embnet.news

Volume 6 Nr1 (ISSN1023-4144)

April 30, 1999

Editorial

In the 90´s it was pop to be in biotech. However when you look at how well these new start-up companies have performed on the stock market, their record is less than impressive. Perhaps for biotech to succeed there needs to be more involvement with bioinformatics. It has often been joked that if you could even just spell bioinformatics correctly it would ensure you a job with any of the major pharmaceutical companies, and if by pure chance you could actually do something useful in bioinformatics then they would throw in a company car as part of the bargain. For the millennium it will be the bioinformatics companies who will be analysing the data in silico and passing their results on to the biotech companies in the hope that they will both reap rich rewards.

Recently Swiss-Prot, because of a recent funding crisis decided to raise revenue by asking commercial users to pay a licence fee. They have their own commercial web site, GENEva BIOinformatics or GENEBIO (www.genebio.com) for short, which is dedicated to providing quality databases, software tools and services to the Life Science industry, particularly in the field of proteomics.

SRS (srs.ebi.ac.uk), which started off as a project at EMBL Heidelberg, has with the blessing of the EMBL Director General become part of Lion (www.lion-ag.de). Lion aims to become a leading genomics information company and they will offer their customers innovative solutions for

- integrated genomics
- functional target characterisation
- biological knowledge into life science information

Contents

Part of the R&D team that worked at the EBI have formed Synomics (www.synomics.co.uk) in Cambridge UK. They hope to foster and develop interactions with major pharmaceutical companies to help them to gain insights into the mechanism of disease and develop many of the new technologies, including genomics, proteomics and combinatorial chemistry. It is a question of " Have data will analyse" Synomics already collaborates with Incyte on a new software initiative (www.synomics.co.uk/news/news0003.htm) in genomics.

Recently in the Sunday Times colour supplement there was a long article about Decode (www.decode.is) and its CEO, Dr. Kari Stefanson, from Iceland, who is working in collaboration with Hoffmann-La Roche (www.apnet.com/inscight/02091998/graphb.htm). The article expressed some concerns about the use of genetic data for either genetic screening before birth, or for sale to insurance companies, to help them evaluate the risks involved in insuring people who may have some hereditary disorder.

Even today, as this editoria is being written, a new collaborative effort between 10 large pharmaceutical companies, the Wellcome Trust and five academic institutes, has been publicly announced. The aim is to identify up to 300,000 single nucleotide polymorphisms or SNPs (news.bbc.co.uk/hi/english/sci/tech/newsid_319000/319872.stm), map at least 150,000 and put all the information into the public domain. The importance of the project to genetic research is equal to both the human genome sequencing and public EST projects.

Since we are sure many of your readers have invested in RealPlayer technology you can listen to Dr Michael Morgan from the Wellcome Trusts views on SNP (No that is not the Scottish Nationalist Party) from the BBC WWW site.

It is Spring time and a thousand flowers are in bloom. Chairman Mao would indeed be happy if he were to read about the chronicle of the future (http://www.chronicle-future.co.uk). and he would no doubt be surprised that even pigs might fly.

EMBnet Editorial Team

INTERviewNET

Michael Wise of the EBI interviewed by Robert Harper

Q1. This is the second time that you´ve been here to the EBI from Australia. What is the attraction?

EBI and Cambridge generally are a real bioinformatics powerhouse in a lovely part of the world (and the weather is not nearly as bad as people say). On the other hand, times are very tough for academics under the current Australian government.

Q2 What is happening down under regarding Bioinformatics?

It's gradually taking off. I was able to establish at Sydney University an undergraduate course in Bioinformatics - B.Sc. (Bioinformatics) - on the basis that a discipline that crosses conventional academic demarcation lines is best taught from day 1, rather than as an afterthought. The course is in its second year this year, and has attracted some very able students.

It is also spawning imitators in other Australian Universities. Sydney University also created a bioinformatics consultation company - Encompass - affiliated to AGIC (the body the currently administers ANGIS). At the same time, the bioinformatics arm of WEHI (Walter and Eliza Hall Institute in Melbourne) and AGREF in Brisbane are going from strength to strength.

Q3 Do you use any of the Angis services while you are in Australia?

I mainly used ANGIS, who developed a lovely web interface based on Entrez, but certain things were better done via SRS, which is available at WEHI.
Neither service is really able to deal with batch queries, so occasionally I used my account at ANGIS to access the data directly. Of course, now that I'm here, I use the services provided by EBI.

Q4 Do you miss the BBQ and kangaroo steak or are you more interested in Python?

I didn't know that Python is widely available in the wilds of Cambridgeshire ;-) Actually, Python is named after the OTHER Monty, so it's entirely appropriate that I use it here. (Editors note: Michael writes programs in Python)

Q5 In a genomics wrestling match who would you bet on, Craig Venter or John Sulston?

John, of course. Looking at the recently announced SNP consortium, in which my sponsor - Bristol Myers Squibb - are are a major player, it's gratifying to see that the big institutional and corporate research initiators are beginning to realise that it's best for everyone when the results of science are put in the public domain.

More to the point, I believe it will only be through the large, publicly funded joint efforts that the currently less interesting parts of genomes ("junk" regions) will be properly completed.

Q6 Is there any truth in the rumour that Australia has been genetically engineering Cricket players for the upcoming Test series?

It's not required! We will win regardless

Good onya mate Michael Wise

Quarterly quotes

Tell me who your hero is and I will tell you what you are like

We learn by listening to what the pioneers in the field have to say, and recently there has been a spate of articles appearing in the British newspapers concerning bioinformatics, genomic research, and genetic engineering.

Here are some of the quotes that have been made, and also links to the featured articles in the Guardian. (Editor´s note: Has the editorial board gone left-wing then?) Need we say that there exists a certain tension in the atmosphere between academia and the private sector.

The Maverick: *Kary Mullis*

We have been breeding selectively for all time. You would like everyone to be funnier and happier, wouldn't you?

In Star Trek they have a doctor who is a hologram. That makes more sense to me.

I am an optimist about having control of future generations. We've been doing it for 2m years.

The Joker: *Craig Venter*

My processing centre will have more computing power than 90% of countries We're going to spend $300 m sequencing

the human genome and will make it available on the Internet.

The Prophet: *Lee Silver*

At the beginning there doesn't seem to be anything wrong in making small changes. But you slowly move to the point where its disastrous.

No one has a problem with a vaccine for asthma. But they do when you talk about giving people a genetic vaccine before birth.

Genetics is going to be more important than physics was. It is going to affect everyone's life.

The Map Maker: *John Sulston*

This will tell us what makes our brains work, and our minds. It will tell us what we are.

You look at all the squiggles and you know there are amazing discoveries to be made. But you haven't a clue what they all mean.

We survived knowing that the earth is not flat; we survived the theory of evolution. Can we survive the understanding of consciousness, I wonder?

Book review

An Introduction to Bioinformatics
By TK Attwood and DJ Parry-Smith
ISBN 0 582 327881. Price 17.95 GBP

Reviewed by Andrew Lloyd, INCBI, the Irish EMBnet Node.

Any biology text that cites Francois Jacob's concept of bricolage in the first chapter has got off to a sound start. An Introduction to Bioinformatics does that and carries on to cover much of what you would expect from a book of that title. The Holy Grail of bioinformatics is identified as the prediction of 3-D protein structure from linear protein sequence. Accordingly, this is very much a protein analysis book and the various chapters come to a climax (Chapter 9) entitled ‘Building a sequence search protocol', which is a "batterie de cuisine" for determining the structure and function of an unknown sequence. The impact of this chapter is greatly enhanced by being integrated with a WWW-based tutorial. An overview roughs-out the structure of the book and each chapter finishes with a summary and a list of further reading.

Despite receiving electronic copy from their authors the publishing world takes ages to bring the printed word to a grateful public, so a commitment to maintain Web based material is essential in a fast moving field such as bioinformatics. Not only this, but the web can be treated as a free colour (and moving, singing and dancing) supplement. Traditional hardcopy publishers must be feeling the ground lurching under their feet. One could have asked for a more specific flag to this essential resource than http://www.awl-he.com/biology/ but perhaps this generic URL was chosen in the best interests of continuity.

There is a reasonably comprehensive review of current bioinformatics related Internet resources but there does not seem to be a WWW supplement to this potentially labile area. I was interested to see such attention being paid to the structure and functionality of the primary and secondary databases. These are often taken for granted and any worthwhile introductory textbook should cover this ground explicitly.

Chapter 6 covers local and global alignments, Dayhoff and Blosum substitution matrices, identity and similarity and homology searching protocols at an appropriate level of detail and addresses some of the most important issues involved. With well designed WWW servers readily available anyone can do a homology search, but estimates are frightening for the proportion of wrong-headed, defective or inadequate homology searches carried out by users. Attwood and Parry-Smith's book is valuable because it gives a fair grounding in how to do an effective homology search and interpret the results. Their emphasis on the value of secondary databases and their provision of one view of how to effectively access this rich seam of information makes it a wonderful asset for those of us who try to teach bioinformatics. If students come away confused about the difference between regular regular expressions and fuzzy regular expressions they will nevertheless have been given enough information to think beyond such relatively simplistic criteria for classifying protein families.

The chapters on DNA sequence analysis are written with less assurance. The issue of ORF detection and intron-boundary prediction - surely at least a Holy Teacup of Bioinformatics in the post-genomic era - is skimmed as a rather trivial codon usage issue. There is rather better coverage of the EST deluge and how it is being tapped, including a nice exploration of how the pharmaceutical world might be trying to exploit dbEST. This is inspiring stuff to have in an undergraduate text; clearly showing that bioinformatics is not only a pretty academic face but also a powerful workhorse for the commercial world.

Crucially there is no chapter on phylogenetics or tree drawing. This is surely to be included in anyone's definition of bioinformatics. I rather admire the authors' decision not

to attempt to cover a topic that they may have felt was outside their expertise. There are a number of books on molecular evolution, and good chapters on phylogenetic theory and practice, but it would be nice to be able to recommend a single book that covers at least the widely recognized core of bioinformatics.

As one author is the manager of one of the more mobile EMBnet nodes, EMBnet and EMBnet's achievements get good coverage. I was particularly flattered to see INCBI being given parity of esteem with NCBI in Table 2.2. The map (Figure 2.3) which locates the Irish EMBnet node in Belfast somewhat dampened my euphoria. I will not enumerate the other errors but have to report that everyone involved has failed Political Geography 101. As the recommended text for Bioinformatics 101 (or equivalent), however, this book will fit the bill nicely.

Using GCG with command scripts

Kimmo Mattila, CSC-Center for Scientific Computing, Finland

For most bioscientists the UNIX interface may be considered an inconvenient environment for using GCG and other programs. Even though SeqLab and SeqWeb nowadays provide an efficient and intuitive user interface to GCG, it is still valuable to know how to use GCG with simple line commands. Working at a quite basic level in the system allows the user to create and develop their own solutions. One way of utilizing the flexibility of UNIX is command script programming. A command script is simply a file, which contains a set of normal UNIX commands, that the command shell will perform automatically in the given order.

1. Constructing a script file

A script file is a simple text file that can be constructed with normal text editors like pico, emacs or vi on your UNIX machine. To create a new script file, for example type:

```
> pico gcg.script
```

A script file usually starts with a command line which defines the UNIX shell to be used. sh-shell is the most used shell for doing scripts but in the case of GCG it is easier to use the csh-shell, so the first line of the script file is:

```
#!/bin/csh
```

After that you add the UNIX commands, you wish to perform. In practice, just type into the file the commands that you would normally use to do the task in an active command shell. If a script line starts with a # mark it will be skipped and the rest of the line considered as a comment.

For example, the following script can be used to create a subdirectory "mapfiles" and copy all .map files to there

```
#!/bin/csh
mkdir mapfiles
cp *.map mapfiles/
```

After saving the script file and closing the editor you can perform the commands in the script file by giving the command:

```
source gcg.script
```

2. Adding GCG commands to the script file

With GCG there are two cases where you should consider using command scripts: when you are routinely doing some tasks several times or when you are doing a GCG job that takes several hours.

Before you can include GCG commands in your script file, you have to add the GCG setup commands to your script. These commands are site dependent. Normally they include sourcing the gcgstartup file in the GCG home directory and running the gcg command. In the examples below, the GCG home directory is assumed to be: /usr/local/gcg. The GCG commands, used in the scripts, are typed as boldface letters.

Normally, when you are running long GCG jobs, it is reasonable give all the parameters, that the GCG command requires on the same command line, instead of approving the values in an interactive way. In many cases you will use the default parameters and all you have to do is to type the GCG command to the script file in the form: GCG-command inputfile-def, like:

```
#!/bin/csh
source /usr/local/gcg/gcgstartup
gcg
map scact.em_fun -def
```

However, if you wish to use other than default values, check the available parameters from the corresponding Command Line Summary of the gcg manual and add the parameters you want to use.

Remember to include the -def option to take care of the rest of the parameters. In the command script below we do map analysis with default parameters, but only for the 500 first bases of the sequence. The output of the analysis is now stored in a file: scact2.map.

```
#!/bin/csh
source /usr/local/gcg/gcgstartup
gcg
map scact.em_fun -END=500 -OUT=scact2.map -
def
```

You can, of course, do several tasks within the same script. You must just remember to proceed in the same order as you normally do after starting GCG. The following script fetches a DNA sequence, does a map analysis on it, translates the sequence into an amino acid sequence and finally runs a netblast search for the sequence.

```
#!/bin/csh
source /usr/local/gcg/gcgstartup
gcg
fetch scact -def
map scact.em_fun -def
translate scact.em_fun -def
netblast -IN1=scact.pep -IN2=swissprot -def
```

3. Batch jobs

On many main frame computers, you can interactively run tasks, that take only limited time, say some hours. Longer processes can be run only through a batch job system. Normally this is not a problem for a GCG user, but in some cases you may have to do even longer jobs than the time limit will allow.

By submitting tens of fasta searches to a batch queue at a time, you do not have to sit by the computer submiting a search only after another has finished. Instead you can just let the computer to do the work and go home to study the UNIX manual or to do something that you enjoy. Below are some guidelines for submitting GCG batch jobs to the NQS queue system. If you are not sure if the computer you are using has an NQS system, then ask your system manager.

Submitting a GCG batch job means sending a GCG command script file to the queue managing system. First, however, you have to add few lines to the script file so that the queuing system knows, the resources you need. These additional lines always start with: #QSUB, followed by the variable.

The things you have to tell to the queue system are usually queue type (-q), required CPU time (-lT) and the memory request (-lM). All these things are site dependent. If you have the NQS system available, you can check the job types and time and memory limits with the command qstat.

Below is a sample script for submitting a GCG batch job.

```
#!/bin/csh
#QSUB -q prime # Queue request to queue:
prime.
#QSUB -lT 9000 # CPU time request 9000s =
2.5h
#QSUB -lM 50mb # Reserve 50 Mb of memory
#QSUB # No more embedded options
source /usr/local/gcg/gcgstartup
gcg
fetch scact -def
map scact.em_fun -def
translate scact.em_fun -def
netblast -IN1=scact.pep -IN2=swissprot -def
```

If you now have the above lines in a file called "gcg.script", you can submit your job with the command:

```
qsub gcg.script
```

After submission, you can follow the status of your batch job with the command:

```
qstat -a
```

4. (A little bit more) advanced topics

The examples above have been quite simple cases. However, if you know the basics of UNIX, you can write small program-like script files, which can include loops and selections. Just think what you want to do and try to figure out, which command would do the task. You can find hints of possible commands from UNIX guides and by studying the manual pages of UNIX commands, like:

```
man rm
```

Try the commands you are going to use interactively before constructing the script file.
Below are some hints to help you get started.
Variables can be set with the command:

```
set variable=(value)
```
and they are recalled with the $ sign:

```
$variable
```

For example, the command

```
echo $variable
writes the value of `variable´ to the output.
```

If you use `` marks like

```
command1 `command2`
```
they make command1 use the product of command2 as an argument.

A pipe, | , guides an output of one command into an input of the second one.

> guides an output of one command into a new file.
>> appends the output of one command into an old file.
< an existing file´s contents are used as input for the preceding command

A loop can be made with the commands:

```
while (condition)
commands
end
```

Below are two script files which give examples of what you can do with just few lines. These sample scripts are not well written and optimized programs, but instead they try to demonstrate, that you don't have to know everything about UNIX in order to do something useful with it.

UNIX commands used in the scripts. (For more detailed description give the command: man command on your UNIX machine.)

echo
writes its arguments to the output

```
expr
```
evaluate arguments as an expression

```
grep
```
search a file for a pattern

```
head
```
reads lines from the beginning of a file

```
rm
```
removes a file

```
set
```
sets a value for a variable

```
sort
```
sorts lines in a file and writes the result to the output.

```
source
```
runs a script file

```
tail
```
reads lines at the end of a file

```
wc
```
word, line and byte or character count

Example 1.

The first example is a batch job script. It reads sequence names from a separate file, called seqnames.txt, that is just a list of sequence names like:

```
scact
sca1b
mg16sr
huadcyc
```

After being submitted with qsub command, the script will go through the list and fetch, translate and run netblast for each of the sequences:

```
#!/bin/csh
#QSUB -q prime # Queue request to queue:
prime.
#QSUB -lT 9000 # CPU time request 90000
#QSUB -lM 50mb # Reserve 50 Mb of memory
#QSUB # No more embedded options
echo "GCG Script"
source /usr/local/gcg/gcgstartup
gcg
set r=(`wc -w < seqnames.txt`)
set n=(1)
while ($n <= $r )
set name=(`head -$n seqnames.txt | tail -1`)
fetch $name -def
set fname=(`ls | grep $name`)
translate $fname -def
netblast -IN1=$name.pep -IN2=swissprot -def
set n=(`expr $n + 1`)
end
```

Example 2.
This second script uses two program loops to study the effects of the gap creation penalty and gap extension penalty in using `gap´ for sequences mg16sr.em_ba and scact.em_fun. The gap creation penalty is varied from 5 to 50 with step size of 5 and the gap extension penalty from 1 to 5 with a step size of 1. The resulting similarity and identity values are sorted according to the similarity and stored in a file "sorted.out". Note that the expr command can only handle integers so, if you would like to use step size less than one, you should use different commands.

```
#!/bin/csh
source /usr/local/gcg/gcgstartup
gcg
echo "Testing Gap" > result.out
set gap=(5)
set n=(1)
while ($gap <= 50 )
set len=(1)
 while ($len <= 5)
gap mg16sr.em_ba scact.em_fun -GAP=$gap -
LEN=$len -OUT=out.pair -def
echo `grep Similarity out.pair` Len= $len
Gap= $gap >> result.out
rm -f out.pair
set len=(`expr $len + 1`)
set n=(`expr $n + 1`)
end
set gap=(`expr $gap + 5`)
end
sort result.out > sorted.out
```

GCG Graphics to Macintosh

Tapani Hyvönen A.I.Virtanen Institute University of Kuopio P.O.Box 1627 FIN-70211 Kuopio, Finland

Introduction

GCG is a basic tool for all molecular biology researchers. Usually, the programs are run from the command line and the input and output files are text files, but many programs also offer graphical output. However, the pictures are not always ready for publications and you must edit them.

In principle, you can produce output in two ways, output to screen (Tektronix, X-Windows) is suitable to get an impression of the graphic result, output to file (a figure or a postscript file) is recommended if you are going to edit the results later on a Macintosh.

You can:

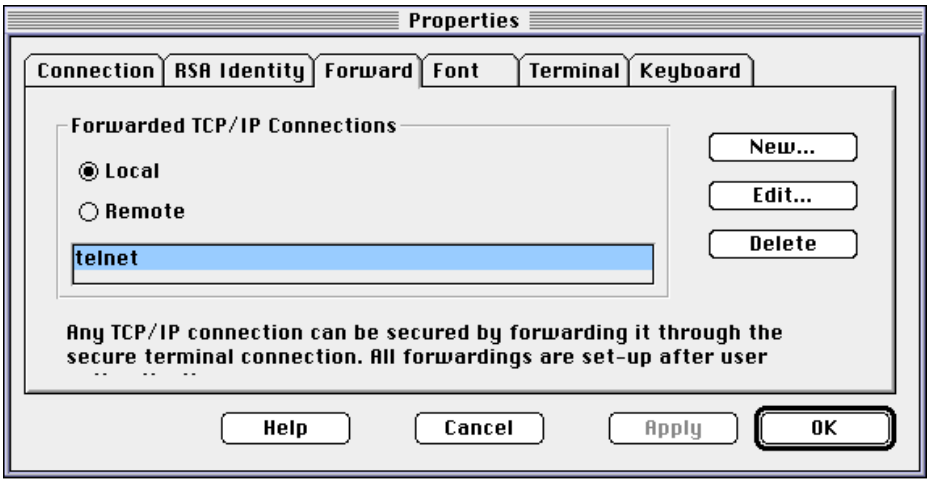
- use Tektronix window to view the graphical output but this is usually unsuitable for publication.
- run programs in X-windows (e.g. SeqLab) or view postscript files by xv. With xv you can also transfer postscript-files to other formats, e.g. JPEG which can be viewed by web browsers.
- save output as a figure file by using -fig(ure) parameter on command line. This works in all GCG programs with graphic output. GCGFigure, a Macintosh program, can read these files and transform them to PICT files which all image programs can read.
- save output as a postscript file. Postscript files can be viewed in X-windows by xv , be printed to a local printer or can be imported to image programs. Though postscript is well documented, programs' ability to read postscript files varies and messages like postscript error or could not parse postscript code are frequent.

SSH, NCSA Telnet and X-windows

For security reasons it is important that passwords are always sent encrypted. The most popular terminal program, NCSA Telnet, does not support encryption. SSH (Secure Shell, Datafellows) can create secure terminal connections and, moreover, can port any TCP/IP program. In other words you can use any insecure TCP/IP-program and secure the communication by SSH.

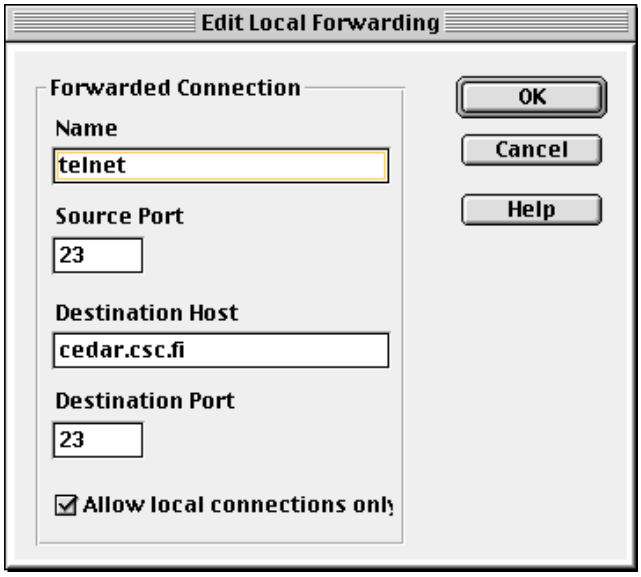
To port X-windows through SSH, open

Setting from - File-menu, click - Properties...-button and select - Forward X11 from -Connection-tab. To port telnet through SSH, choose - Forward-tab and click New-button.



Port telnet through SSH as in figure "Edit Local Forwarding". Telnet port is usually 23 - you can check it by the Unix command

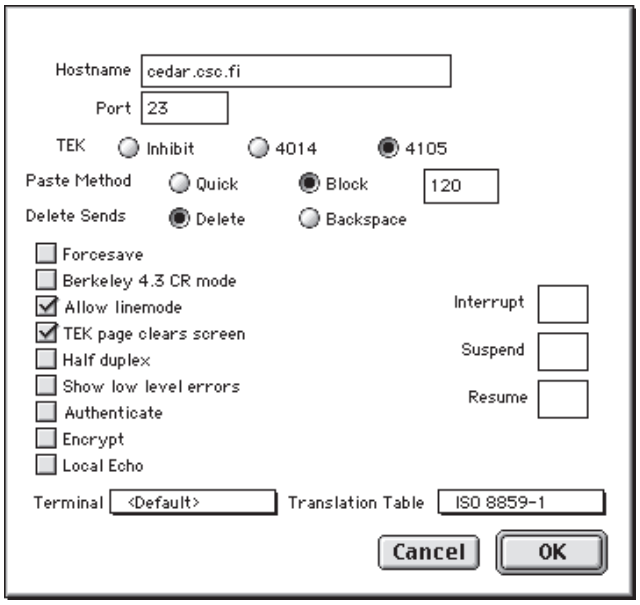
more /etc/services. Choose "Allow local connections only". Save your settings, and by double clicking this file, you can connect to Unix-host securely by SSH.



Similarly, you can can port any insecure TCP/IP connection through SSH (e.g. ftp to port 21, pop3 mail to port 110). Port 21 is used to send only ftp usernames and passwords, data is transferred unciphered to other port. Although data transfer is not secure it is important that usernames and passwords are encrypted. Fetch is maybe the most common Macintosh ftp program. To cipher Fetch, first connect to a Unix-host using SSH. After that, open Fetch and use your Macintosh IP number to make the ftp connection to the Unix-host. Use your Unix-host's username and password to login.

Tektronix graphics

SSH is usually enough to run GCG programs from command line. NCSA Telnet is needed only if you need Tektronix graphics. First, open a SSH connection to a Unix-host, then open NCSA Telnet and open a connection to your Macintosh IP number using the Unix-host's username and password. Addresses like



127.0.0.1 or localhost do not work. From Edit->Preferences->Sessions -menu, choose
Tek4105 radio button.
From File-menu, save your Telnet Set. Onwards, take secure telnet connections by
double-clicking SSH- and TelnetSet-icons.

Start GCG as usual by the command

```
use gcg.
```

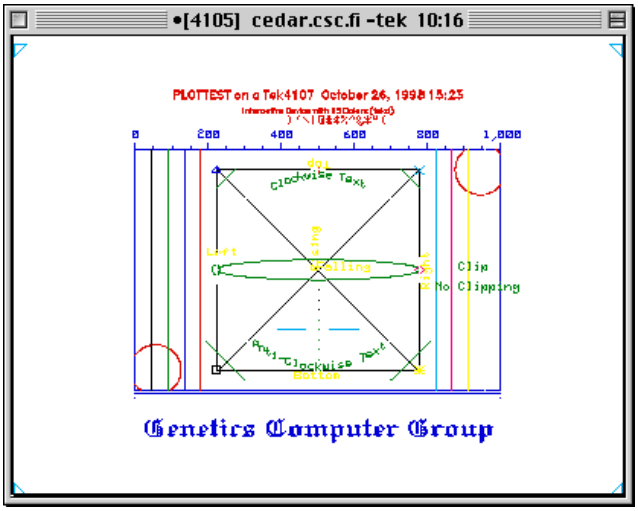
GCG command tektronix defines how GCG programs handle Tek graphics:

```
cedar ~> use gcg
cedar ~> tektronix

Use Tektronix graphics with what
device:
TEK4107
GRAPHON-TEK4014
VT340-TEK4014   SMARTERM-TEK4014
LN03-PLUS
VERSATERM-TEK4105
TEK4014
```

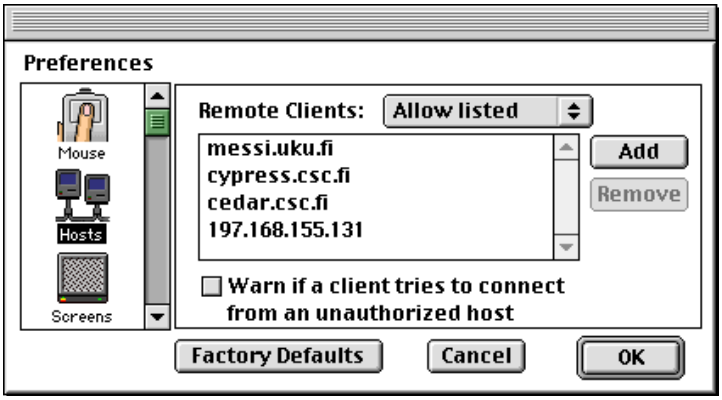
Please choose one (* TEK4107 *) return
To what port is your TEK4107 connected (* /dev/tty15 *) term

Plotting Configuration set to:
Language: tekd
Device: TEK4107
Port or Queue: termm

Use GCG's plottest command to check your configuration, you should get the following Tek-screen:



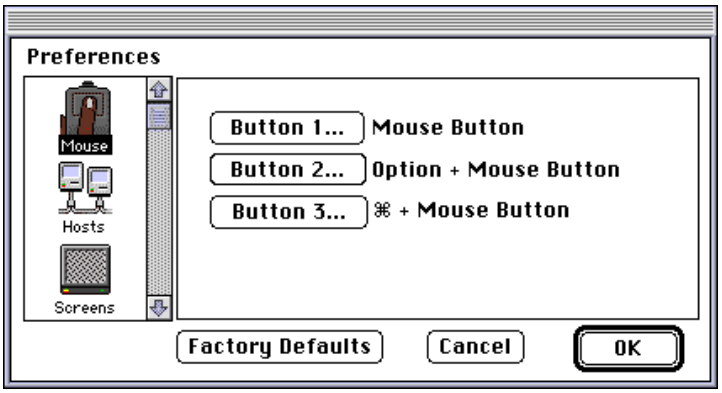
Tek graphics is suitable to view results on a screen. It is possible to copy pictures to a clipboard and paste into graphics programs but it is better to use figure format to



transfer graphics from GCG to Macintosh.

X-windows

If you want to try X-windows but do not want to invest in a commercial software at once, you can download free X-server software (MI/X_PPC, free X-Server software for the



Macintosh, ftp://biow.biocomp.unibas.ch/pub/biocomp/MI_X_PPC.sea.Bin) from Biozentrum of University of Basel.

Preferences-menu defines the preferences (clear, uh?) e.g. which Unix-hosts can connect to X-server. In some other programs, this must be defined by Unix commands xhost or xauth. Again, as previously with forwarding ftp or telnet, the IP number (197.168.155.131 in this example) of your Macintosh is the right address. SSH can form X-windowing environment automatically but if you give Unix command (or put it into .login file) setenv DISPLAY ip_number_of_mac:0.0 the data traffic is not encrypted. In this way, SeqLab or Web browsers run much faster in X-windows if sequence data is not confidential.

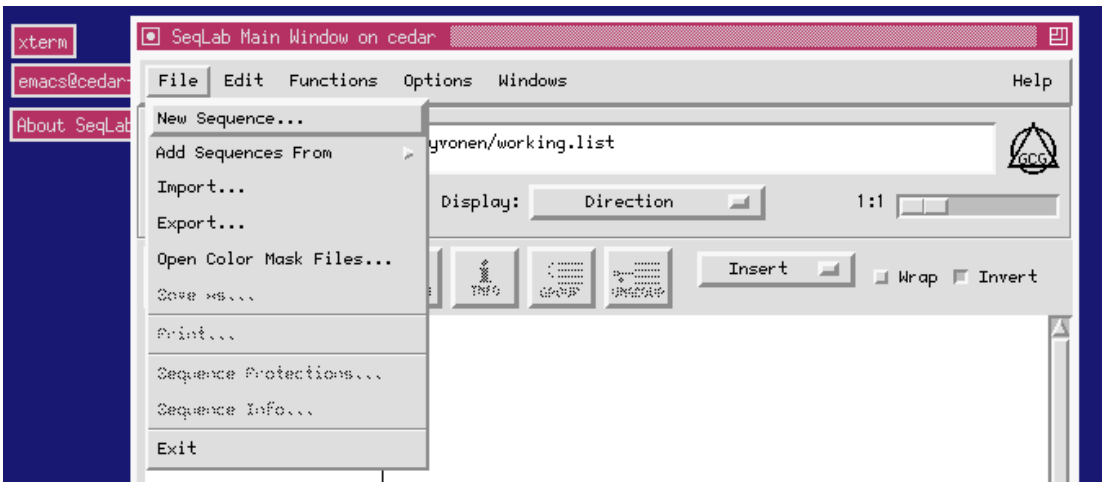
From the Preferences-menu, you can also define how the three buttons of the X-terminal mouse can be emulated by the single button of a Macintosh mouse.
Start the X-server (TNTx program) on the Macintosh before running X-windows programs in Unix. If you get error messages like this

```
Cedar ~> xterm &
[1] 24107
    cedar ~> Warning: Cannot convert string
"-*-*-bold-r-*--*-140-*-*-m-*" to type
FontStruct
```

just press return key. Unix-host's xterm wants to use font which is not available in Macintosh' X-server. The fonts used by the Unix-host can be listed by command

xlsfonts | more, and from this list (long !) you can choose the font you want. To use font 8x13, use command
xterm -fn 8x13 & and xterm starts without error messages. To run SeqLab, start GCG as usually by command
use gcg and thereafter start SeqLab.

```
cedar ~> use gcg
...
cedar ~> seqlab &
[3] 24107
cypress ~> Warning: Cannot convert string "-
*-*-bold-r-
*--*-140-*-
*-m-*"  to
type
FontStruct
```



Use the Help-button to get advice how to use SeqLab. Sometimes the TNTx-program quits itself (aka crashes) when last window is closed or even

when you start SeqLab. On my Macintosh, TNTx/SeqLab is always stable when SeqLab is launched as the third program. Therefore, I usually run two other X-windows programs (xterm or emacs) before running SeqLab. If you cannot tolerate this you will have to switch to a "real" X-server.

GCGFigure

GCGFigure is the best way (IMHO) to transfer graphics from GCG to Macintosh image programs. First, run GCG program with parameter -fig(ure)=anyfile.name. The output file (figure file) is a text file and can be transferred to the Macintosh by Fetch or via clipboard. If the scrollback buffer of Telnet (Edit-menu -> Preferences -> Terminals) or SSH (Edit-menu -> Connection properties -> Terminal-tab) is enough big, copy the file from screen to clipboard, paste into text editor and save as a text file. Open the figure file into GCGFigure and save as a PICT file which all image programs can open. You can download GCGFigure from ftp://alanine.gcg.com from directory /pub/mac/.

PostScript-files

GCG programs can save output as a postscript file. Editing postscript files is difficult and not all image programs cannot read postscript files. You can use the plottest output file to test if the image program you use can handle postscript. Apple Printer Utility can print postscript files and programs for other printers are available from printer manufacturers.

When you have set up GCG, define the postscript output file name and printer used with the command

```
 postscript.

 cedar ~> use gcg
 ...
 cedar ~> postscript

Use Postscript graphics with what device:

 LaserWriter
 Lzr1200
 LN03-ScriptPrinter
  LPS20
 ColorScript-100
 EPSF (single page encapsulated postscript
format)
 CEPSF (color EPSF)
  Please choose one ( * LASERWRITER * ) return
  To what port is your LASERWRITER connected
(* /dev/tty15 *) ps-filename
  Plotting Configuration set to:

  Language: psd
  Device: LASERWRITER
  Port or Queue: ps-filename
```

If you have X-windows connection to Unix-host, you can view postscript files by xv and also transfer to other formats. Also, when you use xv, it is better and faster to define DISPLAY to view files unciphered.

```
setenv DISPLAY ip_number_of_mac:0.0
xv ps-filename
```

ForCon : a software tool for the conversion of sequence alignments

Jeroen Raes and Yves Van de Peer* - Department of Biochemistry, University of Antwerp (UIA), Universiteitsplein 1, B-2610 Antwerpen, Belgium
****To whom correspondence should be addressed (E-mail: yvdp@uia.ua.ac.be)***

key words : sequence alignments – phylogenetic analysis – file conversion

Abstract

ForCon is a software tool for the conversion of nucleic acid and amino acid sequence alignments that runs on IBM-compatible computers under a Microsoft Windows environment.The program converts alignment formats used by all popular software packages for sequence alignment and phylogenetic tree inference.ForCon is available for free on request from the authors or can be downloaded via internet at URL http://bioc-www.uia.ac.be/u/jraes/index.html .It is also included in the software package TREECON for Windows (see http://bioc-www.uia.ac.be/u/yvdp/index.html).

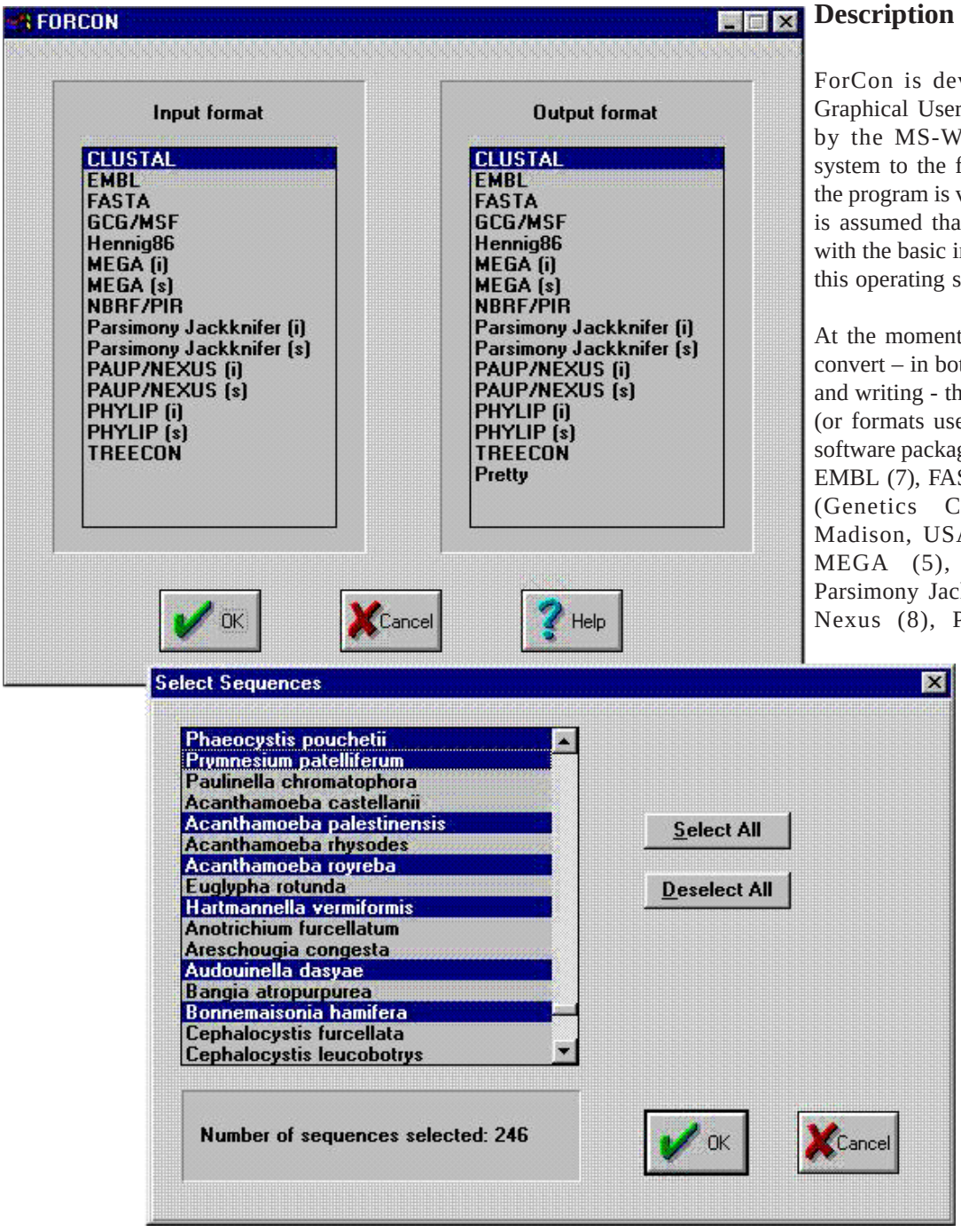
Introduction

ForCon is a user-friendly software tool developed for the easy conversion of nucleic acid and amino acid sequence alignments.Sequence alignments are indispensable for many applications such as the development of probes and the inference of evolutionary trees.At the moment, many software packages for sequence alignment and the construction of evolutionary trees have implemented their own standard of saving and reading sequence alignments.Converting one alignment format into another usually requires the use of a word processor and manual

editing.To circumvent this sometimes slow and tedious work, a program was written to convert different sequence alignment formats automatically.

System requirements

ForCon is written in C++ (Borland 4.5) and runs on IBM-compatible computers operating under a Microsoft Windows environment (Win95, WinNT 3.5, 4.0 and Windows 3.1).For the latter operating system, the Windows 32 bit extension should be installed.The complete installation of ForCon requires about 1.5 Mb of hard-disk space.As dynamic memory allocation is used throughout the program, the size of the sequence alignment is constrained only by the available memory, which allows the conversion of very large files containing many hundreds of sequences.Performance tests were executed on a Pentium 200 Mhz processor with 64Mb of RAM.On this computer, an alignment of 6900 sequences of 1060 nucleotides (7.4Mb) was converted from one format into another in less than 30 seconds.



Upper window: alignment format selection module
Lower window: sequence name selection module

Description of the program

ForCon is developed using the Graphical User Interface provided by the MS-Windows operating system to the full extent.Although the program is very simple to use, it is assumed that users are familiar with the basic interface elements of this operating system.

At the moment, ForCon is able to convert – in both ways, i.e. reading and writing - the following formats (or formats used by the following software packages): CLUSTAL (9), EMBL (7), FASTA (6), GCG/MSF (Genetics Computer Group, Madison, USA), Hennig86 (2), MEGA (5), NBRF/PIR (1), Parsimony Jackknifer (3), PAUP/Nexus (8), PHYLIP (4), and TREECON (10).Other software packages are usually able to read one of the above mentioned formats.A so-called 'Pretty' format can also be generated for the publication of sequence alignments.When sequential and interleaved formats are supported by the original program, they are also both implemented in ForCon.Online help – with examples of the different

alignment formats supported – is also available.

When the program is started, it asks for the sequence alignment input file format and the required output file format (see Figure).Next, a list of all the sequences in the input file is displayed and a particular group of sequences can be selected (see Figure).This is particularly useful since additional editing of the input or output file (e.g. deleting sequences that will not be used in an analysis) can be avoided this way and different data sets can be easily made starting from the same large input file.Furthermore, it is also possible to select particular regions or codon positions from the sequence alignment and to save these only to the new file.

Discussion

To our knowledge, ForCon is the only software tool currently available that converts the large number of sequence alignment formats used by most tree construction programs.The only alternative may be the ReadSeq program developed by D.G. Gilbert (Indiana University, USA), but regarding tree construction, only the PAUP/NEXUS and PHYLIP sequence alignment formats are supported.

ForCon is available for free and can be fetched via the internet at URL http://bioc-www.uia.ac.be/u/jraes/index.html . The program can also be sent via electronic mail on request.Due to the structure of the program, addition of new alignment formats is very easy and users can always contact the authors if implementation of yet another format is desirable.Development of a Java™ version of ForCon has been initiated in order to guarantee platform independence in future versions of the program.

Acknowledgements

Our research is supported by the Special Research Fund of the University of Antwerp.Yves Van de Peer is Research Fellow of the Fund for Scientific Research – Flanders.

References

1.Barker, W.C., J.S. Garavelli, D.H. Haft, L.T. Hunt, C.R. Marzec, B.C. Orcutt, G.Y. Srinivasarao, L.-S.L. Yeh, R.S. Ledley, H.-W. Mewes, F. Pfeiffer and A. Tsugita. 1998. The PIR-international protein sequence database.Nucleic Acids Res. 26:27-32.
2.Farris, J.S. 1989. Hennig86: a PC-DOS program for phylogenetic analysis.Cladistics 5:163.
3.Farris, J.S., V.A. Albert, M. Källersjo, D. Lipscomb and A.G. Kluge. 1995. Parsimony Jackknifing outperforms neighbor-joining.Cladistics 12:99-124.
4.Felsenstein, J. 1993.PHYLIP (Phylogeny Inference Package) User Manual.Department of Genetics, University of Washington, Seattle.
5.Kumar, S., K. Tamura and M. Nei. 1993. MEGA: Molecular Evolutionary Genetics Analysis User Manual.Pennsylvania State University, University Park.
6.Pearson, W.R. and D.J. Lipman. 1988. Improved tools for biological sequence comparison.Proc. Natl. Acad. Sci. USA 85:2444-2448.
7.Stoesser, G., M.A. Moseley, J. Sleep, M. McGrowan, M. Garcia-Pastor and P. Sterk. 1998.The EMBL nucleotide sequence database.Nucleic Acids Res. 26:8-15.
8.Swofford, D. 1993. PAUP (Phylogenetic Analysis Using Parsimony) User Manual. Laboratory of Molecular Systematics, National Museum of Natural History, Smithsonian Institution, Washington, D.C.
9.Thompson, J.D., T.J. Gibson, F. Plewniak, F. Jeanmougin and D.G. Higgins. 1997. The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Res. 24:4876-4882.
10.Van de Peer,Y. and R. De Wachter. 1994. TREECON for Windows: a software package for the construction and drawing of evolutionary trees for the Microsoft Windows environment. Comput. Applic. Biosci. 10:569-570.

It goes to the library. You go to the pub.™

Summary

PubCrawler is a new tool that acts as an alerting service for users of the NCBI PubMed and GenBank databases. Users will periodically be informed about new papers or sequences of interest in these databases. Results are presented as an HTML page which allows usage of browsers for viewing and the integration of hyperlinks. PubCrawler can be automated which spares the user from frequent manual searches for new database entries of interest, bypasses rush hours on the internet, and provides results at the most convenient time.

The program was written in Perl, is freely available, and can be run under MacOS, Win95/98 or UNIX. Additionally, a Web interface is available (still at the experimental stage) which also helps generating configuration files.

Availability: http://www.gen.tcd.ie/pubcrawler

Contact: pubcrawler@acer.gen.tcd.ie

Introduction

Using bibliographic databases on the internet for the search of specific articles is becoming increasingly popular. To stay up to date with the latest development in scientific research, scientists should search regularly for new articles in their fields of interest. Some commercial services, such as ISI's Current Contents Online, provide facilities to do this automatically but these services are often expensive or restrictive in their options. Here we describe free software that provides an alerting service for the largest public-access biomedical database - NCBI PubMed [1, 2]. This is a huge archive of journal abstracts containing over 9 million records which can be accessed for free. The huge number of published articles requires very specific search criteria. Several issues arise from carrying out online queries at PubMed with a Web browser:

1. The same (often lengthy) search strings have to be typed in again and again.
2. New articles can be hidden amongst many others that have been seen before.
3. During peak hours network traffic can slow down and cause long waiting periods.

PubCrawler offers a cheap and efficient solution to these problems. The program was designed to communicate through the internet with Entrez [3], a data retrieval system created by NCBI. Databases that can be accessed with PubCrawler are PubMed and GenBank [4].

The same powerful syntax that is available for Entrez [5] can be used to generate complex search strings. They are stored in configuration files and therefore only have to be written once. Query results are compared with searches carried out beforehand and only new entries are presented. This makes quick scanning for new articles in a specific research field much easier. PubCrawler's output is written to an HTML page that can be viewed with a Web browser. This allows the usage of hyperlinks to retrieve complete article abstracts, to include older results, or as a reference to an excessive number of hits. With the help of a scheduler like cron (comes with Unix, shareware for MacOS, equivalent to Windows' Scheduler) the program can be run at arbitrary times and days. Scheduling it to start every night at three o'clock, for example, would ensure that the user has the latest results available every morning when he starts work. This is also an important means to avoid high network traffic and peak hours at NCBI's Entrez server.

Usage

Before running PubCrawler, the user has to specify search strings and store them in a configuration file which will be read by the program. Each string consists of the database to be queried (PubMed or GenBank), one or more search terms, a set of predefined search fields and boolean operators (AND, OR, BUTNOT). Additionally an alias can be used as a short description for a long search term or to combine the output of several queries into a single one.

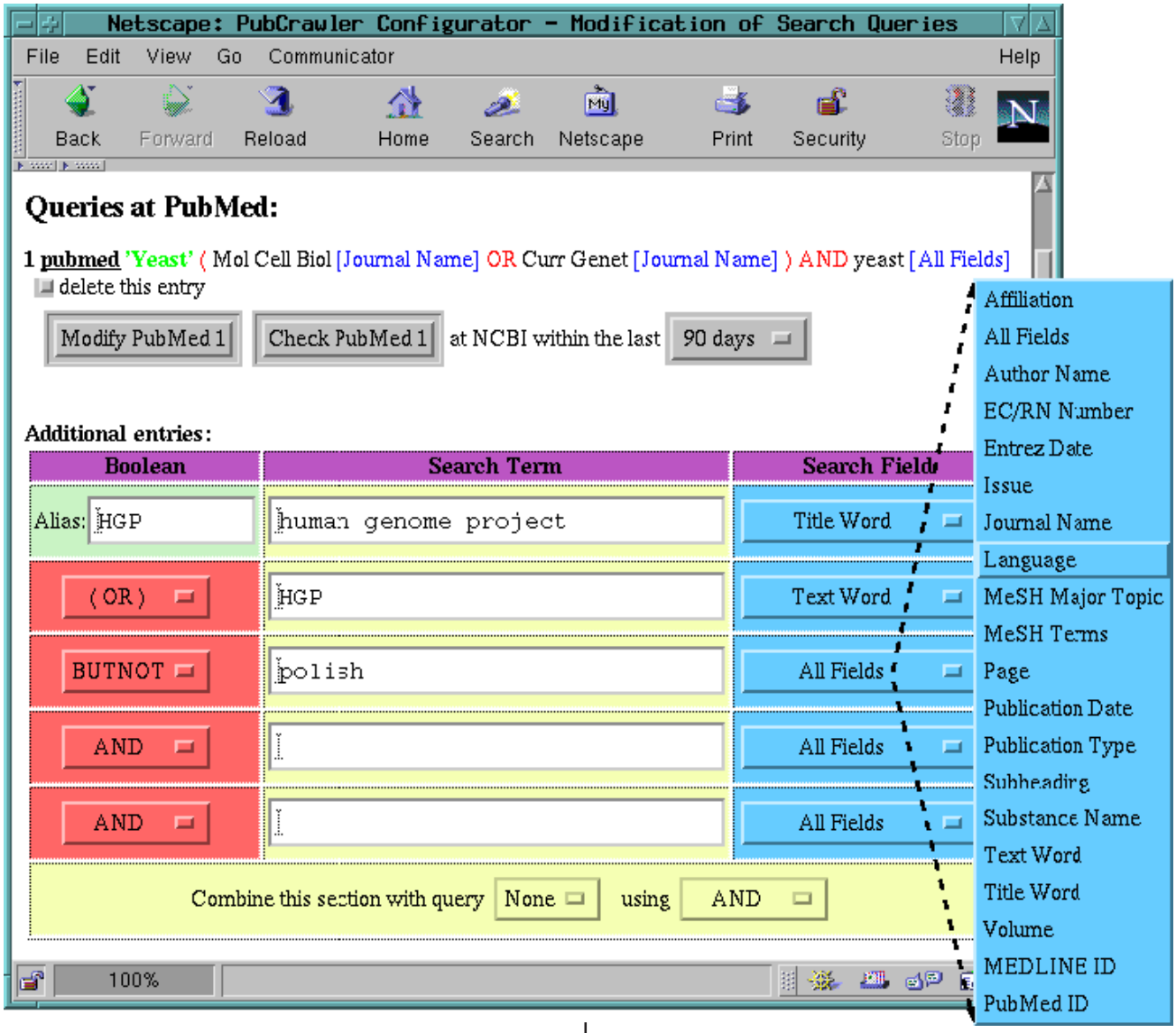
The following example shows an extract from a configuration file (hash symbols mark comments):

```
# The next query searches PubMed for yeast-
# specific information with particular
# emphasis on two journals:
# (The word 'Yeast' will be used as an alias.)

pubmed 'Yeast' ( Mol Cell Biol [JOUR] OR Curr
Genet [JOUR] ) AND
yeast [ALL]

# The next query searches GenBank for all
# human sequences with length between 50000
# and 350000 bases, where JC Venter appears
# as author: (This search has no alias.)

genbank 50000:350000 [SLEN] AND Venter JC
[AUTH] AND human [ORGN]
```

In addition to the search strings the configuration file also holds values for variables, that control the execution and the output of PubCrawler. There are currently 29 parameters available which can be set in the configuration file or via command line options. They allow for a precise adjustment to the user's preferences and the computational environment.

Since the configuration file determines the success of the output of PubCrawler a thorough and correct specification is important. The syntax of the search strings in particular might cause difficulties for novice users. We have set up a web site, PubCrawler Configurator [6], which guides users through the process of generating their configuration file. It also offers the possibility to check search strings at NCBI immediately. This will give a feeling for the kind of output that can be expected from each query. The following graphic shows a snapshot of the PubCrawler Configurator, where one search string has been set up already and another one is being constructed (see next page).

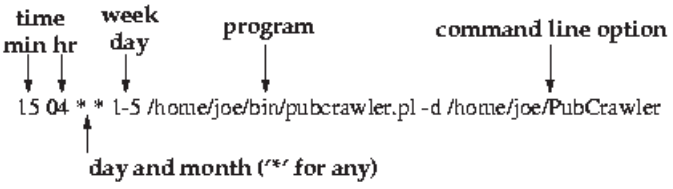
Once the configuration is finished, PubCrawler can be executed in a so-called 'check'-mode, which simulates a run without actually carrying out the queries. This feature allows



verification that everything is set up correctly and helps avoid disappointments the next day in case a wrongly configured PubCrawler was run automatically at night.

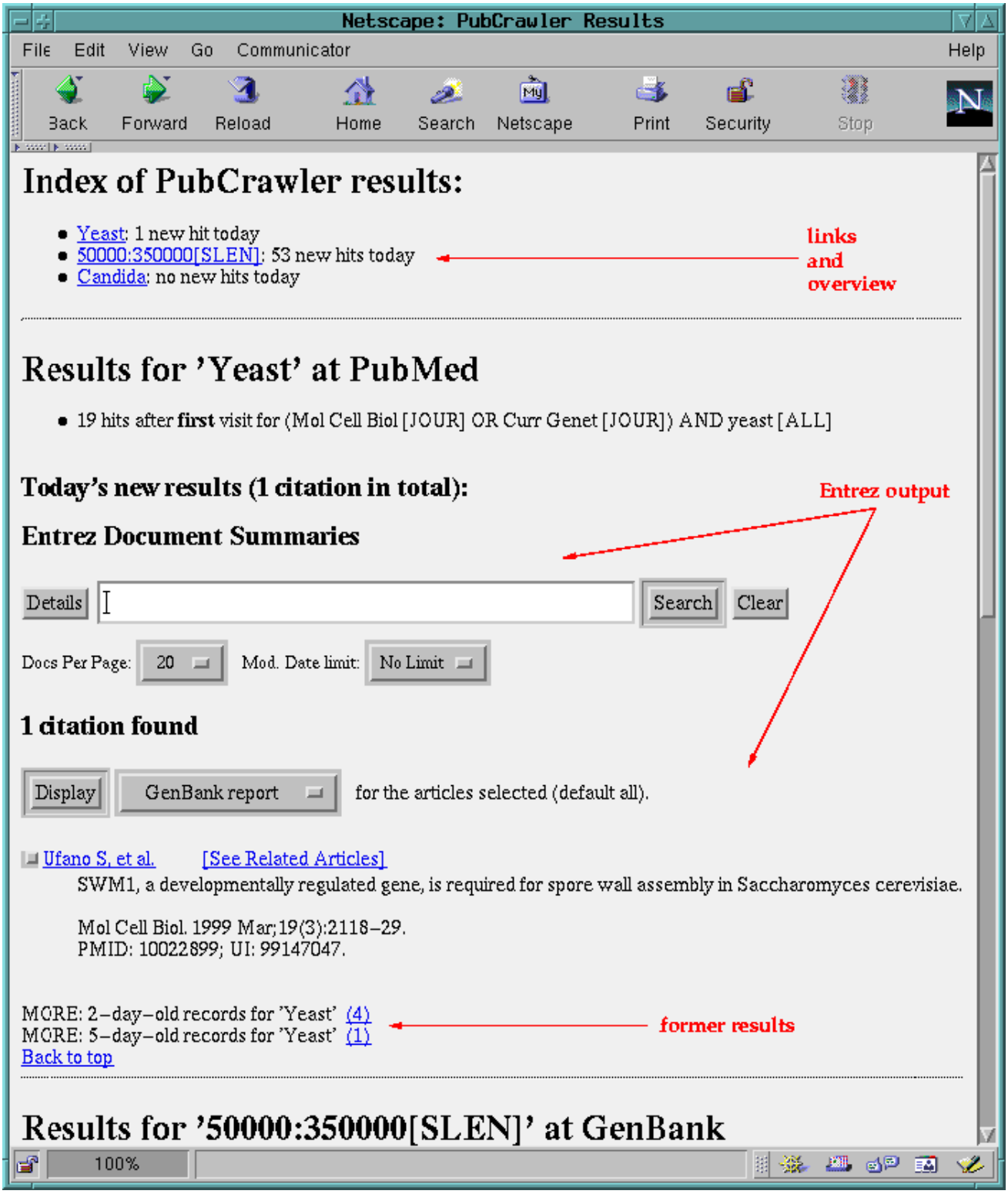
This brings us to the next point - the automation of the program. Since PubCrawler is run and controlled from the command line it is perfectly suited for schedulers. The most famous one is undoubtedly cron which is available on every UNIX system. The equivalent for Windows system is the Scheduler, which is part of Windows 98's system tools and also included in the Plus-package for Windows 95. For MacOS a $10 shareware tool named Cron for Macintosh [7] is available. These little programs can be used to start PubCrawler automatically at times when people are long gone home (or to the pub!). They should be scheduled in a way to avoid high network traffic and busy server times (NCBI's off peak hours are currently from 1am to 1pm GMT).

The following example shows an entry in a crontab file (for UNIX) that would start PubCrawler every weekday morning at 4.15 a.m.:



Besides the time and days any command line options can be specified (like the PubCrawler directory in the example above). More information on the setup of PubCrawler is given on its homepage.

Depending on the number of search queries and the speed of the internet connection the program execution can last between a few seconds and several minutes. After successful completion an output file like the following will be produced:



How it works

PubCrawler makes two HTTP connections (visits) to NCBI. The number of documents retrieved and presented to the user is limited by several parameters, specified in the configuration file. These are:

RELPUBDATE
VIEWDAYS
FULLMAX

maximum age of database entries to be reported
number of days each document will be shown

the maximum number of documents for which a full report is being presented

The first visit collects the UID (unique identifier) numbers of the database entries that match any of the specified queries, provided that their "date stamp" (the date they were added to the NCBI database) indicates that they are younger than the number of days specified by the variable RELPUBDATE.

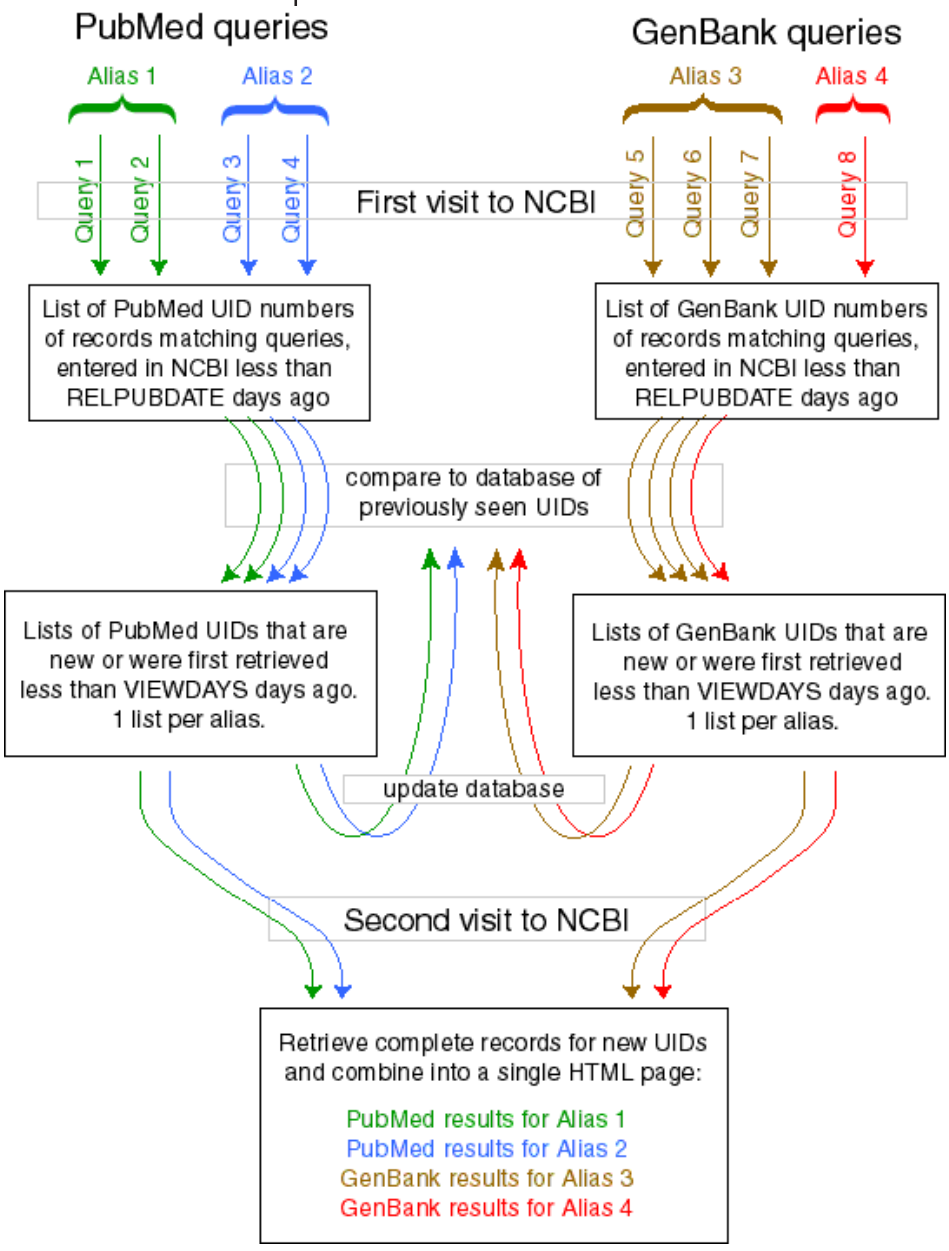
The UID lists are then compared to a database (containing both PubMed and GenBank UIDs). The database stores lists of UIDs and the calendar date that PubCrawler first found each UID (this date may be different from NCBI's date stamp, depending on how frequently PubCrawler is being run). Interesting UIDs -- those that are new or are younger than the VIEWDAYS variable -- are chosen for display.

If the same alias is used for multiple queries, the UIDs matching any of these queries are merged. This means that if the same UID is matched by several queries (within one alias) it will only appear once in the list of hits. If a UID is matched by queries with two different aliases, it will appear in both hit-lists.

The database is updated, and any UIDs that are older than RELPUBDATE are deleted (this prevents the databases from growing indefinitely using the '-d' flag).

A second visit is then made to NCBI to retrieve the complete records for the new UIDs, and these are shown on the results page, grouped alias-by-alias. If a lot of new records have been hit only the first FULLMAX are shown for each alias, with HTML buttons for the rest. The results for each alias also include HTML buttons that allow the user to retrieve the complete records for older entries (those that are not new today, but are less than VIEWDAYS days old).

The following diagram sums up the different stages of a PubCrawler run:



References and Links

1. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov/
2. The NLM PubMed Project, http://www.ncbi.nlm.nih.gov/PubMed/overview.html
3. Entrez, http://www.ncbi.nlm.nih.gov/Entrez/
4. Benson D.A. et al. (1999) Nucleic Acids Res. 27, 12-17
5.http://www4.ncbi.nlm.nih.gov/PubMed/syntax.html
6. PubCrawler Configurator, http://www.gen.tcd.ie/pubcrawler/configurator.html
7. Cron for Macintosh, http://gargravarr.cc.utexas.edu/cron/cron.html

VRML in Molecular Biology

Bringing Virtual Reality to Biology

José R. Valverde - EMBnet/CNB - Madrid, SPAIN

This article shows how VRML can be used to construct dynamic, interactive virtual worlds for use as presentation tools in Molecular Biology.

The problem of demonstrating biological processes

Demonstrating biological concepts with traditional tools is very difficult since biological organisms are complex entities, made up of components that are complex too, and that display complex interactions:

A biological system is composed of real things with three dimensions. It is very difficult to get an idea of its aspect using images and textual descriptions. Furthermore, they are consistent, i.e. it is not easy to see their inner parts without tearing them apart. A virtual 3-D model allows the user not only to manipulate the object freely, looking at it from many points of view, but also allows getting in and out of objects to look at them from virtually any position without destroying them or disrupting their behaviour. Even with good models, their complexity is still a problem which may hide important features among excessive non relevant details. We need special visualization methods to provide enhanced perception ofinteresting properties.

Biological systems are never static, they are dynamic entities in a continuous flow of change, interacting with the environment and with each other. We can't understand them without understanding their behaviour in time. And this, too, has been traditionally difficult, the only tools available being movies -perhaps with added 3D stereo effects-. What is required is a tool that allows users to immerse in the process, move around while things happen and even alter the system behaviour at will until they feel satisfied.

In the following sections we will see how we can better achieve these goals using Virtual Reality models.

What is VRML?

VRML is a shorthand for Virtual Reality Modeling Language (pronounced vermal). VRML is a difficult beast to define, like the blind men in the tale that had different perceptions of the elephant, there are various different perceptions about what VRML is (or may be).

It is a language devised to become a three-dimensional equivalent of HTML, i.e. a 3-D extension to the WWW where instead of navigating through literate documents, you would be navigating through three-dimensional objects. The goal, obviously,is to produce a realistic model of the world where navigation may be more intuitive.

This takes us to a different view: VRML can be used to build models of the world. These models may include 3-D objects, envolvent stereo sound, movies, animations, interactivity and navigation through hyperlinks. With these tools you may specify realistic models of almost anything you can imagine.

Models may be visited with an appropriate program, and depending on the program capabilities and the hardware you have, rendering and interaction may be more or less realistic. The program might only be able to render a 2D flat image, or produce a fully featured experience with datagloves, stereo headset and eye goggles. At its bottom level, VRML is a portable format for the exchange of animated 3D information.

On the other hand, a good use of VRML tools may result in an immersive environment, an interface of such quality that the user feels like he is actually there, inside it, and almost fully out of the Real World.

As a modelling tool it supports models of almost anything, from houses to subatomic particle interactions to imaginary worlds, and association of objects in the model with actions, links, and behaviours (how objects interact with each other and with users).

You can think of VRML as the World Wide Web in a three-dimensional, interactive virtual world. This world is fully networked -like the WWW- and accommodates seamless extension and navigation as well as inter-host communication.

Thus, VRML goes way beyond HTML: it is not only a navigation, visualization and audition tool, it allows you to interact with other users too: i.e. it accommodates multi-user worlds where many people might be present simultaneously interacting and communicating with each other and the objects in the world.

And so, we arrive at another view of VRML as the foundation of cyberspace and future virtual communities as popularized by science fiction writers like William Gibson and Neal Stephenson (Nota bene: if you haven't read Neuromancer or Snow Crash, you should, they not only make a good reading, they also are the original definition of the cyberspace as we conceive it).

If you want to learn more details, the best starting point isWeb3D, the home page of VRML.

Using VRML

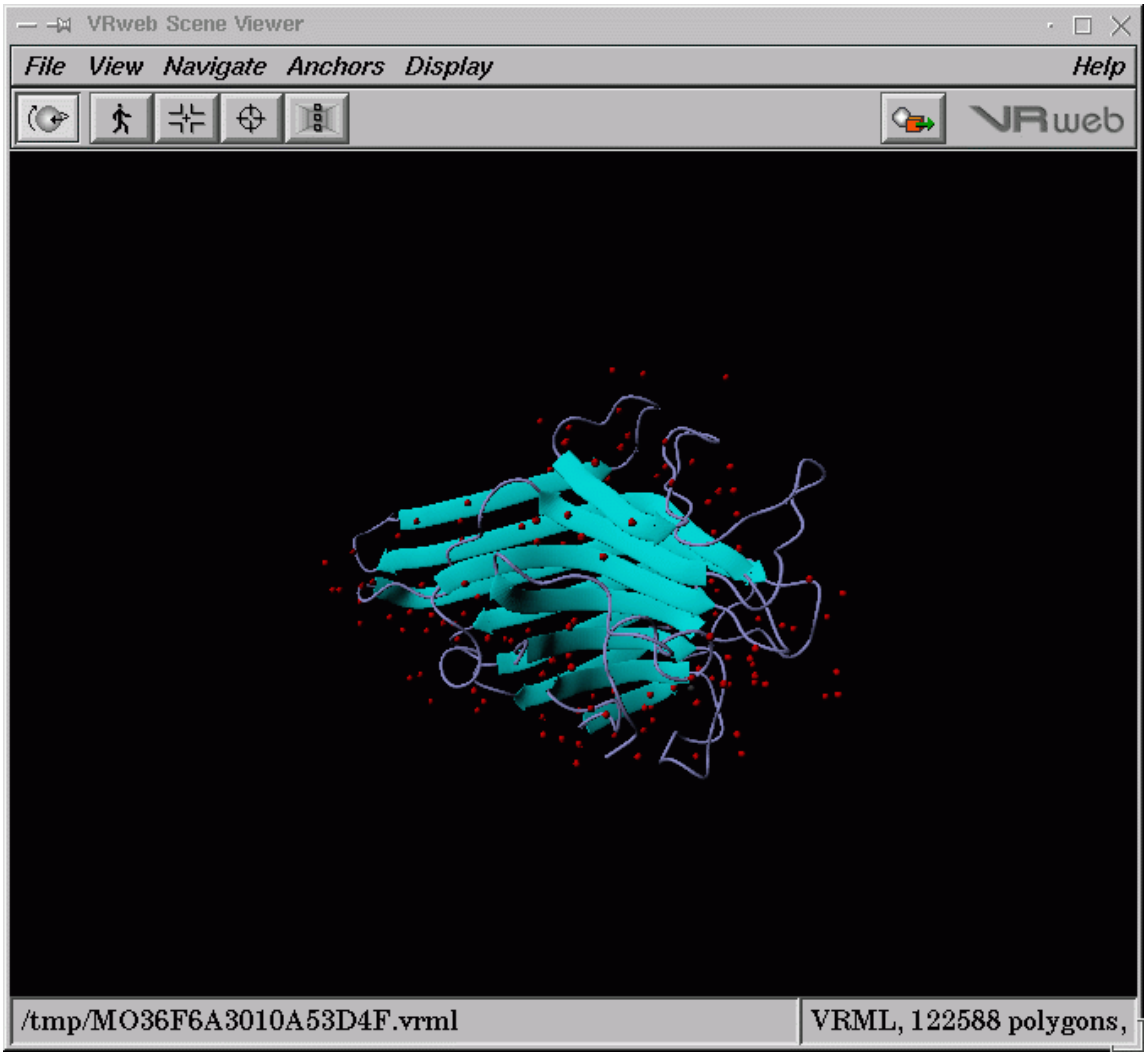
If you have had enough of this propaganda and want to learn how this actually looks like in reality, then read ahead. In this section you will discover that VRML is actually very easy to use.

In order to navigate through VRML worlds you need a VRML browser. Luckily there are many such browsers available for almost any operating system you want, and what'smore they are often free. See The VRML Repository for pointers.

There is only one caveat: early versions of VRML were rather limited, which led to a revamping of its specifications and a new definition of the language. Worlds written according to the first definition (VRML 1.0) are no longer compatible with later worlds (like VRML 2.0 or VRML 98). Therefore, either you have a VRML browser that understands both specifications of the language or, if you want to visualize both ancient and new worlds, you will need two browsers, one for worlds written according to VRML 1.0 specification and the other for files written in VRML 2.0 and later.

This said, how do you go about using your VRML browser? Easy: it may be an independent program or a plug-in that becomes part of your WWW navigator, but in both cases it suffices tocomplete the URL (the network address) of a VRML world, by clicking on a link or entering the address directly. For instance:

Usually, when you reach a VRML link, your WWW browser will load the description of the virtual world and open a window to show the contents, play any sound and movies and allow you to interact with the world and its objects. If you are lucky, you may even be inmersed in the world with stereo glasses and interact with it using your own hands and body.



That's all, simply enter a URL or follow a link from a web page, e-mail message, news posting, whatever, and you are done.

VRML use in Science

So far, we have seen what VRML is and how easy it is to navigate virtual worlds. Our next goal is to see how this can be used as a tool in Molecular Biology. There are various ways in which you can use VRML to present scientific data.

General navigation

The most obvious application is to use it as a replacement for the WWW, providing a 3D world where visitors may find things more easily by using intuitive analogies:

For this, we would provide a virtual world where selecting special objects would launch demonstrations, display lectures or take them to special pages. For example, we might design a virtual room filled with biological specimens and touching any of them might open a WWW page with detailed explanations on them or on related topics. It might be as simple as the navigation system of VRML in Chemistry where clicking a geometric figure takes you to various other places. You can see also an example further below.

An additional advantage is that we are not limited to Real World objects or room allegories, we may as well model a test tube filled with molecules and allow users to jump to descriptions of metabolic pathways by clicking on these molecules. See the examples below for a very simple demo. This example shows that we can go beyond providing a 3D WWW lookalike.

Virtual Molecular Biology

A great advantage of virtual worlds is that we devise them and hence we state the laws governing their behaviour. We are no longer constrained by physical laws and may describe any object and/or behaviour we wish no matter how implausible it may be.

From a Biological point of view this means that we can model any behaviour, or mark any property with suitable attributes, for example, we might display flexibility changes in a molecule or membrane as colour changes, mark the steps of a reaction with sound effects, cut a representation of DNA with a pair of scissors, etc... See below for some examples.

Attributes may also be motion, reactivity to stimulus, interactivity among objects and with the user, changes in time, etc... Adding these kinds of properties allows us to model complex biological events, providing visitors with de luxe views of them. This way they may see how an enzye processes its substrate, or how a muscle contracts, visualize complex molecules and their interactions, and so on. In one word, they can inmerse inside biological systems and move around them to gain better insights on how they are built and how they work.

And what's more, the user himself becomes an extra object of the world when he comes pay a visit. As such, the user can participate too and interact with the modelled objects. User actions may be made to trigger reactions, or he could take in his hands molecules and try to approximate them, looking for alternative docking mechanisms and in general, interact with the virtual world in all sorts of ways.

VRML vs. other specialized software

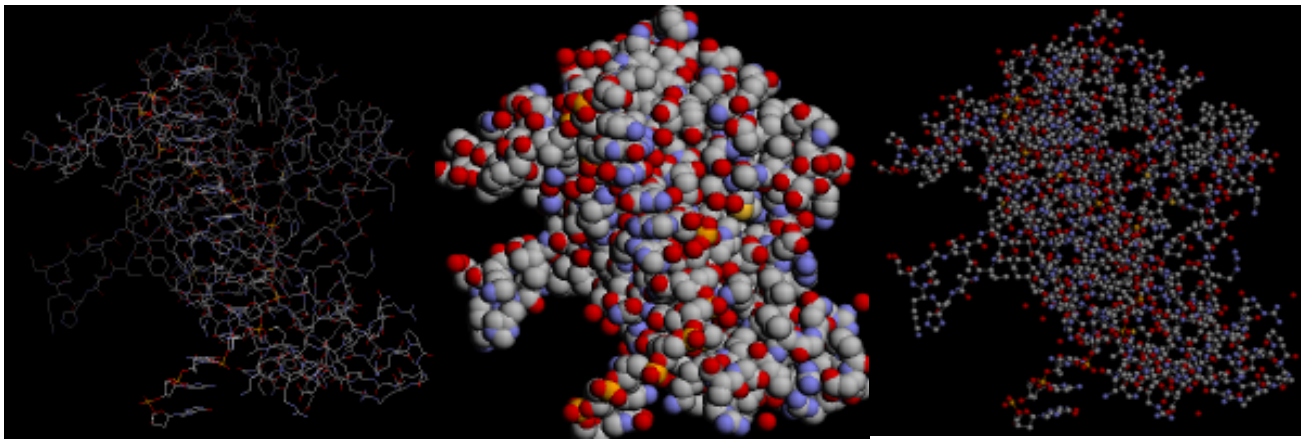
Here we will review other approaches and tools that are commonly used to achieve similar results and see why using VRML may actually be a good idea.

As we have already mentioned, there are already many tools that help us present scientific data in Biology. Some of them are very well established and are commonplace today, like text, pictures, movies and 3D effects achieved with stereoscopic glasses. Many of them have already been translated to the WWW, as demonstrated by Molecules-R-Us.

These are the traditional generic tools. With the coming of computers, more specialized tools have appeared which are tailored for special biological systems. RasMol, for instance, is a popular tool among Molecular Biologists for visualization of macromolecular structures. Their main limitation is their specialization: they are devised for the specialist and thus are not intuitive for novice users (like students) or scientists coming from other fields. For example: imagine a student looking at a wireframe model of a protein to understand what an alpha helix may be and trying to decide if a spacefill or ball-and-stick model will show helices any better. As an example ofhow users can employ RasMol to see downloaded structures, see this article.

Using VRML allows you to select which features are shown and how they look and provide that model to your users. They only need a standard browser, and you can be sure they see immediately what you want. If you so want, you may provide alternative models and a way for your users to switch among them, just like if they were using RasMol or any other such tool. Compare the above images with what you get from the previous VRML example to see what we mean.

We haven't played fair to specialist tools here. It is true that you can achieve similar results with some of them, but this



will usually require that the user also downloads a script that will process the sample to produce the desired representation. An example of using scripts to automate RasMol tasks is provided in theTopits2Rasmol home page. With this approach our student does not need to learn RasMol, but he will have more work to do, and if we want everything to happen automatically we need to define a special MIME type and tell all prospective users how to use it. A rather difficult task if these are spread over the World, like remote students or colleagues from other institutions. And all ofthem still need to get the specialized tool we have chosen to use our script and be able to see the objects.

When using VRML we would do the processing ourselves with a suitable specialized tool, then we would convert the processed model(s) to VRML and provide it to our users. They will not need any extra steps, and what's even better, they will use a standard, generic tool -the browser- to visualize the model, instead of being forced to get, or even buy a potentially expensive specialist tool simply to glance at our results. We can always provide the original files for the specialist, but lay users will still be able to study our work. We might even provide alternate models or methods for users to select or change the properties of the model they get, as is shown in the VRML in Chemistry page.

It is when we want to enter into more complex models where we can gain the most. Specialist tools may allow you to build customized worlds with pieces taken from separate parts and arrange them at will, but this usually implies more difficulties or goes beyond the abilities of popular packages. For example, it is not trivial to gather molecules from separate PDB files and arrange them at will using RasMol. Instead, we might use MolMol or some other program and provide a tailored script but then we start requiring our users to install further programs, MIME types, etc...

Things get tougher as we add behaviours to our models, specially when they are interactive behaviours. Simply put, there are not that many tools that allow you to associate as much arbitrary dynamical properties to objects in a model without getting into complex programming, and even so, some things -like multiuser worlds- are plainly impossible. We can -and usually do- build our models using specialized tools, but we can not require every user to use them too. At this point what we need is a simple, foolproof system that anybody can use and that is widely available. Either we build a new program tailored for each model, or we use a generic modelling tool, like VRML.

All these advantages come at a cost. It means that instead of providing our raw experimental data to our readers, we need to create the models ourselves and convert them from whatever tools we use for our work to this language. For the user, too, it implies an extra cost: to build VRML models we need to convert all our data to spatial coordinates and this may result in huge files and conversely in long and costly download times on slow network connections. Furthermore, as the complexity of the virtual world increases, so does the need for memory and computer power by the user.

Here, the use of specialized software may have an edge since it may be able to process simplified representations of data, or reduce complexity using specialised knowledge and assumptions (e.g. an atom might be specified by its center coordinates instead of a sphere or a series of polygons). Actually, a bad decision when building a VRML model may result in files that are prohibitively big or complex if we are not careful when choosing the level of detail used.

Authoring VRML

It is time to review now how we can start using the advantages of VRML. In this section we will see a few basic approaches to generating virtual reality worlds with VRML, so you know where to start to make your own models. As we shall see, it is not difficult at all to get started, and you can get nice results with little effort.

The easiest way to start is to use a specialized scientific tool that can generate VRML worlds. You just feed it your scientific data, process it to get the world you want and save it as a VRML file. For example, there are various programs

that can convert molecular structures to VRML worlds:MolMol,pdb2vrml or MolScript are a few of them.

Once you have got yourself started, if you feel adventurous, you may want to add other objects to the scene that can not be generated by these scientific tools. It is time to introduce a VRML authoring tool. This is simple a 3D editor that allows you to create 3D worlds and save them as VRML and to read previous models and modify them. The VRML Repository is a good place to start looking for these kind of programs. These tools are not generally very useful for building biological models from scratch since it would be too cumbersome to add by hand all the elements unless it is a very simple system. Instead you will normally generate the initial model using a specialized scientific tool, and then read it into one of these programs to combine with other models or add new objects, properties, etc...

Finally, you can always try and do it the hard way, entirely by hand, writing the world description directly in VRML using a text editor. VRML is a simple language and creating worlds using it is rather easy, see theLighthouse tutorial (or our local mirror) to get started. But again, this is inconvenient for very complex worlds like those used in Biology. Imagine yourself trying to type the coordinates of all the atoms in a simple protein. However, it is often very useful to know VRML for finely tuning your models or adding special behaviours that are not easily expressed with current authoring tools. As usual with any fine piece of art, the final touches are better done by hand, and you can dramatically enhance your models by hand coding (explicitly expressing) your ideas.

The examples below further expand on these ideas.

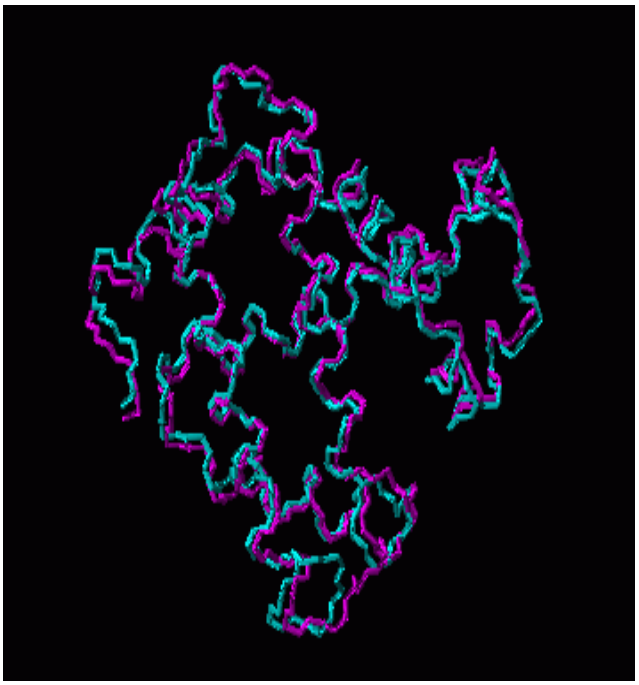
A few VRML examples

Here you will find a few examples demonstrating some of the capabilities of VRML. They are not exhaustive and I have kept them simple to reduce the size of the models, but you can see most of the basic principles and advantages of using VRML.

We start with a VRML 1.0 model that represents the aligned backbones of two molecules, human and horse myoglobin. I used this to illustrate the utility of aligning molecules to spot related properties and similarity.

The following VRML 1.0 model represents the backbone of the myoglobin molecules of human and horse. Both molecules were loaded into MolMol, aligned to show their structural similarities, backbone bonds were selected to clean up visualization of structural features and different colors were assigned to each of the molecules to facilitate the distinction.

To build the model, both molecules were loaded into MolMol and aligned to show their structural similarity. To make the



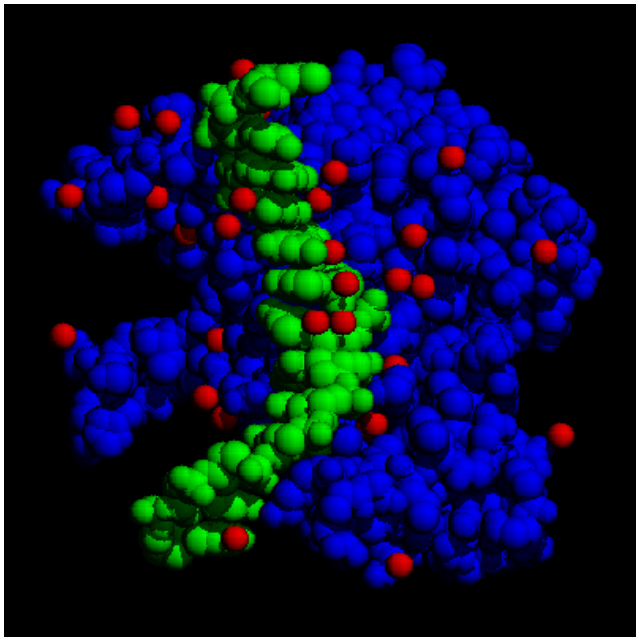
similarity more evident, I chose to display only backbone bonds, hence cleaning the image from superfluous details, and then selected each molecule separately assigning a different colour for each one to facilitate the distinction.

Despite its apparent simplicity this is an example of something that requires various complex steps to get to a useful visualization. It required an appropriate combination of molecules, filtering of meaningless elements and assignment of special properties (here only colour) to highlight important features. Getting to it using MolMol may be easy for advanced users, but would be a labyrinth for the novice.

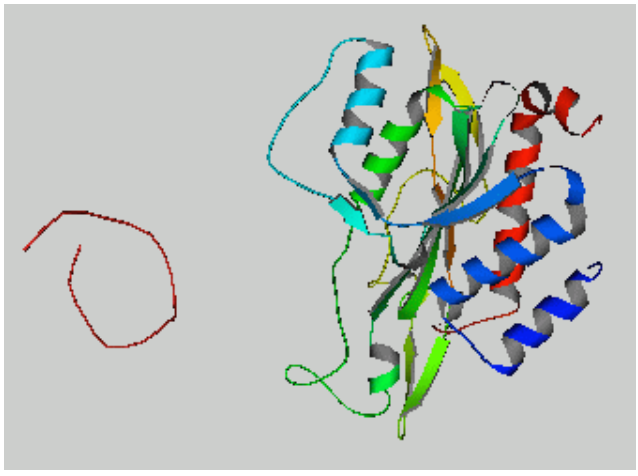
Next we deal with a VRML 1.0 model of the restriction enzyme EcoRI with its DNA substrate. Each of the molecules is a hyperlink to a separate web page: clicking on them will open the linked page in your WWW navigator. While the pages selected do not say much, they suffice to show how you can use this feature to convert your world in a navigation tool. All that is needed is that the links point to more useful pages, or to other VRML worlds, or to any other place where additional or related information may be found.

This model was generated loading the PDB structure into MolMol and saving it as a VRML model. Since I only wanted to addthe hyperlinks, all I had to do was hand-edit the model to include the appropriate nodes with the links to the desired web pages.

The following are VRML 2.0 models. I chose VRML 2.0 to



take advantage of its advanced properties, like sensors, timers and interpolators. Using them we can add dynamic properties to the objects in our models. We begin our dynamic modeling with a model of EcoRI and its DNA substrate. Here, the DNA substrate is placed away from the enzyme molecule, and clicking on it (the DNA) makes it start moving towards the molecule until it finds its docking position. You can click on the DNA molecule as many times as you want, and every time it will start its promenade again. While it is moving you can move the model, change your position freely and see how it all happens from anywhere in the virtual world.



This model was a bit more complex to generate. In this case I wanted to use VRML 2.0, but MolMol only generates VRML 1.0 models. Hence, I loaded the molecules, chose a suitable point of view and saved them in PDB format. This was to be the final position of DNA coupled to the enzyme. The next step was to use MolScript to generate the VRML 2.0 model that I was to use. This is a static model to which I wanted to add movement. All that I had to do was to edit it to define the appropriate transformations to specify the initial positions of the molecules, associate a touch sensor to the DNA and define a behaviour for the DNA when touched (a few lines of VRML code).

I finalize with a very simple model of the interactions of an enzyme with its substrate. Since all I want is to show how to implement basic interactions, and how properties like colour may be dynamically changed I have kept this example as simple as possible. Actually it is so simple that it has been entirely written by hand.

A nice feature of this model is that it is very easy to convert into a more realistic simulation. All that is needed is a substitution of the basic models of the enzyme and its substrate by the corresponding molecular models (a simple cut and paste operation) generated by any program, and an adaptation of the transformations to fit the sizes of the molecules.

There are many more things that can be done. You may attach proximity sensors to objects to detect when another object (or the user) is getting too close (useful for simulating electrostatic interactions for example), map textures on objects (useful for showing contour maps for example) or associate alternative representations to an object (so that e.g. when a user pushes a virtual button the representation changes for instance from a spacefill to a wireframe model), include sounds, movies, etc... There is virtually no limit to what you can do. Actually, VRML includes support for scripting languages like Javsacript and Java, hence there is no limit to the complexity you can give your models.

Node News

Upcoming courses

TITLE: 2nd Annual Irish EMBnet Course

APPLICANT: Andrew T. Lloyd INCB

DESCRIPTION:

There is a clear demand for a basic course in bioinformatics among the molecular biological community in this country. The EMBnet Course in December 1998 was 3x over-subscribed.

Sequence analysis, multiple sequence alignment, database interrogation and phylogenetic trees are the topics most often asked for. This 4 day course will raise awareness of bioinformatics tools and resources and has clear educational benefits for the participants.

This course is being held in the second largest city in the country, to extend the catchment area and facilitate provincial molecular biologists

TITLE: Linkage analysis

APPLICANT: Kimmo Mattila CSC - Center for Scientific Computing Finland

DESCRIPTION:

At the moment CSC has several genetic linkage analysis programs available. Even though these programs are installed on powerful servers they are not extensively used. This is not because there would not be enough researchers interested in the topic, but the reason seems to be the lack of know how and tradition in using these methods.

To improve the situation CSC would like to organize a five days specialist course on linkage analysis. The course is intended for scientists and graduate students studying genetics. The aim of the course is to present the basics of the theory of linkage analysis and give on overview of some of the tools available. The course includes lectures, given by Dr. Joseph Terwilliger from the University of Colombia, and hands-on exercises, that are guided by Pekka Uimari (CSC).

We encourage the participants to bring with them their own real life sample problems to be discussed and studied in the exercises.

The course will be held in the premises of CSC in Espoo, Finland. The course is free of charge for academic researchers in Finland and in other EMBnet countries, however the class room limits the amount of participants to 20.

TITLE: Bioinformatics for System Managers

APPLICANT: Jose R. Valverde EMBnet/ CNB

DESCRIPTION:

The goal of this course is to inform system managers of the basic concepts of Bioinformatics, the most commonly used tools and databases and their installation and maintenance, so they can better assist users in their respective institutions and provide them with the basic tools for
Sequence Analysis.

TITLE: Bioinformatics

APPLICANT: Jose R. Valverde EMBnet/CNB

DESCRIPTION:

The goal of this course is to provide a general introduction to Bioinformatics for inexperienced users, including both, lecture presentations and practical tuition on the computer.

The course should inform new users in the basics and possibilities of bioinformatics and increase visibility of EMBnet and user awareness of EMBnet services.

TITLE: EMBarchive - an Eightfold path to better bioinformatics teaching

APPLICANT: Andrew T. Lloyd INCBI

DESCRIPTION:

A project to gather, from the EMBnet community, materials that will help run EMBnet training courses in bioinformatics and redistribute them to EMBnet.

The EMBnet Nodes

National Nodes

Argentina
Dr Oscar Grau
IBBM Facultad de Ciencas Exactas Universidad Nacional de LaPlata Argentina
Email: grau@biol.unlp.edu.ar
Tel:+54-21-250497 Fax:+54-21-259223

Australia
Dr Tim Littlejohn
ANGIS Electrical Engineering Building J03 University of Sydney
Sydney NSW 2006 Australia
Email: tim@angis.org.au
Tel:+61 2 9351 2948 Fax:+61-2-9351 5694

Austria
Dr Martin Grabner
BioComputing Centre Vienna University
Computing Centre Dr Bohr Gasse 9 Vienna.
Email: martin.grabner@cc.univie.ac.at
Tel: +43-1-4277-14141 Fax: +43-1-7986224

Belgium
Dr Robert Herzog
BEN Université Libre de Bruxelles CP300 Paardenstraat 67 1640 Sint Genesius Rode Belgium
Email rherzog@ulb.ac.be
Tel: +32-2-6509762 Fax:+32-2-6509767

Canada
Dr Christoph Sensen
National Research Council of Canada Institute of Marine Biosciences 1411 Oxford St Halifax Nova Scotia Canada B3H 2Z1
Email: sensencw@niji.imb.nrc.ca
Tel:+1-902-4267310 Fax:+1-902-4269413

China
Professor Jingchu Luo
College of Life Sciences Peking University Beijing 100871 China
Email: luojc@lsc.pku.edu.cn
Tel:+86-10-6275 5206 Fax:+86-10-6275 1843

Cuba
Dr Ricardo Bringas
Centre for Genetic Engineering PO Box 6162 Havana Cuba
Email: bringas@cigb.edu.cu
Tel:+53-7 218200 Fax:+53-7 218070

Denmark
Mr Hans Ullitz-Moller
BioBase - Danish Human Genome Centre Aarhus Universitet Ole Worms Alle 170-171 DK-8000 Aarhus C Denmark
Email: hum@biobase.dk
Tel:+45-86139788 Fax:+45-86131160

Finland
Dr Kimmo Mattila
CSC Center for Scientific Computing PL 405 (Tietotie 6) 02101 Espoo Finland
Email: erja.heikkinen@csc.fi
Tel:+358-9-4572433 Fax:+358-9-4572302

France
Dr Philippe Dessen
Infobiogen 7 rue Guy Moquet - BP8 94801 Villejuif Cedex France
Email: dessen@infobiogen.fr
Tel:+33-1-45595241 Fax:+33-1-45595250

Germany
Dr Martin Ebeling
Department of Molecular Biophysics (0810) German Cancer Research Centre Im Neuenheimer Feld 280 69120 Heidelberg Germany
Email: m.ebeling@dkfz-heidelberg.de
Tel:+49/6221-42-2342 Fax:+49/6221-42-2333

Greece
Dr Babis Savakis
FORTH Insitute of Molecular Biology PO Box 1527 711 10 Heraklion Crete Greece
Email: savakis@nefeli.imbb.forth.gr
Tel:+30-81-212647 Fax:+30-81-231308

Hungary
Dr Endre Barta
Agricultural Biotechnology Centre Szent-Gyorgyi u. 4 PO Box 410 2100 Godollo Hungary
Email: barta@abc.hu
Tel:+36-28-430127 Fax:+36-28-420096

India
Prof MW Pandit
Centre for DNA Fingerprinting (CDFD) CCMB Campus Uppal Road Hyderabad 500 007 India
Email: cdfddbt@hd1.vsnl.net.in
Tel: +91-40-7150008

Ireland
Dr Andrew Lloyd
INCBI Dept Genetics Trinity College Dublin 2 Ireland
Email: atlloyd@tcd.ie
Tel:+353-1-608-1969 Fax:+353-1-679-8558

Israel
Dr Leon Esterman
Biological Computing Division Weizmann Institute of Science Rehovot 76100 Israel
Email: lsestern@weizmann.weizmann.ac.il
Tel:+972-8-9343934 Fax:+972-8-9466269

Italy
Dr Marcella Attimonelli
Area di Ricerca CNR-BARI Via Amendoal 166/5 70126 - Bari Italy
Email: marcella@area.ba.cnr.it
Tel:+39-80-5482130 Fax:+39-80-5484467

Netherlands
Dr Jack Leunissen
Caos/Camm Centre University of Nijmegen Toernooiveld 6525 ED Nijmegen Netherlands
Email: jackl@caos.kun.nl
Tel:+31 24 365 22 48 Fax:+31 24 365 29 77

Norway
Ms Karin Lagesen
Biotechnology Centre of Oslo University of Oslo
Gaustadalleen 21 0317 Oslo Norway
Email: karin.lagesen@biotek.uio.no
Tel:+47-22958756 Fax:+47-22694130

Poland
Dr Piotr Zielenkiewicz
Institute of Biochemistry and Biophysics Polish Academy of Sciences Pawinskiego 5a 02-106 Warsawa Poland
Email: piotr@ibbrain.ibb.waw.pl
Tel:+48-2-6584703 Fax:+48-39-121623

Portugal
Dr Pedro Fernandes
Instituto Gulbenkian de Ciencia Rua da Quinta Grande Apt. 14 2781 Oeiras Codex Portugal
Email: pfern@pen.gulbenkian.pt
Tel:+351-1-443 1408 Fax:+351-1-443 5625

Russia
Professor Sergei Spirin
Belozersky Institute of PhysicoChemical Biology Moscow State University Laboratory Korpus A - Room 612 119899 Vorobyevy Gory - MOSCOW Russia
Email: sas@brodsky.genebee.msu.su
Tel:+7 (095) 932 8825 Fax:+7 (095) 939 3181

South Africa
Dr Win Hide
SANBI Private Bag X17 Bellville 7535 University of the Western Cape South Africa
Email: winhide@techno.sanbi.ac.za
Tel:+27 21 959 3645 Fax:+27 21 959 2512

Spain
Dr JoseRamon Valverde
CNB Universidad Autonoma de Madrid Campus de Canto Blanco 28049 Madrid Spain
Email: jvalverde@cnb.uam.es
Tel:+341-5854543 Fax:+341-5854506

Sweden
Mr Nils-Einar Eriksson
Uppsala Biomedical Centre Box 570 S-721 23 Uppsala Sweden
Email: Nils-Einar.Eriksson@bmc.uu.se
Tel:+46-18-471 40 17 Fax:+46-18-55 17 59

Switzerland
Dr Victor Jongeneel
ISREC Bioinformatics Group Chemin des Boveresses 155 CH-1066 Epalinges Switzerland
Email: victor.jongeneel@isrec.unil.ch
Tel:+41-21-692-5994 Fax:+41-21-653-4474

Turkey
Dr Zehra Sayers
National Bioinformatics Node (NBN) MAM GMBAE NBN POB 21 41470 Gebze-Kocaeli Turkey
Email: zehra@bioinfo.rigeb.gov.tr
Tel:+90-262-6412300 ext 4007 Fax:+90-262-6412309

United Kingdom
Dr Alan Bleasby
SEQNET DRAL Daresbury Laboratory Daresbury Warrington WA4 4AD England
Email: ajb@dl.ac.uk
Tel:+44 1925 603351 Fax:+44 1925 603100

Specialist Nodes

EMBL-EBI
Dr Rodrigo Lopez
EBI Hinxton Hall Hinxton Cambridge CB10 1SD England
Email: rls@ebi.ac.uk
Tel: 1223 494438 Fax:+44 1223 494 468

ETI
Dr Peter Schalk
ETI biodiversity Center Universiteit van Amsterdam Mauritskade 61 1092 AD Amsterdam theNetherlands
Email: pschalk@eti.uva.nl
Tel:+31-20-5257239 Fax:+31-20-5257238

HGMP-RC
Dr Martin Bishop
HGMP Resource Centre Hinxton Cambridge CB10 1SB UK
Email: mbishop@hgmp.mrc.ac.uk
Tel:+44 1223 494500Fax: +44 1223 494512

Hoffman-LaRoche
Dr Daniel Doran
Pharma Preclinical Research Hoffman-LaRoche CH-4002 Basel Switzerland
Email: daniel.doran@roche.com
Tel:+41 61 688 8270 Fax:+41 61 688 1075

ICGEB
Dr Sandor Pongor
ICGEB Padriciano 99 34012 Trieste Italy
Email: pongor@genes.icgeb.trieste.it
Tel:+39 40 3757300 Fax:+39 40 226555

MIPS
Dr Werner Mewes
MIPS Max Planck Institut fur Biochemie Am Klopferspitz 18a D-82152 Martinsried Germany
Email: mewes@mips.biochem.mpg.de
Tel:+49 89 8578 2656 Fax:+49 89 8578 2655

Pharmacia
Dr Staffan Bergh
Pharmacia-Upjohn AB Strandbergsgatan 49 112 87 Stockholm Sweden
Email: staffan.bergh@eu.pnu.se
Tel:+46-8-6959884

Sanger Centre
Mr Peter Rice
The Sanger Centre Wellcome Trust Genome Campus Hinxton Cambridge CB10 1SA England
Email: pmr@sanger.ac.uk
Tel:+44 1223 494967 Fax:+44 1223 494919

UCL-BCM
Dr Terri Attwood
Biomolecular Modelling Unit University College London WC1E 6BT England
Email: attwood@bsm.bioc.ucl.ac.uk
Tel:+44 171 419 3879 Fax:+44 171 380 7193

Dear reader,

If you have any comments or suggestions regarding this newsletter we would be very glad to hear from you. If you have a tip you feel we can print in the Tips from the computer room section, please let us know. Submissions for the BITS section are most welcome, but please remember that we cannot extend space beyond two pages per article. Please send your contributions to one of the editors. You may also submit material by Internet E-mail to:

emb-pub@dl.ac.uk

You are invited to contribute to the LETTERS TO THE EDITOR section.

If you had difficulty getting hold of this newsletter, please let us know. We would be only too happy to add your name to our mailing list. This newsletter is also available on-line using any WWW client via the following URLs:

The Online version, (ISSN 1023-4152) :

- ***http://www.uk.embnet.org/embnet.news/vol6_1/contents.html***
- ***http://www.be.embnet.org/embnet.news/vol6_1/contents.html***
- ***http://www2.ebi.ac.uk/embnet.news/vol6_1/contents.html***
- ***http://www.ie.embnet.org/embnet.news/vol6_1/contents.html***

A Postscript version (ISSN 1023-4144) is available. You can get it by anonymous ftp from:

- ***ftp.uk.embnet.org in the directory pub/embnet.news/***
- ***ftp.be.embnet.org in the directory pub/embnet.news/***
- ***ftp.ebi.ac.uk in the directory pub/embnet.news/***
- ***ftp.ie.embnet.org in the directory pub/embnet.news/***

A pdf version (ISSN 1023-4144) in Acrobat 3 format is also available. You can get it by anonymous ftp from:

- ***ftp.uk.embnet.org in the directory pub/embnet.news/***
- ***ftp.be.embnet.org in the directory pub/embnet.news/***
- ***ftp.ebi.ac.uk in the directory pub/embnet.news/***
- ***ftp.ie.embnet.org in the directory pub/embnet.news/***

Back issues are available at most of these sites.

Publisher:

EMBnet Administration Office.
c/o Jan Noordik
CAOS/CAMM Centre
University of Nijmegen
6525 ED Nijmegen
The Netherlands

embnet.news
Vol.6, No.1, 1998
30 April 1999
ISSN 1023-4144