swissid);
print &get_list($blastline[$i],$swissentry);
}
#sub routine: get Swiss-Prot ID
#parameter : Result of BLASTService.
#return : Swiss-Prot ID
sub get_SwissId{
local $blast_result = @_[0];
local @blast_list=split(/¥|/,$blast_result);
return $blast_list[3];
}
#sub routine : get tabular format data;
#@parameter 1: Result of BLAST Service;
#@parameter 2: Result of GetEntry;
#@return : tablulr format data;
sub get_list{
local $blast_result = @_[0];
local $getEntry_result=@_[1];
local @blast_list=split(/¥|/,$blast_result);
$getEntry_result =~ s/¥s{2,}//g;
local @getEntry_list =
split(/¥n/,$getEntry_result);
local $protein_sysmbol = "";
local $definition = "";
for($j=0;$j< @getEntry_list;$j++){
if(substr($getEntry_list[$j],0,2) eq
'ID'){
local
@protein_symbol_list = split(/¥s/,$getEntry_list[$j]);
$protein_sysmbol=$protein_symbol_list[1];
}
if(substr($getEntry_list[$j],0,2) eq
'DE'){
$definition=substr($getEntry_list[$j],2);
break;
}
}
return
$blast_list[0]."¥t".$blast_list[1]."¥t".$blast_list[3]."¥t".$p
rotein_sysmbol."¥t".$definition."¥n";
```

## 4.2 Ensembl – PDB workflow

This workflow identifies correspondence between all of the human genes in Ensembl and 3D structure of proteins in PDB by referring to cross-reference in INSD and UniProt/SwissProt in the following steps:

a) prepare a list of PDB entries that have cross-references to UniProt/Swiss-Prot by use of the SRS Web services
b) retrieve the cross-reference information of UniProt/Swiss-Prot entries found in the step a) by use of the Getentry Web services
c) prepare a list of UniProt/Swiss-Prot entries that have cross-reference to PDB by use of the SRS Web services
d) retrieve the cross-reference information of PDB entries found in the step c) by use of the Getentry Web services
e) integrate the results in step b) and d)
f) acquire all the human genes from Ensembl by use of the Ensembl Web services
g) integrate the results of the step e) and f)

Table 2 is the summary of linkages among INSD, UniProt/Swiss-Prot and PDB recognized by only Ensembl and the workflow.

| Linkage by | INSD | Uni-Prot/Swiss-Prot | PDB |
|---|---|---|---|
| a) | 18,695 | 18,713 | 1,273 |
| b) | 18,695 | 18,713 | 1,320 |
| c) | 19,462 | 19,477 | 1,390 |

Table 2: The correspondence among INSD, Uni-Prot/Swiss-Prot and PDB: a)= only by Ensembl (Nov., 2004), b)= by the workflow (Nov., 2004), c)=by the workflow (July, 2005).

It is obvious that the workflow recognized correspondence between INSD entries and PDB entries that are not given from Ensembl only.

## 4.3 Other Workflows

In addition to the workflows in the section 4.1 and 4.2, the following workflows are usable at http://www.xml.nig.ac.jp/workflow/index.html:

a) Blast-ClustalW WorkFlow
   Run blastn against DDBJ database with a given DNA sequence and compare the alignment regions of high similar sequences by using ClustalW.
b) BLAST WorkFlow
   Run multiple blast against DDBJ, Uniprot-swissprot and PDB continuously with a sequence of an Accession number.
c) E.coli bacteria genome work flow
   Map all E.coli piece entries registered in DDBJ to the genome sequence.
d) E.coli bacteria genome annotation WorkFlow

Evaluate the annotation of E.coli genome by using the workflow of mapping piece entry,

e) Splicing workflow
Compare the similarities between splicing structure and homology by using the genes of human and mouse.

The list in the above will be expanded eventually

## 5 CONCLUSIONS

In the age of OMICS, various types of large-scale data resources will be available in the Internet and wait for utilization. If all the data sources are available as Web services, users are able to write a program that: identifies Web services in the directory, binds to Web services and fetch results into local databases. XML, SOAP and Web services will greatly expand the world constructed on HTML and HTTP. Therefore, public data sources are requested to provide Web services and register them into a directory. Workflows can also be deposited in a directory and re-used by user communities. Then workflows of workflows will be developed further. Web services are actually accessible from other sites than DDBJ: BioMoby [5], PDBj (http://pdbj.protein.osaka-u.ac.jp/SOAP/), KEGG (http://www.genome.jp/kegg/soap/) and so on [7-9]. In this way, the world of Web services is self-expanding.

## ACKNOWLEDGEMENTS

The development of the DDBJ Web services has been supported partly by "Research and Development of Biological Portal Site of the New Generation" project through the Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government and also by the Institute for Bioinfomatics Research and Development (BIRD) of Japan Science and Technology Agency (JST).

## REFERENCES

[1] Nucleic Acids Research: Database Issue. Nucleic Acids Research, Vol. 33, pp. D1-D679, (2005).

[2] L. Stein. Creating a bioinformatics nation. Nature, 417: 119--120, 2002.

[3] H. Sugawara and S. Miyazaki. Biological SOAP servers and web services provided by the public sequence data bank. Nucleic Acids Research, 31:3836-3-839, 2003.

[4] H. Sugawara and T. Gojobori (edts). *Utilization of* DDBJ services (in Japanese), Kyoritsu-shuppan, Tokyo, 2003.

[5] M. D. Wilkinson and M. Links. BioMoby: an open source biological web services proposal. Brief. Bioinf., 3:331--341, 2002.

[6] L. Wang, J. J. M. Riethoven and A. J. Robinson. XEMBL - distributing EMBL data in XML format. Bioinformatics, 18: 1147--1148, 2002.

[7] J. Wang and Q. Mu. Soap-HT-BLAST: high throughput BLAST based on Web services. Bioinformatics, 19:1863--1864, 2003.

[8] T. M. Casstevens and E. S. Buckler. GDPC: connecting researchers with multiple integrated data sources. Bioinformatics, 20:2839--2840, 2004.

[9] F. Iragne, A. Barre, N. Goffard and Daruvar, A. De. AliasServer: a web server to handle multiple aliases used to refer to proteins. Bioinformatics, 20:2331—2332, 2004.