InBit: An Integrated Bioinformatics Toolkit for Biologists

Ranojit Sarker[1], Sanghamitra Bandyopadhyay[2] and Ujjal Maulik[3]

[1]*Dept. of biotechnology, Institute of Technology and Marine Engineering DH Road, 24 PGS(s), India*
[2]*Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India*
[3]*Dept. of Computer Science and Engineering, Jadavpur University, kolkata, India*

ranojitsarker@gmail.com, sanghami@isical.ac.in, drumaulik@cse.jdvu.ac.in

Abstract

Here we report on the development of an Integrated Bioinformatics Toolkit (InBit) that is application-based software, integrating various bioinformatics programs for general purpose biological data analysis like gene prediction, gene analysis, protein analysis, sequence alignment, etc. InBit will enable the biologists to use the various programs for their research purpose without being bothered with their various technical aspects. It will also help graduate students and those who are newly using bioinformatics tools. Though the individual software that is integrated in InBit are available on the web and are quite user-friendly, they often become difficult to use when the size of the data set becomes quite large. In these cases, it becomes necessary to use standalone versions of many of this software, but this requires some prior knowledge of UNIX operating system. InBit merges the standalone properties of the software in a user-friendly menu-driven environment. It is open-ended UNIX/LINUX based software that can be extended by the user if so desired. Its hardware requirements are also very basic, requiring a optimum of 1GB RAM for efficient performance.

Keywords: *Bioinformatics tools, Biosoftware, Gene analysis toos, protein analysis tools*

1. Introduction

Computational analysis of biological data has been practiced for a long time, but this was a minor interest until advances in computer science and sequencing technology led to a rapid extension in the number of stored sequence databases such as GenBank, EMBL, and Swiss-Prot etc. Now with the revolution of computer science various tools for exploiting biological data i.e., gene and protein sequence and structure prediction and analysis, gene expression profiles, alignment tools and structure visualization tools are available and also increasing in number day by day. With the advances of the technology this scientific domain is not only restricted to research institutions but also has spread over several undergraduate institutions. The research scientists and students from life sciences domains having very basic of computational knowledge use these large numbers of bioinformatics tools. The followings reasons trouble them to use bioinformatics tools:

1. Individual tools are available at various servers, so it difficult and time consuming for one to find out them to use.
2. Most of the programs are web based. Because of large input data, when computation becomes extensive, the tools become very difficult to use and requires stand-alone version of the program.

3. Individual stand-alone programs have complex installation and utilization procedures.
4. Large number of the public domain tools is Unix/Linux platform based, which is not also a popular operating system for the bioinformatics tools user group.

Few attempts [Biegert et al 2006, Rampp et al 2006, Gracy and Chiche 2005,] have been taken to solve one of the afore mentioned problem i.e., integrate the tools on a single platform, where the user can find the necessary tools required by him. But the other problems still remain because of the fact that all the available integrated tools kits are available only web based. So where Internet facility is not available and also the speed of the Internet is not good people face a lot of trouble to utilize the tools.

So, we tried to develop software, InBit that aims to

1) Collect several basic standalone bioinformatics software and subsequently bring them over a single platform that is LINUX based.
2) Make the integrated toolkit stand alone, so it can be usable without Internet accessibility
3) Distribute the entire toolkit freely; so that researchers and research students in the life science domain be able to use different data analysis tools easily.

2. Features

2.1. System Recommendation

The program can be run in any LINUX / UNIX system. InBit requires perl and Perl Tk software installed in the LINUX / UNIX system. Hardware requirement is very basic and optimum RAM requirement is 1 GB.

2.2. Tools

We have integrated basic Bioinformatics tools for gene prediction, gene analysis, protein analysis, alignment tools for pair wise and multiple alignments and visualization tool. The program is open ended and any number of tools can be added or removed if required by the user.

2.3. Basic User Interface

All the tools are organized in to several classes and put in different menu on the Tool Bar. Programs in the same class are put in the submenu. User just needs to select specific program from the menu and submenu to execute that program. A file managing option is also provided on the Tool Bar. All basic file-managing tasks can be managed using the file menu like opening a sequence file or saving an output file. The user interface is provided with a text area where user can paste input sequence and also with a browsing option to include the input file. After coping input file in the text area or browsing it one need to select a specific program for the program menu, then the program is executed in the back end and the output file is generated in the text area. User than can manipulate that file using file menu.

3. Functionalities

Following flowchart explain the functionalities of the InBit:

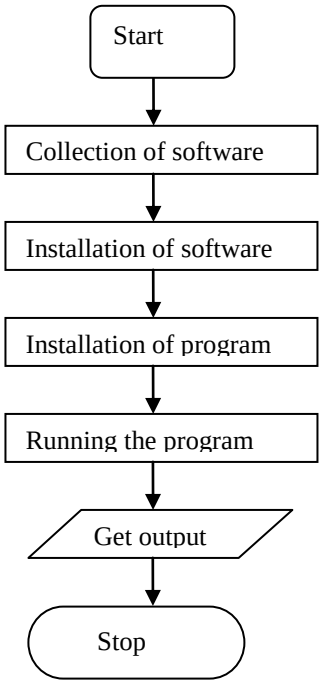

Figure 1: Overall Steps to Execute InBit

Example: Execute merger program to merge two sequences.

User will reach the InBit interface by executing the following command.

perl <path>InBit

This command will automatically compile and run the program.

Then the following steps that are to be followed to run the program:

Step 1: From the menu bar (see Fig. 2) gene analysis option need to be selected under which there is the target program merger, which merges 2 sequences.

Step 2: Clicking merger will open the merger page. The form like page needs to be filled up with appropriate data by the user.

Step 3: Mandatory fields cannot be remaining blank. Input sequences must be in fasta format. In case there is mistake it will show error.

Step 4: Other options like advanced and associated qualifiers may be filled up as per user requirement and interest.

Steps 5: After filling the form clicking on run button will give the output in separate window. This can further be saved in any place.

4. Installation of Software

4.1. Installation of PERL/Tk

Step - by - step the commands to build the Tk extension to Perl are (for the dynamically linked version) roughly as follows:

Unpack Perl/Tk outside the Perl distribution (i.e. outside the perl build, perl install, or perl lib areas).

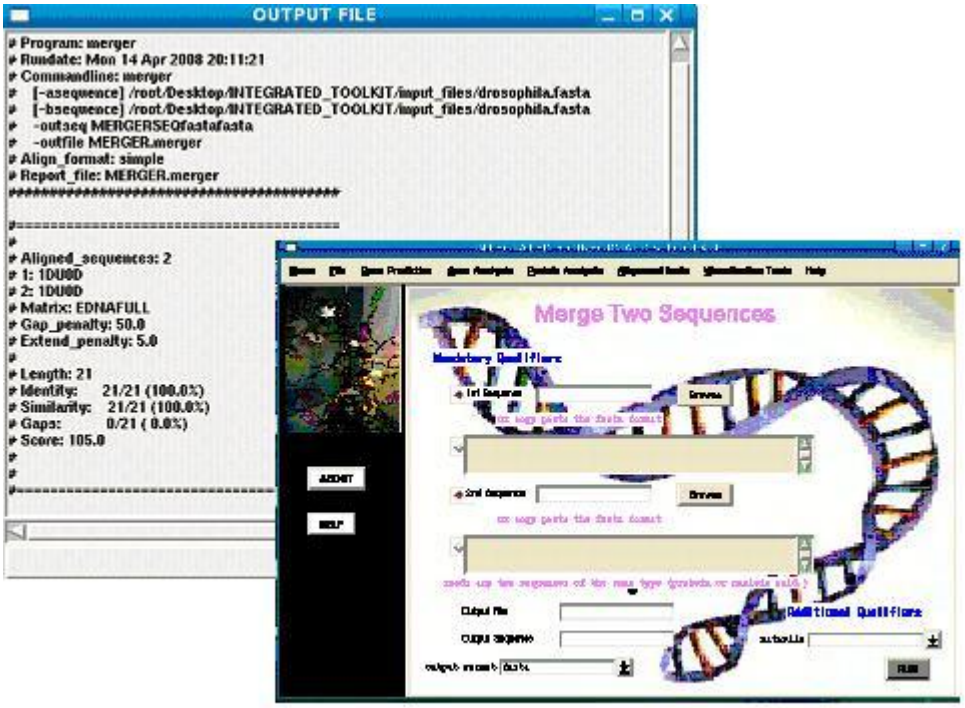

Figure 2: This figure explains how to find a merger program from the user interface and execute it to merge to sequences.

GunzipTk-804.027.tar.gz

tar -xvf Tk-804.027.tar

Change to the directory distribution unpacks to

cd -804.027

Compilation

Have perl generate a custom Makefile.

#perl Makefile.PL

If Makefile.PL reports that it cannot find X, version on the command line can be specified: perl Makefile.PL X11=/usr/local/X11R5

If X's include and lib are not under a common parent they can be specified separately:

perl
Makefile.PLX11INC=/usr/local/share/X11R5/includeX11LIB=/usr/local/arch/X11R5/lib

Building

make

make test

Now installation can be done by using one of the make targets:

make install

4.2. Installation of EMBOSS

EMBOSS is available for download from the primary site at the UK EMBnet node by anonymous ftp3. This directory contains the EMBOSS package and several associated packages (collectively known as EMBASSY) that are distributed with EMBOSS. These are downloaded to a suitable location.

Documentation is available on the WWW at the EMBOSS web site4.

FreeBSD distributions from 4.2 onwards now include EMBOSS as an optional package maintained by Johann Visagie.5

The EMBOSS and EMBASSY packages are downloaded to a suitable directory.

EMBOSS-1.0.0.tar.gz

First the EMBOSS distribution is unpacked by the command

gunzip EMBOSS-1.0.0.tar.gz

tar xf EMBOSS-1.0.0.tar

This will create a new directory, EMBOSS-1.0.0 or similar.
Following command help to enter within the EMBOSS directory

cd EMBOSS-1.0.0

Compilation

Building EMBOSS is easy. It follows the usual GNU style of

./configure,

make,

make install.

./Configure

To accept the default configuration, just ./configure typed and EMBOSS will get on with it.

However to make some changes to the configuration parameters according to local policy, the configuration script will attempt to find the neccessary components in the system to

determine how to successfully build EMBOSS. It typically expects the GNU C compiler (gcc) and several standard libraries that should already be part of the Unix/Linux system. EMBOSS should configure, compile and run on most modern Linux distributions straight out of the box.

Installation Directory

The directory in which eventually we wish to install EMBOSS need to have permission. It may be put into somewhere else other than the standard location of /usr/local/emboss.

The installation directory is controlled by the --prefix argument. Suppose in a case all applications owned by a non-priviledged user and installed in a package specific directory under

/site/prog

./configure --prefix=/site/prog/emboss

will install EMBOSS under /site/prog/emboss.

The binaries will be installed in /site/prog/emboss/bin with shared libraries installed in /site/prog/emboss/lib.

System wide data are installed in /site/prog/emboss/share/EMBOSS/data, and the configuration files (ACD files) for the applications will be installed in /site/prog/emboss/share/EMBOSS/acd (or for EMBASSY in directories corresponding to the package name.)

Documentation is installed in /site/prog/emboss/share/EMBOSS/doc.

The installation directory should be specified using a full path otherwise interesting failures may occur.

The individual directories for installation can be modified with other configuration commands but this is usually not necessary.

./configure --help give more information on the directories that can be changed and other configuration options can be used. This may take a short time as various messages scroll up the screen.

Reconfiguration

It is not uncommon to make typos or other mistakes when running ./configure.To run configure again the file config.cache must be deleted and make clean should also be run before running ./configure with the correct options.

Configuring EMBOSS graphics

Depending on your system one may need to explicitly configure the graphics.

The PLPLOT library can produce output to many devices but requires certain libraries that are NOT distributed with EMBOSS. To get X-windows based output one must have X installed else PLplot will not build the required driver. The location of X-windows library may need to specify with the configuration options:

--x-includes=DIR (X include files are in DIR)

To explicitly configure PLPLOT without X-windows, --without-x is used.

To get PLPLOT to produce PNG images we will need to have the z2, png3 and gd4 librarys installed. gd version >= 1.6.3 must be used as the older versions support GIF which is NOT supported in later versions. If for some reason system do not have the required libraries and system support group will not update them for the system then all three latest versions (z,gd,png) must be installed to a new directory and then this new directory is added to configure line for EMBOSS

./configure --with-pngdriver=my_dir where the z, png and gd libraries were each installed using ./configure --prefix=my_dir

It may also be helpful to ensure that the LD LIBRARY PATH environment variable is set appropriately to include the libraries in the path.

EMBOSS can be told explicitly to not to include PNG support with --without-pngdriver. If ./configure has found a suitable PNG library by watching for something like the following then we can mention above command.

When running ./configure:

checking if png driver is wanted... yes

checking for inflateEnd in -lz... (cached) yes

checking for png_destroy_read_struct in -lpng... (cached) yes

checking for gdImageCreateFromPng in -lgd... (cached) yes

This means that the configuration script has located the PNG libraries on system. If there is message indicating that ./configure could not find the libraries or that the version of gd was too old then one should install the latest versions of the libraries yourself and rerun configure with the correct --with-pngdriver value.

Building EMBOSS

Building EMBOSS is a matter of typing 'make' and going to find something else to do for the next ten minutes to half an hour depending on the speed of your system. EMBOSS will first build the shared libraries (PL PLOT, AJAX, and NUCLEUS) and then build the applications.

Plenty of warnings (especially on SGI systems) might be seen complaining about libraries not being used to resolve any symbols. These can be safely ignored.

Assuming that compilation was successful, now 'make install' can be run. After a few minutes and many page full of messages, EMBOSS should be installed where specified in the --prefix option (or in the default location of /usr/local/emboss if --prefix was not specified).

Testing EMBOSS installation

EMBOSS installation is tested by trying the program 'wossname'

wossname -auto |more

This should give a long list of programs that are available. Press space to page down through the list. This is just the EMBOSS programs and doesn't include any of the EMBASSY programs.

OTHER WAYS TO INSTALL EMBOSS

Installing EMBOSS in package format

EMBOSS can be installed on almost all Unix/Linux operating systems using the instructions above, but the package format can be far more convenient. A package is a precompiled set of binaries with installation instructions that can be set up on your system with a minimum of work. In some cases the package will check for the correct libraries and install those as neccessary.

Installing EMBOSS on FreeBSD

A FreeBSD EMBOSS package has been created by Johann Visagie6 of Electric Genetics. This will be distributed on the installation CD's and through the normal distribution channels from FreeBSD version 4.2 onwards. For the FreeBSD user with an up-to-date ports tree7, installing EMBOSS reduces to two simple commands (as root):

cd /usr/ports/biology/emboss

make install

The EMBOSS documentation will be installed in /usr/local/share/doc/EMBOSS instead of the default location.

4.3. Installation of GLIMMER

To install Glimmer3, the compressed tarfile glimmer302.tar.gz is downloaded from the website6. Then uncompress the file by typing

tar xzf glimmer302.tar.gz

A directory named glimmer3.02 should result. In that directory, is a subdirectory named src . Chane to src directory

cd glimmer3.02

cd src

Within the src subdirectory type

make (or alternately gmake).

This will compile the Glimmer3 programs and put the executable files in the directory glimmer3.02/bin. These files can be copied or moved to whatever directory is convenient to the user.

4.4. Installation of RASMOL

RasMol may be obtained by either by anonymous FTP7 or downloaded from the web8. Then uncompress the file by typing

tar xzf Rasmol.tar.gz

A directory named Rasmol should result. In that directory, is a subdirectory named src . Chane to src directory

cd Rasmol

cd src

Within the src subdirectory type "xmkmf" to generate a "Makefile" for your particular system from the distributed Imakefile Alternatively (or if the first method fails), copy the file Makefile.in to Makefile, using the command

cp Makefile.in Makefile

Then modify the contents of the Makefile to determine your local C compiler. The default set up is for an 8bit UNIX workstation with the X11 shared memory extension, compiled using the GNU C Compiler. And also modify the #defines in the file rasmol.h (see below) Note: IBMPC should not be defined.

The file rasmol.h contains a number of #define directives that control the runtime behavior of the program. The following directives may be defined or undefined to suite the local site.

THIRTYTWOBIT

SIXTEENBIT

EIGHTBIT

This determines whether RasMol will display and produce 8bit, 16bit or 32(24) bit output. By default the symbol EIGHTBIT is defined producing images with up to 256 colours.

This symbol must be defined if IBMPC is defined.

A typical UNIX build:

```
        /* #define IBMPC        */
/* #define MSWIN        */
        /* #define APPLEMAC     */
        #define X11WIN
        #define UNIX

        /* #define DIALBOX      */
        #define SOCKETS
        #define TERMIOS
        #define PROFILE
        #define MITSHM
```

Compilation

The software is compiled using the UNIX make utility. (i.e. type "make")
Place the 'rasmol' executable on the execution PATH, i.e. /usr/local/bin
Install rasmol.hlp as /usr/local/lib/rasmol/rasmol.hlp

5. Conclusion

InBit is a powerful open-ended integrated bioinformatics toolkit. Standalone facility, well organized and user-friendly interface make it a useful toolkit in the bioinformatics domain.

References

[1] Biegert, A., Mayer, C., Remmert, M., Soding, J. and Lupas, A. N. The MPI Bioinformatics Toolkit for protein sequence analysis. Nucleic Acids Research, 2006, Vol. 34, Web Server issue

[2] Rampp, M., Soddemann, T. and Lederer, H., The MIGenAS integrated bioinformatics toolkit for web-based sequence analysis. Nucleic Acids Research, 2006, Vol. 34, Web Server issue

[3] Gracy, J. and Chiche, L., PAT: a protein analysis toolkit for integrated biocomputing on the web, Nucleic Acids Res. 33:65-71, 2005.

[4] Salzberg. S., Delcher, A. Kasif, S. and White O. Microbial gene identification using interpolated Markov models, Nucleic Acids Research 26:2 (1998), 544-548

[5] Rice,P. Longden,I. and Bleasby,A. EMBOSS: The European Molecular Biology Open Software Suite (2000) , Trends in Genetics 16, (6) pp276--277