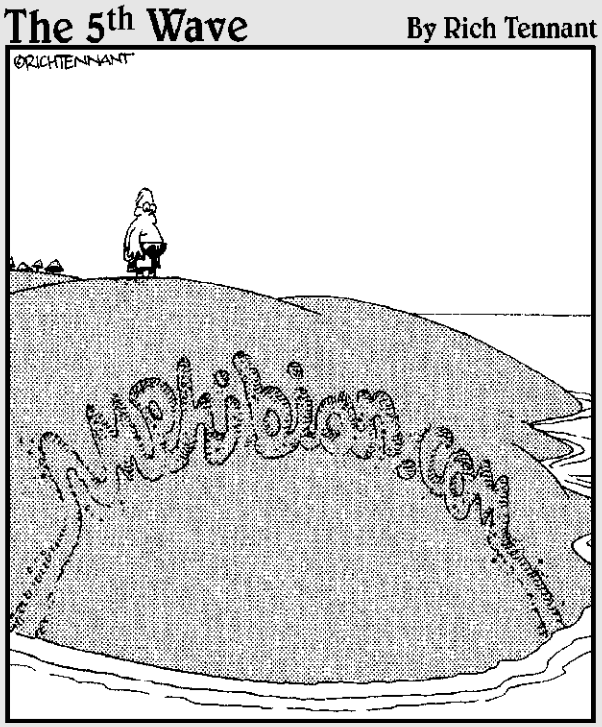
Specialist: Advanced Bioinformatics Techniques



In this part . . .

This part is dedicated to a major slice of bioinformatics: predictions. Accurate predictions are the ultimate dream of biologists. It is a very practical dream because computers are cheap whereas experiments are expensive.

In this part we introduce a Biology Holy Grail: the prediction of protein structures. We also present you with its more successful nucleic acid counterpart: the prediction of RNA structure. We call this sort of stuff *advanced* bioinformatics because the results of these methods only make sense if you know how to interpret them and if you have a good understanding of their limitations. These aren't methods that you can trust blindly, and in this part, we tell you when to keep your eyes peeled!

Working with Protein 3-D Structures

In This Chapter

- Looking at 3-D structures and why they're important
- Predicting the secondary structure of a protein
- Finding the best 3-D model for your sequence
- Creating pictures of 3-D protein structures for your talks and reports
- Analyzing a sequence from a structural point of view
- Displaying interactive (rotating) 3-D molecular models on your own PC

> *There is no foreseeable limit to the number of proteins whose structure is worth analyzing, since each will have its own unique function which demands explanation in structural terms.*
>
> — John C. Kendrew, Nobel Lecture 1962

When you study a protein, you are usually interested in its function. One thing all biologists know these days is that there is a tight relationship between the structure and the function of a protein. As soon as you find an interesting feature at the level of the protein sequence — perhaps a motif or an evolutionarily conserved segment — the right thing to do is to ask what this feature looks like within the 3-D structure.

Typical questions you want to ask are

- Do the amino acids at given sequence positions contribute to the stability of the protein structure?
- Why is this sequence segment conserved (or variable)?

perspective.

From Primary to Secondary Structures

Biologists refer to the sequence of a protein as its *primary structure,* as opposed to the 3-D or *tertiary* structure that is the final shape of the protein. No prize for guessing that an intermediary level of structure exists that is known as the *secondary* structure!

When the first crystallographers started looking at protein structures, they discovered (and predicted) that there was a hierarchy in the way that amino acid sequences fold onto themselves to become a biologically active molecule. Amino acids first look to their immediate neighbors in the sequence to form regions of regular, periodical conformations (backbone shapes). After this is done, the chain collapses further by folding those preshaped regions onto each other (or onto unstructured regions), leading to the final 3-D structure in which residues that are far apart in the sequence come in direct contact with each other.

There are three types of local segments:

- **Helices:** Where residues seem to be following the shape of a spring. The most common are the so-called alpha helices.
- **Extended or Beta-strands:** Where residues are in line and successive residues turn their back to each other.
- **Random coils:** When the amino-acid chain is neither helical nor extended.

Within the latter category, biologists like to distinguish those cases where the chain makes a sharp turn (90° or more) by labeling them *loops.*

Predicting the secondary structure of a protein sequence

Predicting the secondary structure of proteins was one of the hottest goals of the 1990s. It is fair to say that this is one of the great successes of that decade. Nowadays, fairly good servers are available that use Hidden Markov Models

However, bear in mind that this is *only* a prediction; as with all predictions, it can be more or less inaccurate.

To get started using PSIPRED, do the following:

1. **Prepare your favorite protein in a FASTA-formatted .txt file.**

   We used GenPep NP_360043, *Rickettsia conorii* sequence TolB for this example. You can fetch it from the NCBI Web server (`www.ncbi.nlm.nih.gov`); see Step 1 in the later section "Looking at sequence features in 3-D."

2. **Point your browser to** `bioinf.cs.ucl.ac.uk/psipred/`**.**

   This displays the home page of the Protein Structure Prediction server, which is maintained by the Bioinformatics Unit of University College in London (U.K.).

3. **Scroll down the page and click CLICK HERE TO ACCESS THE SERVER.**

   A one-page input form appears.

4. **Copy your sequence and paste it into the Input Sequence window.**

   Both FASTA and plain (with no header) formats are accepted here.

5. **Keep the default pre-selected option in the Choose Prediction Method section.**

6. **Keep the default pre-selected option in the Filtering Options.**

7. **Enter your e-mail address in the Submit Sequence section.**

   You'll also find a field in this section where you can enter a short name for your sequence; this is handy if you need help remembering what the analysis was all about.

8. **Click the Predict button.**

   This server returns its results by e-mail, usually in less than 30 minutes (sometimes more, depending on its load). By default, the server runs PSIPRED, the secondary structure prediction method.

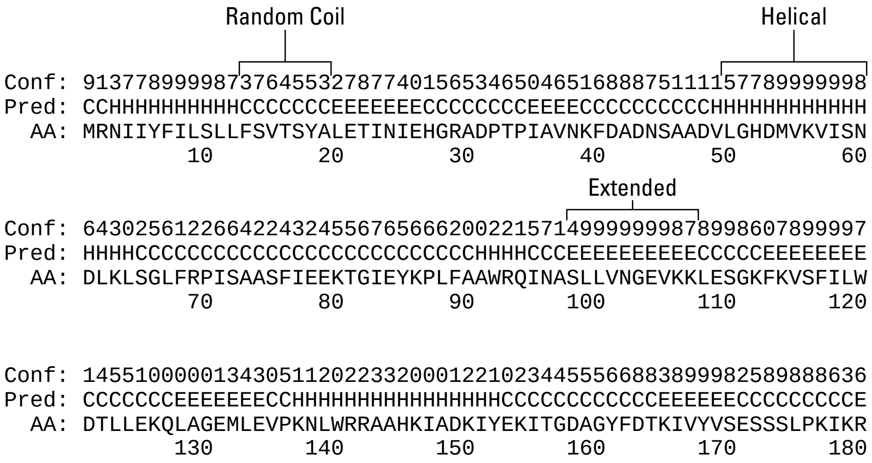
   Alternatively, you can select three other prediction methods: one for transmembrane segments and two for fold recognition. Feel free to explore these if you have time.

If everything is fine, a new page pops up and informs you that your prediction job has been submitted and that the result will be sent to the e-mail address you provided in Step 7 (check to make sure that the e-mail address is correct!).

from MSN's Hotmail or Google's Gmail). No results will be returned.

When you come back, look for an e-mail from `psipred@cs.ucl.ac.uk`, similar to what you see in Figure 11-1. The output is a simple text file in which each line of the sequence (AA) is now printed side by side with two other lines:

- The *prediction* line (Pred), consisting of H, E, and C characters, denoting the predicted conformation for each residue (H = *Helical,* E = *Extended,* and C = *Random Coil*).
- The *confidence* (Conf) line, consisting of digits from 9 to 0, indicating the reliability of the prediction for each position (9 = high, 0 = poor).


Random Coil
Helical
Extended

```
Conf: 913778999987376455327877401565346504651688875111157789999998
Pred: CCHHHHHHHHHHCCCCCCCEEEEEEEECCCCCCCCEEEECCCCCCCCCCHHHHHHHHHHH
  AA: MRNIIYFILSLLFSVTSYALETINIEHGRADPTPIAVNKFDADNSAADVLGHDMVKVISN
              10        20        30        40        50        60

Conf: 643025612266422432455676566620022157149999999878998607899997
Pred: HHHHCCCCCCCCCCCCCCCCCCCCCCCCHHHHCCCEEEEEEEEEEECCCCCEEEEEEEEE
  AA: DLKLSGLFRPISAASFIEEKTGIEYKPLFAAWRQINASLLVNGEVKKLESGKFKVSFILW
              70        80        90       100       110       120

Conf: 145510000013430511202233200012210234455566883899982589888636
Pred: CCCCCCCEEEEEEEECCHHHHHHHHHHHHHHHHHCCCCCCCCCCCEEEEEEECCCCCCCCCE
  AA: DTLLEKQLAGEMLEVPKNLWRRAAHKIADKIYEKITGDAGYFDTKIVYVSESSSLPKIKR
             130       140       150       160       170       180
```

Figure 11-1: Typical PSIPRED e-mail: Partial output for R. conorii TolB (NP_360043).

TIP

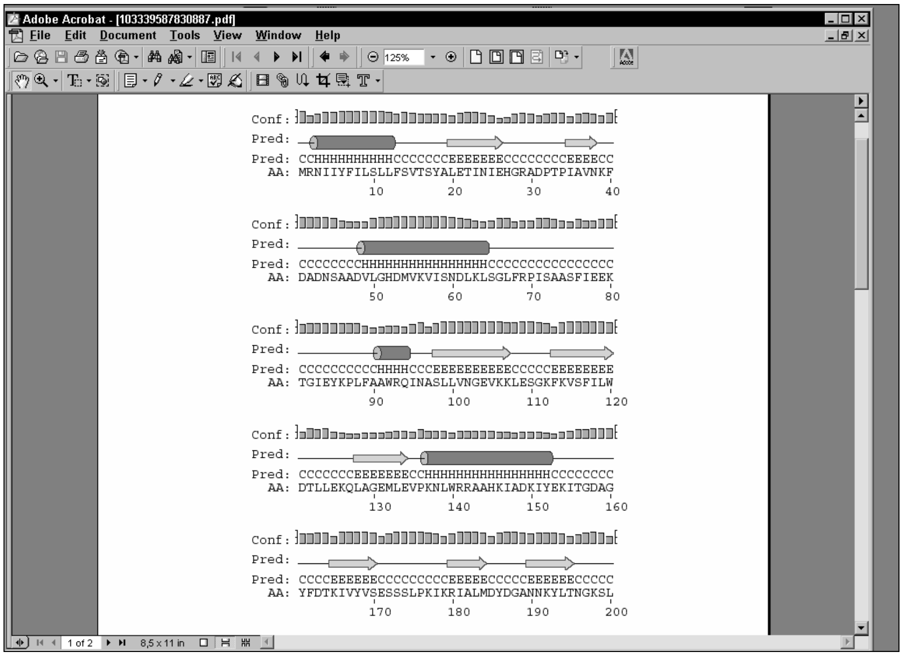
To get the various columns well lined up as in Figure 11-1, copy the e-mailed results into a Word document and use a constant spaced font such as Courier 10.

PSIPRED is very popular because of its accuracy (over 80 percent correctly predicted positions, on average) and also because it produces a nice graphic output that comes in handy for publication. To retrieve the graphic files, look at the very end of your PSIPRED e-mail. It reads like this:

opportunity to download the graphic files in a number of different formats:

- **PostScript File:** `http://bioinf2.cs.ucl.ac.uk/psiout/xxxxxx.ps`
- **PDF File:** `http://bioinf2.cs.ucl.ac.uk/psiout/xxxxxx.pdf`
- **JPEG Page 1:** `http://bioinf2.cs.ucl.ac.uk/psiout/xxxxx_1.jpg`
- **JPEG Page 2:** `http://bioinf2.cs.ucl.ac.uk/psiout/xxxxx_2.jpg`

If your computer is already set up with Adobe Acrobat, select the graphic file in PDF format by simply clicking the appropriate URL. That should open up a display similar to what's shown in Figure 11-2. You may also download the JPEG version if you want to incorporate it into a PowerPoint presentation.



```
Conf:
Pred:
Pred: CCHHHHHHHHHHCCCCCCCEEEEEEECCCCCCCCEEEECC
  AA: MRNIIYFILSLLFSVTSYALETINIEHGRADPTPIAVNKF
              10        20        30        40

Conf:
Pred:
Pred: CCCCCCCCHHHHHHHHHHHHHHHHCCCCCCCCCCCCCCCC
  AA: DADNSAADVLGHDMVKVISNDLKLSGLFRPISAASFIEEK
              50        60        70        80

Conf:
Pred:
Pred: CCCCCCCCCCHHHHCCCEEEEEEEEEECCCCCEEEEEEEE
  AA: TGIEYKPLFAAWRQINASLLVNGEVKKLESGKFKVSFILW
              90       100       110       120

Conf:
Pred:
Pred: CCCCCCCEEEEEEEECCHHHHHHHHHHHHHHHHCCCCCCCC
  AA: DTLLEKQLAGEMLEVPKNLWRRAAHKIADKIYEKITGDAG
             130       140       150       160

Conf:
Pred:
Pred: CCCCEEEEEECCCCCCCCCEEEEECCCCCEEEEEECCCCC
  AA: YFDTKIVYVSESSSLPKIKRIALMDYDGANNKYLTNGKSL
             170       180       190       200
```

Figure 11-2: Typical PSIPRED graphics: Partial output for R. conorii TolB (NP_360043).

Predicting additional structural features

PSIPRED is ideal if you need a quick answer to a basic question, such as *What is the secondary structure of my protein?* If you want to know more, you must turn to other sites capable of predicting a much broader variety of features in your protein.

The PredictProtein server is probably the most comprehensive site for protein structure analysis. Unfortunately, the original site (based at Columbia University) is also very busy — and response times can be more than a day or two. Luckily, the PredictProtein server has numerous mirror sites throughout the world — including Europe, the United States, Asia, and Australia (but beware: some of the URLs listed on the server's home page are obsolete). Here is a list of up-to-date URLs:

Europe:	`www.predictprotein.org/`
	`www.cmbi.ru.nl/bioinf/predictprotein/`
USA:	`cubic.bioc.columbia.edu/predictprotein/`
	`www.sdsc.edu/predictprotein/`
Asia:	`www.cbi.pku.edu.cn/predictprotein/`

Your basic PredictProtein server

To check out a PredictProtein server, have your protein sequence handy in another browser window or in a local file, and follow these steps:

1. **Point your browser to** `www.sdsc.edu/predictprotein/`**.**

   The PredictProtein Server page dutifully appears.

   We found (for now!) that the San Diego site tends to be the more responsive.

2. **Click the** Submit a Protein Sequence for Prediction **link (the first bullet).**

   A default submission form appears.

3. **Enter your e-mail address in the top text field.**

4. **Select the output format by checking a box.**

   In the default mode (Results on Our Web Site), the server sends you an e-mail that indicates the URL of your result file on the PredictProtein

So, in all cases, the server sends you an e-mail; with an address where to find your results or with the results themselves.

5. **Copy your sequence from another browser window (or your local file) and paste it into the Paste or Type Your Sequence box.**

   The server only accepts a single protein sequence, in one-letter code, in the raw (plain) format. No FASTA header.

6. **If you like, you can enter a short name for your sequence in the One-Line Name of Protein text field.**
7. **Click the Submit/Run Prediction button.**

Now all you have to do is check your e-mail from time to time. To estimate how long that will take, click the yellow Wait button at the bottom-left of the form to display the status of the PredictProtein queue.

A default analysis returns the following:

- A secondary structure prediction on the three conformational states (H = *Helical,* E = *Extended,* and C = *Random Coil*)
- A prediction of the solvent accessibility of the various residues
- A prediction of transmembrane helices and their topologies
- A prediction of globular regions in your protein
- A prediction of the coiled/coil regions of your protein
- A description of the PROSITE motifs matching your sequence (for more on PROSITE motifs, see Chapter 6)
- A description of the putative domain structure for your protein (Prodom domains)
- A list of likely homologues of your query sequence found in the Swiss-Prot protein database using the similarity-search program PSI-BLAST
- A multiple alignment of your query with these homologous sequences generated by the MaxHom program
- A prediction of bound cysteines (disulphide bonds) in your sequence
- A description of the composition-biased regions in your sequence

These various features are reported only if they occur in your query sequence. If you need the details on any of these predictions, you can run additional programs by using the Advanced or Expert input form.

1. **Point your browser to** `www.sdsc.edu/predictprotein/`**.**

   You’re back at the UC-San Diego PredictProtein Server page.

2. **Click the** Submit Requests to META-PP **link (the second bullet).**

   This displays a one-page input form.

3. **Enter your e-mail address in the top text field.**

   This server returns its results by e-mail.

4. **Copy your sequence from another browser window (or from a local file) and paste it into the Paste or Type Your Sequence box.**

   A one-letter code sequence given in the raw (plain) format is fine.

5. **Scroll down to find the table of options available and choose the methods that you’re interested in.**

   Most of these services are alternative methods for doing secondary-structure prediction or threading. The threading methods try to fit your sequence into a known 3-D structure. (We describe threading methods later in this chapter.)

6. **Click the Submit/Run Prediction button.**

TIP

Although it may seem convenient to submit all these analyses at once, our experience is that it is quicker and less confusing to go to the relevant sites and use them without the META intermediary.

Finding the best server around

TIP

There are many servers around the world and, as is to be expected, many use different kind of algorithms — which usually means that some perform faster than others. Of course, their authors keep updating them. If you want to know which one is currently the best server, you can check out EVA, the secondary-structure server monitoring system, at `cubic.bioc.columbia.edu/eva/`. (EVA is short for *EV*aluation of *A*utomatic protein structure prediction.)

From the Primary Structure to the 3-D Structure

In the previous section, we show you how to get some preliminary information about what your protein sequence may look like when it’s folded into a that secondary structure predictions are far less useful than we would like them to be. They fall short of being the real thing: the detailed spatial representation of your molecule. In this section, we show you how to gather and use 3-D information to better understand what goes on in your protein sequence.

The good news for biologists is that plenty of experimental 3-D structure information is available on the Internet. As was the case when the molecular biologists of the world agreed to centralize their sequence data into the GenBank/EMBL/DDBJ databank repository, all structural biologists have agreed to deposit their 3-D structure coordinates into a single database: the Protein Data Bank. Everybody refers to this database by its acronym: PDB.

Retrieving and displaying a 3-D structure from a PDB site

Like other data repositories, the Protein Data Bank (PDB) offers a rather daunting interface that wasn't particularly designed with the nonspecialist in mind. Yet, in those rare cases where you know precisely what you're looking for — and even know what you're doing! — you may want to retrieve a protein 3-D structure dataset directly from one of the PDB sites. Before you query the PDB, be sure to collect some precise information about the structure you're looking for — such as the exact protein name or (even better) its PDB identifier. You can usually obtain this identifier from such user-friendly sources as the ExPASy/Swiss-Prot server or by using the various NCBI query tools. (See Chapter 4 for more on how to use these tools.)

Here's how to obtain and display a PDB structure. For now, let's assume that we are looking for the structure of an *Escherichia coli* (*E. coli*) protein named TolB, with PDB ID code 1CRZ.

1. **Point your browser to** `www.rcsb.org/pdb/`**.**

   This takes you to the PDB home page.

2. **Enter the PDB ID code** 1CRZ **in the search box in the middle at the very top of the page.**

3. **Click the adjacent SEARCH button.**

   Figure 11-3 shows the resulting output. This one-page Structure Summary form presents the essential information on this protein structure — including a small graphic, a bibliographical reference, its function, its

4. **Click the** All Images **link in the Display Options below the structure image (refer to Figure 11-3).**

   A new page appears, with an enlarged still view of the structure on the top, and an interactive view (a Java applet) at the bottom.

5. **Right-click the Still Image (in color).**

   A menu of options now allows you to either print the image, copy it, or save it to your own hard drive. It is now suitable for inclusion in reports or presentations. (Including such a figure is a great way to impress your boss or your professor!)

   But wait, there's more! You can actually get a better feeling for the shape of this protein using the bottom gray-scale Interactive View as follows:

6. **Right-click the Interactive View picture.**

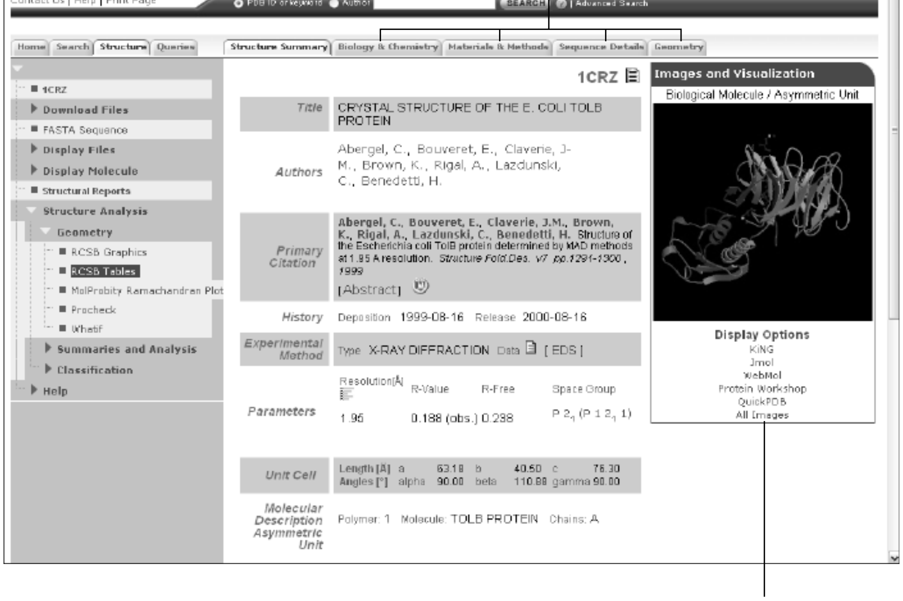
   You are offered a large menu of viewing options. Explore them to see which functions of this Java applet actually work on your specific system. For instance, selecting Spin On puts the molecule in an endless rotation. (To stop the merry-go-round, select Spin Off.) This is a great way to gain a better understanding of the 3D shape of the molecule.

7. **Left-click and drag from any position in the structure, as shown in Figure 11-4.**

   This sets the molecule in controlled motion. Use this mode to inspect specific structural regions at your own pace, from any angle that you choose. (You can zoom in as well.) In addition, double-clicking allows you to select a given position from which you can measure distances and angles. Try it out for yourself to see what you can do; it's fun, and you may impress your colleagues with your brand new expertise in structural biology.

Feel free to play with the other display options (KiNG, WebMol, . . ., etc.) and see which one works best for you. See the section "Exploring the sequence/PDB structure relationship the interactive way," later in this chapter, for a tutorial on a more advanced tool that requires downloading and installing the CnD3 free software on your machine.

The preceding steps list shows you how to jump directly from the PDB entry to a visual representation of the molecule. This may not be enough, and you may also want to keep the PDB entry for further study on your own computer. Read on to find out how.



1CRZ

Title	CRYSTAL STRUCTURE OF THE E. COLI TOLB PROTEIN
Authors	Abergel, C., Bouveret, E., Claverie, J-M., Brown, K., Rigal, A., Lazdunski, C., Benedetti, H.
Primary Citation	**Abergel, C., Bouveret, E., Claverie, J.M., Brown, K., Rigal, A., Lazdunski, C., Benedetti, H.** Structure of the Escherichia coli TolB protein determined by MAD methods at 1.95 A resolution. *Structure Fold.Des. v7 pp.1291-1300 , 1999* [Abstract]
History	Deposition 1999-08-16 Release 2000-08-16
Experimental Method	Type X-RAY DIFFRACTION Data [EDS]

Parameters	Resolution[Å]	R-Value	R-Free	Space Group
	1.95	0.188 (obs.)	0.238	P 2_1 (P 1 2_1 1)

Unit Cell						
Length [Å]	a	63.18	b	40.50	c	76.30
Angles [°]	alpha	90.00	beta	110.88	gamma	90.00

Molecular Description Asymmetric Unit: Polymer: 1 Molecule: TOLB PROTEIN Chains: A

Click here for an interactive view.

Figure 11-3: The PDB structure summary output form for query 1CRZ.

WARNING!

PDB files aren't meant to be read by nonspecialists. They are rather long because they contain a vast amount of data. For instance, the 1CRZ contains the detailed x, y, z coordinates of 3,488 atoms — as well as additional information about their connectivity!

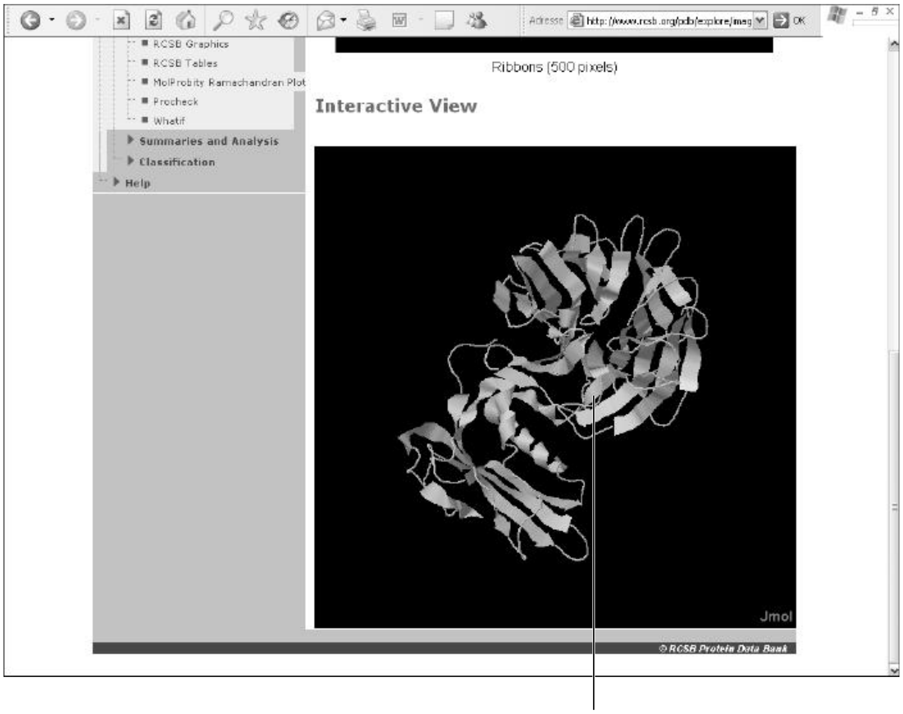
If you're sure that you need to download a PDB file, you can simply stop after Step 3 in the previous step list — and look for the various links provided in the leftmost blue column of the screen. (Refer to Figure 11-3.) The possibilities are

- Download Files (to save them to your own hard drive)
- Get the FASTA Sequence
- Display Files (to look at their internal format)
- Display Molecule
- Structural Reports

ous tags at the top of the form. For instance, the Sequence Details link gives you a nice graphical picture of the actual secondary structure along with the sequence. You can compare it to the prediction that we made earlier in this chapter.

Guessing the 3-D structure of your protein

The previous section shows how to retrieve and quickly display a known 3-D–structure recorded in the PDB. This is definitely useful but falls short of addressing the questions we listed at the beginning of this chapter. Basically, we still do not know what the interesting bits of our own sequence look like in their 3-D context.



Click here to drag and rotate the molecule.

Figure 11-4: Interactive view of the 1CRZ PDB entry.

it fold? A simple way to answer this question is to look for a homologous protein with a known 3-D structure in the PDB. The next steps list shows you how.

Imagine, if you will, that we've just determined the sequence of the TolB gene of the bacteria *Rickettsia conorii* and we are curious about its structure. To satisfy our curiosity, we do the following:

1. **Fetch the protein sequence from NCBI:**

   a. Open a window in your browser and go to `www.ncbi.nlm.nih.gov`.

   b. Choose Protein from the Search drop-down menu.

   c. Type the identifier **NP_360043** in the query window and then click Go.

   d. When the answer comes back, change the Display format to FASTA.

   The *Rickettsia conorii* TolB protein sequence is now ready for use.

2. **Open a new browser window and point it to the NCBI BLAST server at** `www.ncbi.nlm.nih.gov/BLAST/`**.**

3. **Click the** Standard Protein-Protein BLAST [blastp] **link.**

   It is the first choice in the upper-right corner. The blastp input page appears.

4. **Select** pdb **from the Choose database drop-down menu.**

5. **Copy the *Rickettsia conorii* TolB sequence from the other browser window and paste it into the blastp query window.**

6. **Deselect the Do CD-search box (to simplify your output).**

7. **Click the BLAST! button.**

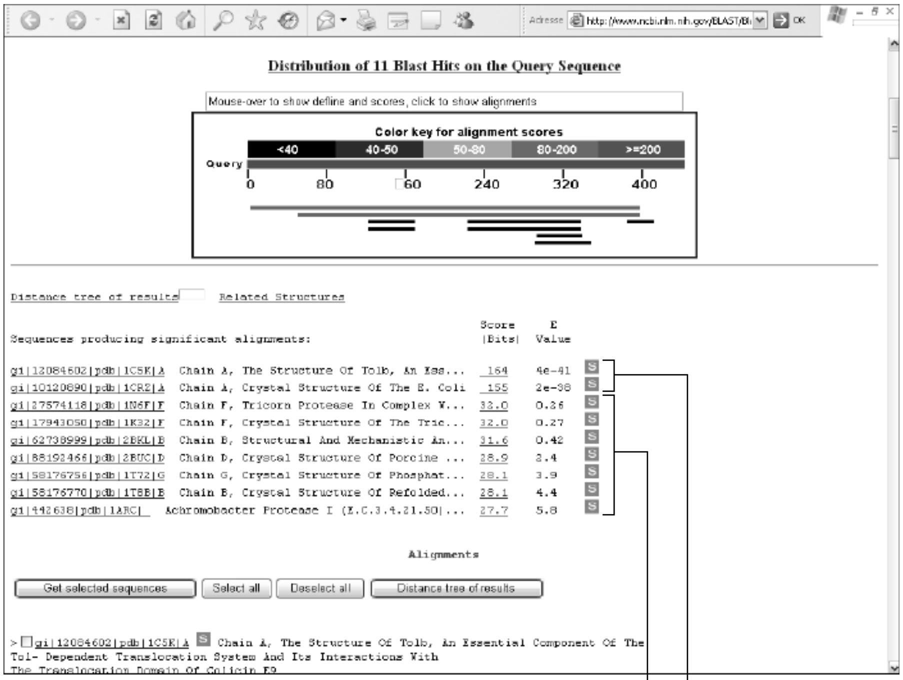
   An intermediate page appears, telling you that your request has been successfully submitted and put into the Blast Queue.

8. **Click the Format! button on the intermediate page and wait for the Blastp search to finish.**

Figure 11-5 shows an example of the kinds of results you can expect. There are two strong matches with E-values much lower than 1. They are identified as

1. The homologous PDB sequences have 29 percent identical residues with our query.
2. The homology region covers the entire sequences.

By experience, we know that, given such a percentage of identical residues, the 3-D structure of our query protein (the *Rickettsia* TolB) is probably quasi-identical to the 3-D structure of its *E. coli* homologue. And so, we have a nearly perfect 3-D model to work with. The next section shows you how to use this model to further interpret the TolB protein sequence.


Adresse http://www.ncbi.nlm.nih.gov/BLAST/Bl
OK
Distribution of 11 Blast Hits on the Query Sequence
Mouse-over to show define and scores, click to show alignments
Color key for alignment scores
<40
40-50
50-80
80-200
>=200
Query
0
80
60
240
320
400
Get selected sequences
Select all
Deselect all
Distance tree of results

```
Distance tree of results   Related Structures

                                                                 Score    E
Sequences producing significant alignments:                      (Bits)  Value

gi|12084602|pdb|1C5K|A  Chain A, The Structure Of Tolb, An Ess...  164   4e-41
gi|10120890|pdb|1CR2|A  Chain A, Crystal Structure Of The E. Coli  155   2e-38
gi|27574118|pdb|1N6F|F  Chain F, Tricorn Protease In Complex W...  32.0  0.26
gi|17943050|pdb|1K32|F  Chain F, Crystal Structure Of The Tric...  32.0  0.27
gi|62738999|pdb|2BKL|B  Chain B, Structural And Mechanistic An...  31.6  0.42
gi|88192466|pdb|2BUC|D  Chain D, Crystal Structure Of Porcine ...  28.9  2.4
gi|58176756|pdb|1T72|G  Chain G, Crystal Structure Of Phosphat...  28.1  3.9
gi|58176770|pdb|1T8B|B  Chain B, Crystal Structure Of Refolded...  28.1  4.4
gi|442638|pdb|1ARC|   Achromobacter Protease I (E.C.3.4.21.50|...  27.7  5.8

                                Alignments

>gi|12084602|pdb|1C5K|A Chain A, The Structure Of Tolb, An Essential Component Of The
Tol- Dependent Translocation System And Its Interactions With
The Translocation Domain Of Colicin E9
```

Non-significant matches

Two very good matches in PDB

Figure 11-5: Blastp output of the Rickettsia conorii TolB sequence against PDB.

ments to identify the most significant regions of a sequence: Conserved residues (or, alternatively, highly variable ones) are often key to predicting or understanding a protein function. Going further, by precisely locating these conserved residues in space, it's possible to come up with additional clues as to their biological roles: Such residues, for example, can delineate a cavity at the active site of an enzyme — or, if they are found at the surface, one can assume they are good candidates for interactivity with other molecules, and so on.

We can use the example of the TolB protein family (of relatively unknown function) to illustrate the interplay between multiple alignments and structural analysis. Here's how it's done:

1. **First fetch several TolB homologue sequences from various bacterial species from NCBI:**

   a. Open a window in your browser and go to `www.ncbi.nlm.nih.gov`.

   b. Choose Protein from the drop-down Search menu.

   c. In the query window, type in the following identifiers: **NP_360043 (*R. conorii*)**, **NP_415268 (*E. coli*)**, **NP_404737 (*Y. pestis*)**, **NP_249663 (*P. aeruginosa*)**, and **NP_438543 (*H. influenzae*)**; then click Go.

   d. When the answer is returned, change the Display format to FASTA.

   e. Finally, use Send to Text to get rid of any parasitic characters.

   Five TolB protein sequences are now ready for use. Now we can build a multiple alignment out of these sequences.

2. **Open a new browser window and go to** `www.igs.cnrs-mrs.fr/Tcoffee/`**.**

3. **Click on Regular in the very top TCOFFEE option line.**

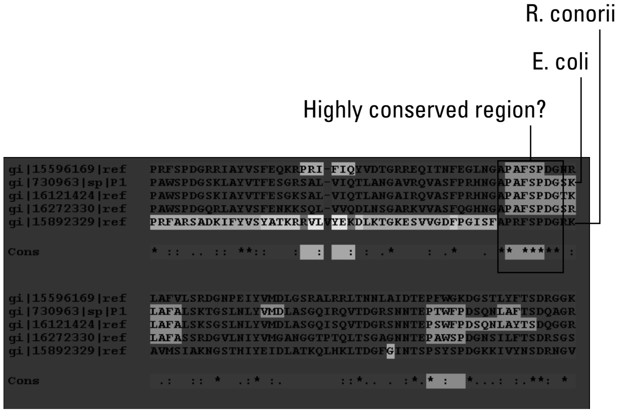
4. **Copy the five TolB sequences from the other browser window and paste them into the Tcoffee input window.**

   Be sure to include their FASTA headers.

5. **Click Submit (at the bottom of the form).**

   The Tcoffee Results form automatically appears after an intermediary waiting page.

is to figure out where in the TolB structure these residues are located.



```
gi|15596169|ref    PRFSPDGRRIAYVSFEQKRPRI-FIQYVDTGRREQITNFEGLNGAPAFSPDGNR
gi|730963|sp|P1    PAWSPDGSKLAYVTFESGRSAL-VIQTLANGAVRQVASFPRHNGAPAFSPDGSK
gi|16121424|ref    PAWSPDGSKIAYVTFESGKSAL-VIQTLANGAIRQVASFPRHNGAPAFSPDGTK
gi|16272330|ref    PAWSPDGQRLAYVSFENKKSQL-VVQDLNSGARKVVASFQGHNGAPAFSPDGSR
gi|15892329|ref    PRFARSADKIFYVSYATKRRVLVYEKDLKTGKESVVGDFPGISFAPRFSPDGRK

Cons               * :: .. :: **::   :  :   : : .*    : .*   . ** ***** :


gi|15596169|ref    LAFVLSRDGNPEIYVMDLGSRALRRLTNNLAIDTEPFWGKDGSTLYFTSDRGGK
gi|730963|sp|P1    LAFALSKTGSLNLYVMDLASGQIRQVTDGRSNNTEPTWFPDSQNLAFTSDQAGR
gi|16121424|ref    LAFALSKSGSLNLYVMDLASGQISQVTDGRSNNTEPSWFPDSQNLAYTSDQGGR
gi|16272330|ref    LAFASSRDGVLNIYVMGANGGTPTQLTSGAGNNTEPAWSPDGNSILFTSDRSGS
gi|15892329|ref    AVMSIAKNGSTHIYEIDLATKQLHKLTDGFGINTSPSYSPDGKKIVYNSDRNGV

Cons                .:  :: *  .:* :.       ::*.. . :*.* :  *...: :.**: *
```

Figure 11-6: An intriguing segment of strictly conserved residues in the TolB proteins.

Exploring the sequence/PDB structure relationship the interactive way

To interpret the significance of an invariant segment in our TolB proteins, we must be able to display its 3-D structure and *turn this structure around interactively* so we can look at it *from any angle.* Simultaneously, we must be able to refer to the protein sequence, and to *highlight* the residues we find interesting.

In short, we need a protein model that we can rotate around and analyze in parallel with its sequence. Not long ago, this kind of business was still the privilege of protein crystallographers working on expensive computers. Not anymore! If you can bear with us until the end of this chapter, you'll be doing it on your own PC in just a few minutes. Just follow our instructions:

1. **Point your browser to** `www.ncbi.nlm.nih.gov/Structure/`**.**

   The structure server of the NCBI appears.

2. **In the Search Entrez Structure/MMDB window at the top of the page, enter the PDB code of your relevant model. (MMDB stands for "*M*olecular *M*odeling *D*ata*B*ase.")**

   For our example, we'll use 1CRZ.

   If you have the choice of several structures, a simple rule is to choose the one with the best resolution; for our example, the best resolution is 1CRZ (that is, 1.95 Angstroms). You'll find the Angstrom value in the ID card of each PDB entry (refer to Figure 11-3).

shown in Figure 11-7.

This MMDB form exhibits a prominent View 3-D Structure button. What will happen next depends on the software already installed on your computer.

5. **Click the View 3-D Structure button.**

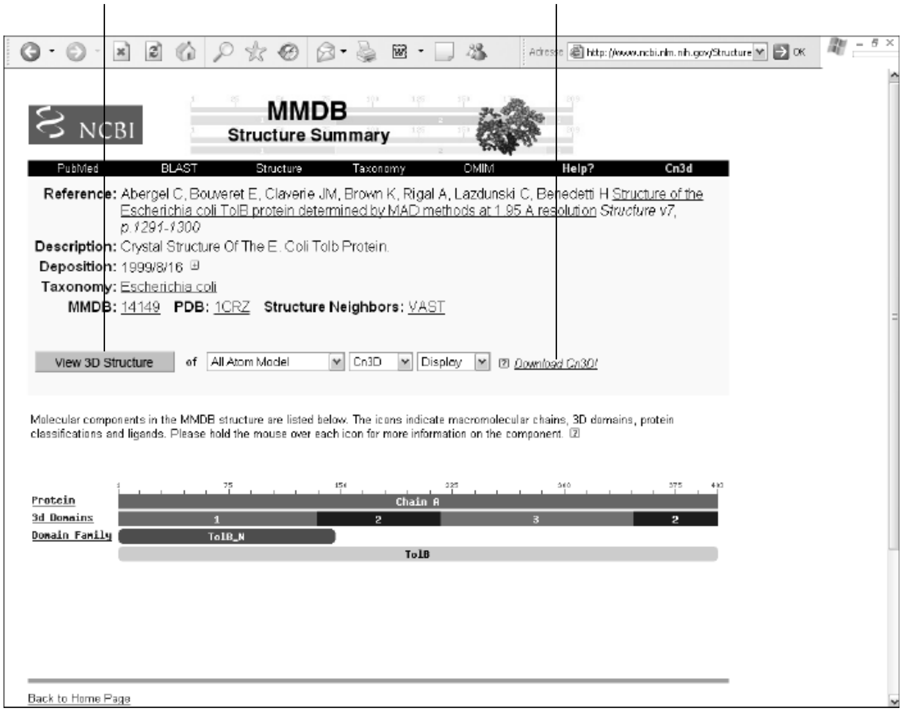
   Doing so downloads a file with the atomic coordinates of the 1CRZ protein structure. This file is formatted for the 3-D structure viewer Cn3D program. If nobody ever used your computer to display a protein structure in 3-D, a dialog box will pop up, asking you what to do with this file. It's telling you that your system doesn't contain the Cn3D program required to display the structure. In order to continue, you have to install the program. Here's how:

   a. Click the Download Cn3D! link.

      This takes you to the Cn3D home page, as shown in Figure 11-8. From here you can access a complete tutorial and/or continue the installation process.

Click here to view.

Click here to install Cn3D.



Reference: Abergel C, Bouveret E, Claverie JM, Brown K, Rigal A, Lazdunski C, Benedetti H Structure of the Escherichia coli TolB protein determined by MAD methods at 1.95 A resolution *Structure v7, p.1291-1300*

Description: Crystal Structure Of The E. Coli Tolb Protein.

Deposition: 1999/8/16

Taxonomy: Escherichia coli

MMDB: 14149 **PDB:** 1CRZ **Structure Neighbors:** VAST

Download Cn3D!

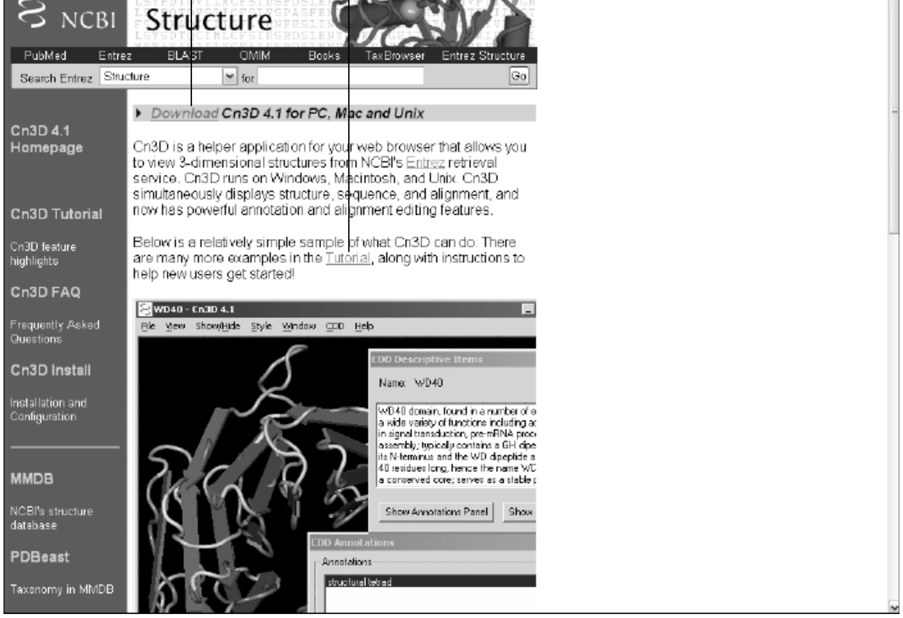
Molecular components in the MMDB structure are listed below. The icons indicate macromolecular chains, 3D domains, protein classifications and ligands. Please hold the mouse over each icon for more information on the component.

Protein

3d Domains

Domain Family

Figure 11-7: MMDB structure card for PDB entry 1CRZ.



Download Cn3D 4.1 for PC, Mac and Unix

Cn3D is a helper application for your web browser that allows you to view 3-dimensional structures from NCBI's Entrez retrieval service. Cn3D runs on Windows, Macintosh, and Unix. Cn3D simultaneously displays structure, sequence, and alignment, and now has powerful annotation and alignment editing features.

Below is a relatively simple sample of what Cn3D can do. There are many more examples in the Tutorial, along with instructions to help new users get started!

WD40 domain, found in a number of e
a wide variety of functions including ac
in signal transduction, pre-mRNA proc
assembly; typically contains a GH dipe
its N-terminus and the WD dipeptide a
40 residues long, hence the name WD
a conserved core; serves as a stable p

Figure 11-8: The Cn3D home page.

b. Click Download Cn3D for PC, Mac, or Unix link.

c. In the window that appears (see Figure 11-9), select the version suitable for your computer: PC, Mac, or Unix.

Whatever platform you select, a new page appears with an FTP address, similar to the Windows page shown in Figure 11-10.

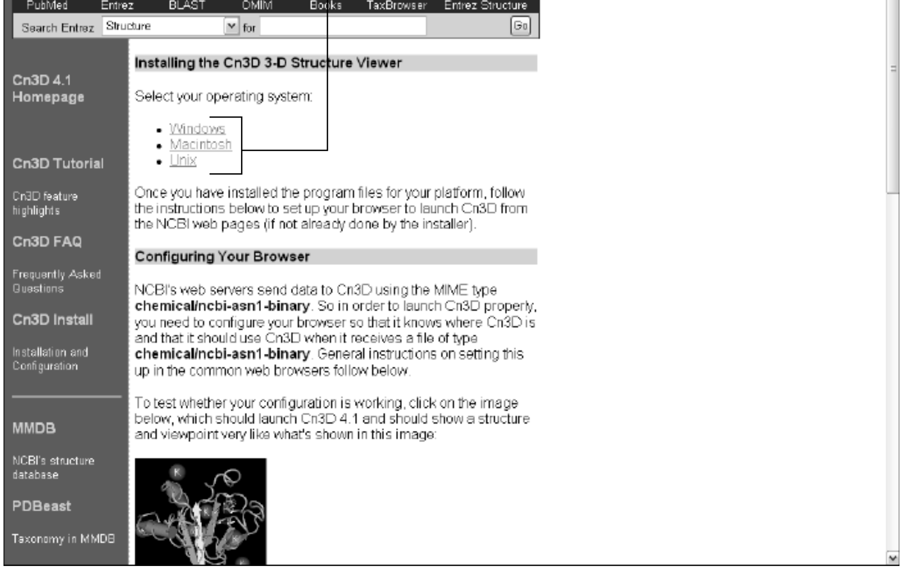
d. Click the FTP address link.

For our Windows system, the link is

```
ftp://ftp.ncbi.nih.gov/cn3d/Cn3D-4.1.msi
```

This downloads the CnD3 installer, which takes just a few seconds.

The system may also require you to install a newer version of the Windows Installer. Just follow the instructions. This also only takes a few extra seconds.



Installing the Cn3D 3-D Structure Viewer

Select your operating system:

- Windows
- Macintosh
- Unix

Once you have installed the program files for your platform, follow the instructions below to set up your browser to launch Cn3D from the NCBI web pages (if not already done by the installer).

Configuring Your Browser

NCBI's web servers send data to Cn3D using the MIME type **chemical/ncbi-asn1-binary**. So in order to launch Cn3D properly, you need to configure your browser so that it knows where Cn3D is and that it should use Cn3D when it receives a file of type **chemical/ncbi-asn1-binary**. General instructions on setting this up in the common web browsers follow below.

To test whether your configuration is working, click on the image below, which should launch Cn3D 4.1 and should show a structure and viewpoint very like what's shown in this image:

Figure 11-9: Installing Cn3D: System selection.

e. After everything is done, locate the Cn3D-4.1.msi icon on your desktop and click it twice.

The Cn3D installation wizard welcomes you.

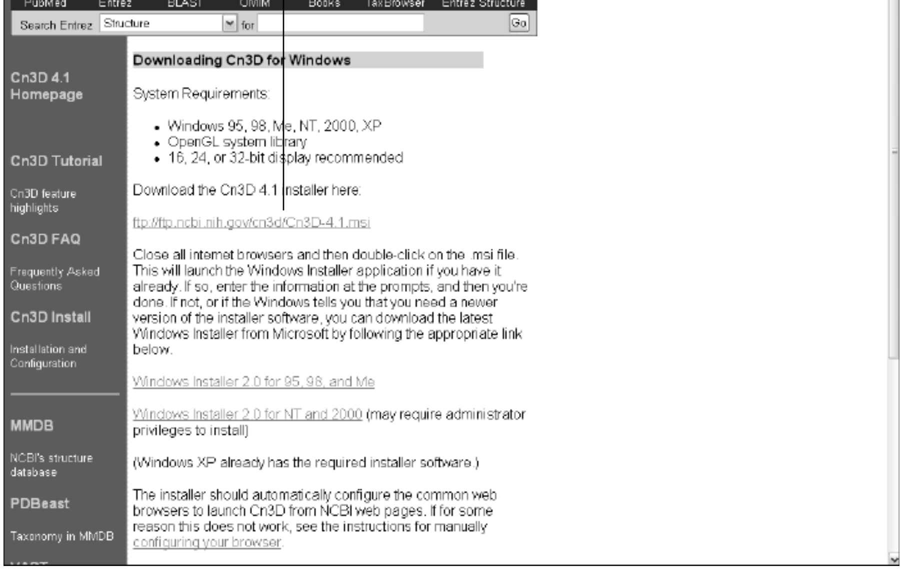
f. Follow its instructions and fill out the forms.

You have to provide a name, an organization, and select a destination folder (default is fine). After the last step, the wizard confirms the successful installation.

Now that Cn3D is in your system, go back to NCBI and redo the sequence of steps that we describe in the previous section.

6. **Click the View 3D Structure button again.**

Now that Cn3d is installed, the miracle occurs, and your screen should look like Figure 11-11. At the top of the MMDB page, the 3-D protein structure that corresponds to the 1CRZ PDB entry is displayed in a live window.



Downloading Cn3D for Windows

System Requirements:

- Windows 95, 98, Me, NT, 2000, XP
- OpenGL system library
- 16, 24, or 32-bit display recommended

Download the Cn3D 4.1 installer here:

ftp://ftp.ncbi.nih.gov/cn3d/Cn3D-4.1.msi

Close all internet browsers and then double-click on the .msi file. This will launch the Windows Installer application if you have it already. If so, enter the information at the prompts, and then you're done. If not, or if the Windows tells you that you need a newer version of the installer software, you can download the latest Windows Installer from Microsoft by following the appropriate link below.

Windows Installer 2.0 for 95, 98, and Me

Windows Installer 2.0 for NT and 2000 (may require administrator privileges to install)

(Windows XP already has the required installer software.)

The installer should automatically configure the common web browsers to launch Cn3D from NCBI web pages. If for some reason this does not work, see the instructions for manually configuring your browser.

Figure 11-10: Installing Cn3D: Installing the installer.

7. **Spin your protein around.**

   You can turn your protein around simply by moving your mouse; in fact, you can spin that protein around like a partner when you're dancing salsa! This is impressive, fun, and tremendously useful!

8. **Prepare your protein for taking a picture.**

   Zoom in and out with the drop-down menus on the top bar of the window.

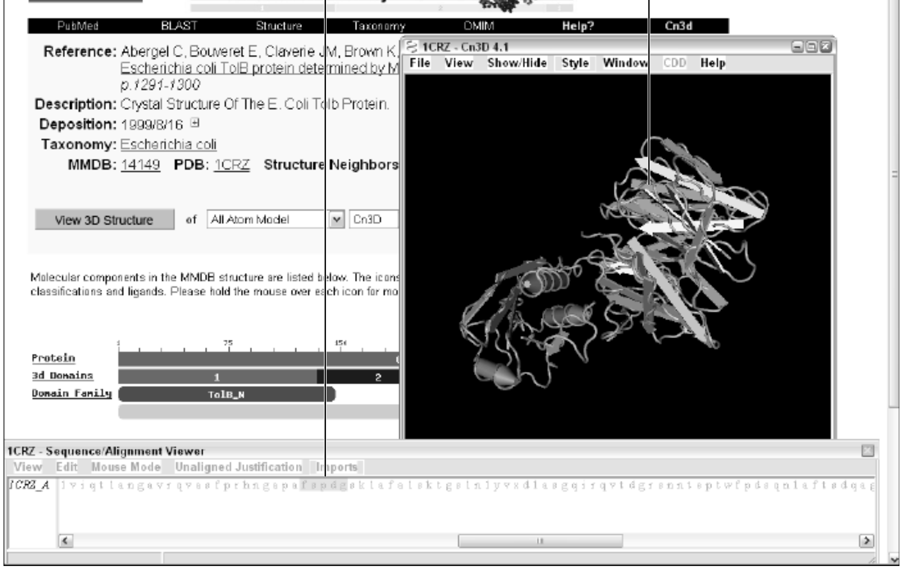
   Change the appearance of the model (wire, tube, ball and stick, or even solid space filling).

9. **Save a picture by choosing File⇨Export PNG from the main menu in the Cn3D window.**

   This allows you to save a picture of the protein structure after choosing its appearance and orientation according to your personal preferences.

REMEMBER

This spectacular show runs locally on your computer, independently of your Internet connection. Terminate your browser, and the live molecule stays on your screen! Neat! It also means that your data doesn't travel over the Internet. (This is important if you work for a paranoid boss.)



Reference: Abergel C, Bouveret E, Claverie JM, Brown K, Escherichia coli TolB protein determined by M p.1291-1300

Description: Crystal Structure Of The E. Coli Tolb Protein.

Deposition: 1999/8/16

Taxonomy: Escherichia coli

MMDB: 14149 **PDB:** 1CRZ **Structure Neighbors**

Molecular components in the MMDB structure are listed below. The icons classifications and ligands. Please hold the mouse over each icon for mo

1CRZ_A l v i q t l a n g a v r q v e s f p r h n g e p a f s p d g s k l a f a l s k t g s l n l y v x d l a s g q i r q v t d g r s n n t e p t w f p d s q n l a f t s d q a g

Figure 11-11: Interactive 3-D display of the E. coli TolB protein sequence and structure.

Analyzing the relationship between sequence and structure interactively

Last but not least, you probably noticed another window displayed below the live molecule. (Refer to Figure 11-11.) This is the sequence viewer. By default, this window displays the protein sequence associated with the PDB entry. Here we have the sequence of the *E. coli* TolB molecule.

The time has come to find out where in the structure that curious region of invariant residues that read “FSPDG” is located. With Cn3D, it could not be easier.

1. **In the sequence viewer top bar, click the View button.**
2. **From the pop-up menu that appears, choose the Find Pattern option.**
3. **In the pattern input box that pops up, type** FSPDG**.**

The corresponding segment is immediately highlighted in bright yellow in the sequence AND in the structure. To locate it in the sequence, scroll along the sequence by using the bottom cursor. To locate it in the structure, turn the molecule around.

being the most variable parts of proteins. (They are freer to mutate.) The strict conservation of these residues suggests that they may have a specific role in the protein function, such as participating in an interaction with another protein or a small molecule.

Beyond This Chapter

In this chapter, we've provided you with a very general idea of what biologists do when they analyze the structure of their proteins with their computers. The applications shown in this chapter probably cover 90 percent of what nonspecialists do when they study the relationships between their sequences and its potential structure.

However, we haven't even mentioned the many other tools and sophisticated approaches that exist for studying protein structures. So here we give you a quick list of what we have *not* talked about, with some related URLs if you're curious about delving deeper into 3-D structures.

Finding proteins with similar shapes

Following the determination of the 3-D structure of your preferred protein, you'll want to know if its shape is unique (what the specialists refer to as a "new fold") or if there are many like it. Just prepare a PDB file with the atomic coordinates of your protein and submit it to NCBI's structure-structure similarity search service (VAST) at `www.ncbi.nlm.nih.gov/Structure/VAST/vastsearch.html`.

Finding other PDB viewers

There are many ways to spin molecules around; you should be aware that, apart from Cn3D, other very powerful structure viewers are available over the Internet, such as

- **RasMol:** `www.rasmol.org`
- **DeepView, Swiss-PdbViewer:** `swissmodel.expasy.org/spdbv/`

teins, as well as the threading problem (see the upcoming section "Threading sequences onto PDB structures"). The three main structure classification sites are

- **The CATH classification:** `www.cathdb.info`
- **The Dali 3D neighbor finding server:** `www.ebi.ac.uk/dali/`
- **The SCOP classification:** `scop.mrc-lmb.cam.ac.uk/scop/`

Doing homology modeling

The detailed modeling of the 3-D structure of a protein with significant sequence similarity to a homologous protein of known structure can go much beyond the simple example we have presented here. Specialists use energy minimization techniques to replace the residues in the original model and refine their positions. A couple of good entry points to find out more about this homology modeling approach are

- **Modeller:** `salilab.org/modeller/modeller.html`
- **SWISS-MODEL:** `swissmodel.expasy.org`

Folding proteins in a computer

Some groups are making constant progress in their attempts to generate a structural model for a protein solely from its sequence by simulating its folding process. This is referred to as the *ab initio* (or *in silico*) folding approach. The successes are still limited to a certain type of small protein sequences. Find out more about it at the Folding@home project Web site (`folding.stanford.edu/`).

Threading sequences onto PDB structures

Other groups have taken up the inverse problem of finding out which of the known structures might be best suited as a folding mold to a given protein sequence. This research field is known as *threading.* In the context of a growing database of structures, threading methods might perform increasingly

- **UCLA Fold recognition:** www.doe-mbi.ucla.edu/Services/FOLD/
- **The PROSPECT server:** compbio.ornl.gov/structure/prospect/

Looking at structures in movement

Some specialists are developing techniques to simulate the movements known to occur in protein structures, when stably folded. Understanding molecular dynamics is an important part of understanding the details of protein-protein interaction, or (for that matter) of enzymatic reactions. Find out more about molecular dynamics and protein movements at

- **The Brooks Lab site:** brooks.Scripps.edu/
- **The El Nemo site:** www.igs.cnrs-mrs.fr/elnemo/
- **The Database of Macromolecular Movements site:** molmovdb.mbb.yale.edu/MolMovDB/

Predicting interactions

Finally, researchers are tackling the difficult problem of predicting whether — and how — two different protein molecules (or a protein and a small molecule) might interact with each other. This is technically referred to as *molecular docking.* The top performance in this field is still very inadequate. In most cases, the 3-D structures of the complex of two interacting proteins cannot yet be computationally derived from their individual structures. Visit the following Web sites to find out more about docking:

- **AutoDock:** www.scripps.edu/mb/olson/doc/autodock/
- **FlexX:** www.biosolveit.de/FlexX/
- **FTDock:** www.bmm.icnet.uk/docking/
- **Hex:** www.csd.abdn.ac.uk/hex/
- **HotDock:** wwwcs.uni-paderborn.de/~lst//HotDock/index.html

Working with RNA

In This Chapter

- Searching databases for RNA molecules
- Predicting the secondary structure of an RNA molecule
- Publishing an RNA secondary structure
- Finding non-coding RNAs over the Internet

> *Biology has at least 50 more interesting years.*
>
> — James D. Watson

In this chapter, we show you that it's possible to use a computer to analyze the most fascinating molecule of them all: the RNA molecule. RNA is incredible because it does almost anything you can think of — including transporting and transmitting genetic information (just like DNA) and performing catalytic functions (just like proteins). (The sidebar "RNA secondary structures" provides a closer look.) In fact, the more that biologists study RNA, the more they wonder why nature bothered inventing DNA and proteins!

There are sound arguments that support the belief that RNA is the true ancestral support of life, but we don't plan to go so far back in time in this chapter! We want to concentrate on another really exciting fact about RNA: It is one of the most clear-cut successes in bioinformatics. If you use them well, RNA computer-based methods can tell you as much as any million-dollar experiment, simply because RNA is much more predictable than DNA or proteins — and thus lends itself well to computer analysis.

This chapter is not here to give you an accurate view of the RNA world. This is simply too complex a subject. (You'll have to go back to your textbooks if you want the full story.) What we show here are the few things that you can easily do on a computer if you know what you're looking for and if you want to find out whether detailed RNA analysis could play a role in your project.

In the first section of this chapter, we show you how you can use mfold, a famous online server, to predict and display the secondary structure of your their sequence similarity is very low. This is a very powerful technique that you can use to discover new members of an established gene class — or new members of an RNA family you've discovered yourself.

The last section gives you a very brief overview of one of the most active field of these last years: the study of small RNA molecules that are known as miRNAs or siRNAs. We show you resources you can use to look for these little guys and use them efficiently in your research.

Sadly, the big players, such as NCBI or the EBI, haven't put as much effort on the RNA side of things as they have in other fields. The good news is that the RNA community is very active and open. On the Internet, you can find a wealth of high-quality resources available to anyone who needs them, including specialized databases and powerful software. We introduce some of those resources in the last section of this chapter.

Predicting, Modeling, and Drawing RNA Secondary Structures

A major biological advance of the 1970s was the discovery that RNA can have complex 2-D and 3-D structures, as shown in Figure 12-1. The very exciting thing about these structures is that they obey laws that seem much simpler and much more predictable than the laws that reign in the world of proteins. Many of the RNA short- and long-range secondary structures rely on standard Watson and Crick base pairing, just as DNA does.

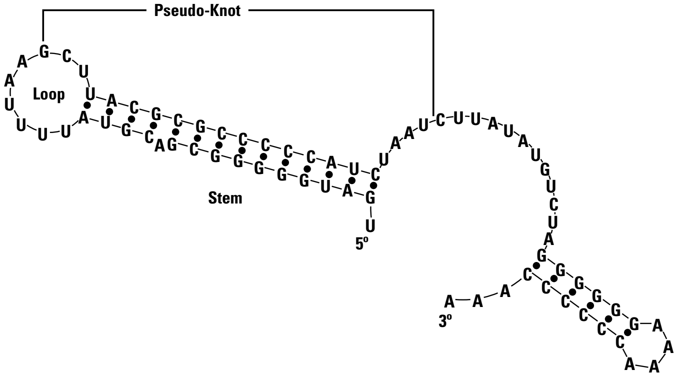


Figure 12-1: Typical secondary structures in RNA.

of its exposed bases are protected from water. A good way to protect an RNA base from the solvent is to pair it with another RNA base. Of course, all possible base-pair combinations are not equally good. Pairing a guanine with a cytosine (for example) is more stabilizing than pairing an adenine with a uracile.

The pairing of these bases forms the *RNA secondary structure.* When a molecule contains two long stretches that are complementary, they yield a nice stable stem. (Refer to Figure 12-1.) The unpaired bases between the stem strands make up a loop. Stems don't have to be perfect; they can also contain unpaired residues (which RNA gurus name *bulges*).

We assume that the natural tendency of the RNA molecule is to reach its most stable conformation by assembling a nice collection of pairwise interactions, giving the molecule the highest stability it can have. This concept is what we call the *lowest-energy model.* Such a stable RNA structure always has a negative energetic value (such as –70 Kcal/mol); if you want to unfold it, you need to provide some energy (heat).

As is true for proteins, we don't know exactly how the RNA molecule finds this lowest energy form — but we know this happens very rapidly contains. Many parameters — such as the stacking of base pairs (making some stems much more stable than others) and loop sizes — influence the fold. Sophisticated algorithms (like that found in the mfold program) can take these subtle effects into account when computing an optimal fold.

Tertiary interactions also play a role in overall stability. These include *pseudo-knots* (again, refer to Figure 12-1) that are usually long-range interactions between a loop and another portion of the RNA molecule. The interaction between the RNA molecule and other chemical elements — such as ions, proteins, or other RNAs — also plays an important role in its stabilization.

Unfortunately, we have difficulties predicting tertiary interactions or the effect of the proteins on the folding of an RNA molecule. So, when you predict the secondary structure of your RNA, this prediction usually depends on the assumption that this RNA folds on its own in the cell. Because this is almost never true, there is always a chance for your prediction to be partly or totally incorrect. The general rule is that the most energetically stable features tend to be reasonably close to the truth.

Using Mfold

Mfold is an implementation of the Zuker algorithm that makes it possible to predict the energetically optimal secondary structure of an RNA molecule. It uses a sophisticated model that can take into account many realistic physical parameters that affect the RNA folding — such as pH, temperature, and the local composition bias of your RNA.

judge the stability of the optimal fold. It is also possible to force mfold to respect an interaction that you know is correct.

You cannot force pseudo-knots in mfold.

To try mfold, you'll need to do a couple of things. First and foremost, you'll need to fetch an RNA sequence. "Where?" you might ask. Well, some hunting is in order; when it comes to non-coding RNA, standard DNA databases are poorly equipped. A few specialized databases exist, but most of them are dedicated to ribosomal RNA. The last section of this chapter gives you the Internet address of a few of these resources (see Table 12-2).

What's the best length for your RNA sequence? Good question. The time mfold needs to analyze sequences increases very rapidly with their length. For instance, if it takes 1 second to fold a molecule that's 100 bases long, it takes about 10 seconds to fold a sequence that's 200 bases long.

The current limit of the server we're going to use for our example is 3,000 bases. This is convenient for most non-coding RNAs except the largest ribosomal ones.

With the preliminaries out of the way, it's time to do some mfolding. We're going to use a sample RNA sequence, but you're free to use one of your own if you want:

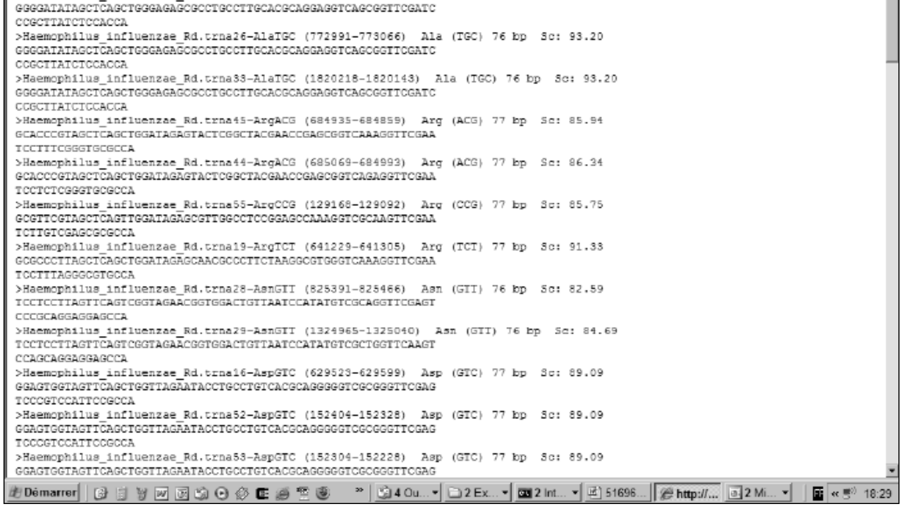
1. **Point your browser to** `lowelab.ucsc.edu/GtRNAdb/Haem_infl/Haem_infl-tRNAs.fa`.

   This returns a list of tRNA sequences predicted by Sean Eddy's tRNA scan server of the Haemophilus influenza genome. The sequences are ordered according to the reliability of their predictions. Sequences at the top of the list are almost 100 percent sure.

2. **Select the first tRNA sequence, as shown in Figure 12-2, and copy it to the Clipboard.**

3. **Point your browser to** `www.bioinfo.rpi.edu/applications/mfold/`.

   The Applications page of the Center for Bioinformatics (Rensselaer Polytechnic Institute) duly appears.



```
GGGGATATAGCTCAGCTGGGAGAGCGCCTGCCTTGCACGCAGGAGGTCAGCGGTTCGATC
CCGCTTATCTCCACCA
>Haemophilus_influenzae_Rd.trna26-AlaTGC (772991-773066)  Ala (TGC) 76 bp  Sc: 93.20
GGGGATATAGCTCAGCTGGGAGAGCGCCTGCCTTGCACGCAGGAGGTCAGCGGTTCGATC
CCGCTTATCTCCACCA
>Haemophilus_influenzae_Rd.trna33-AlaTGC (1820218-1820143)  Ala (TGC) 76 bp  Sc: 93.20
GGGGATATAGCTCAGCTGGGAGAGCGCCTGCCTTGCACGCAGGAGGTCAGCGGTTCGATC
CCGCTTATCTCCACCA
>Haemophilus_influenzae_Rd.trna45-ArgACG (684935-684859)  Arg (ACG) 77 bp  Sc: 85.94
GCACCCGTAGCTCAGCTGGATAGAGTACTCGGCTACGAACCGAGCGGTCAAAGGTTCGAA
TCCTTTCGGGTGCGCCA
>Haemophilus_influenzae_Rd.trna44-ArgACG (685069-684993)  Arg (ACG) 77 bp  Sc: 86.34
GCACCCGTAGCTCAGCTGGATAGAGTACTCGGCTACGAACCGAGCGGTCAGAGGTTCGAA
TCCTCTCGGGTGCGCCA
>Haemophilus_influenzae_Rd.trna55-ArgCCG (129168-129092)  Arg (CCG) 77 bp  Sc: 85.75
GCGTTCGTAGCTCAGTTGGATAGAGCGTTGGCCTCCGGAGCCAAAGGTCGCAAGTTCGAA
TCTTGTCGAGCGCGCCA
>Haemophilus_influenzae_Rd.trna19-ArgTCT (641229-641305)  Arg (TCT) 77 bp  Sc: 91.33
GCGCCCTTAGCTCAGCTGGATAGAGCAACGCCCTTCTAAGGCGTGGGTCAAAGGTTCGAA
TCCTTTAGGGCGTGCCA
>Haemophilus_influenzae_Rd.trna28-AsnGTT (825391-825466)  Asn (GTT) 76 bp  Sc: 82.59
TCCTCCTTAGTTCAGTCGGTAGAACGGTGGACTGTTAATCCATATGTCGCAGGTTCGAGT
CCCGCAGGAGGAGCCA
>Haemophilus_influenzae_Rd.trna29-AsnGTT (1324965-1325040)  Asn (GTT) 76 bp  Sc: 84.69
TCCTCCTTAGTTCAGTCGGTAGAACGGTGGACTGTTAATCCATATGTCGCTGGTTCAAGT
CCAGCAGGAGGAGCCA
>Haemophilus_influenzae_Rd.trna16-AspGTC (629523-629599)  Asp (GTC) 77 bp  Sc: 89.09
GGAGTGGTAGTTCAGCTGGTTAGAATACCTGCCTGTCACGCAGGGGGTCGCGGGTTCGAG
TCCCGTCCATTCCGCCA
>Haemophilus_influenzae_Rd.trna52-AspGTC (152404-152328)  Asp (GTC) 77 bp  Sc: 89.09
GGAGTGGTAGTTCAGCTGGTTAGAATACCTGCCTGTCACGCAGGGGGTCGCGGGTTCGAG
TCCCGTCCATTCCGCCA
>Haemophilus_influenzae_Rd.trna53-AspGTC (152304-152228)  Asp (GTC) 77 bp  Sc: 89.09
GGAGTGGTAGTTCAGCTGGTTAGAATACCTGCCTGTCACGCAGGGGGTCGCGGGTTCGAG
```

Figure 12-2: Selection of a tRNA sequence.

As an alternative, you can also use one of the two European mirrors of mfold:

`bibiserv.techfak.uni-bielefeld.de/cgi-bin/mfold_submit`

`bioweb.pasteur.fr/seqanal/interfaces/mfold-simple.html`

or its Australian counterpart:

`mfold.burnet.edu.au`

4. **Scroll down the page and click the RNA Folding link.**

   The mfold server page appears, as (partially) shown in Figure 12-3.

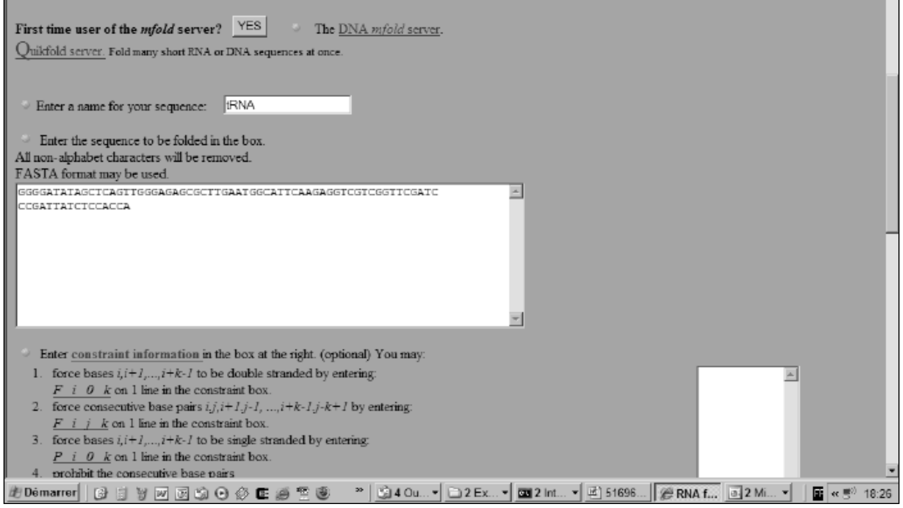
5. **Enter a title for your sequence in the Name field.**

   For our example, we named our sequence tRNA.

6. **Paste your tRNA sequence into the Sequence window.**

   You can safely keep all the other default parameters.

   See the next step list (in the “Forcing interaction in mfold” section) if you want to provide the program with some predefined constraints.



First time user of the *mfold* server? YES ○ The DNA *mfold* server.

Quikfold server. Fold many short RNA or DNA sequences at once.

○ Enter a name for your sequence: tRNA

○ Enter the sequence to be folded in the box.
All non-alphabet characters will be removed.
FASTA format may be used.

```
GGGGATATAGCTCAGTTGGGAGAGCGCTTGAATGGCATTCAAGAGGTCGTCGGTTCGATC
CCGATTATCTCCACCA
```

○ Enter constraint information in the box at the right. (optional) You may:

1. force bases *i,i+1,...,i+k-1* to be double stranded by entering:
   F i 0 k on 1 line in the constraint box.
2. force consecutive base pairs *i,j,i+1,j-1, ...,i+k-1,j-k+1* by entering:
   F i j k on 1 line in the constraint box.
3. force bases *i,i+1,...,i+k-1* to be single stranded by entering:
   P i 0 k on 1 line in the constraint box.
4. prohibit the consecutive base pairs

Figure 12-3: Mfold server home page.

7. **Scroll down the page and click the Fold RNA button near the bottom of the form.**

   The Output page (eventually) appears, as shown in Figure 12-4. The output itself looks rather complicated because it contains links to your results in almost any format you can think of. It also contains thermodynamic information concerning your model, so that you can tell how stable they are.

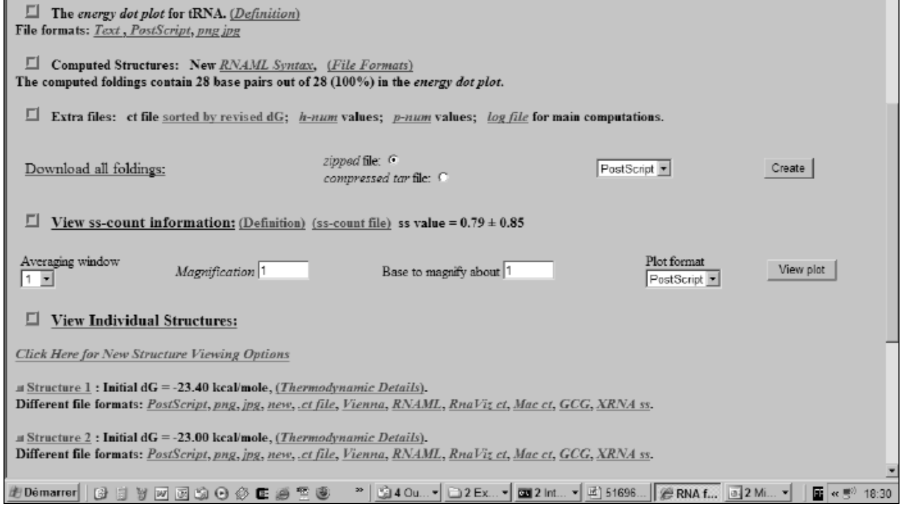
8. **Display the jpg picture of the best-scoring model.**

   a. Scroll down the Output page until you find the section titled View Individual Structures (refer to Figure 12-4).

      The structures are ranked according to their stability. For instance, Structure1 is the most stable model, Structure2 is the second best, and so on.

   b. Look for the listing of different file formats, and then click the jpeg link associated with the Structure1 link. This will load a new page, containing the most stable secondary structure of your sequence (Figure 12-5). This is not necessarily the correct one, but a good candidate.

9. **Save your secondary structure prediction by right-clicking on the image that has just appeared and then choosing the Save As option.**



□ The *energy dot plot* for tRNA. *(Definition)*
File formats: *Text* , *PostScript*, *png jpg*

□ Computed Structures: New *RNAML Syntax*, *(File Formats)*
The computed foldings contain 28 base pairs out of 28 (100%) in the *energy dot plot*.

□ Extra files: ct file sorted by revised dG; *h-num* values; *p-num* values; *log file* for main computations.

Download all foldings: *zipped* file: ● *compressed tar* file: ○ PostScript Create

□ View ss-count information: (Definition) (ss-count file) ss value = 0.79 ± 0.85

Averaging window 1 *Magnification* 1 Base to magnify about 1 Plot format PostScript View plot

□ View Individual Structures:

Click Here for New Structure Viewing Options

Structure 1 : Initial dG = -23.40 kcal/mole, (*Thermodynamic Details*).
Different file formats: *PostScript*, *png*, *jpg*, *new*, *.ct file*, *Vienna*, *RNAML*, *RnaViz ct*, *Mac ct*, *GCG*, *XRNA ss*.

Structure 2 : Initial dG = -23.00 kcal/mole, (*Thermodynamic Details*).
Different file formats: *PostScript*, *png*, *jpg*, *new*, *.ct file*, *Vienna*, *RNAML*, *RnaViz ct*, *Mac ct*, *GCG*, *XRNA ss*.

Figure 12-4: Output of the mfold server.

Interpreting mfold results

After you've given mfold your sequence, it looks for the arrangement that yields the secondary structures with the lowest possible energy. Of course, the correctness of this secondary structure depends on the correctness of the energy model; if the model is wrong, the structure is wrong. (See the "RNA secondary structures" sidebar.)

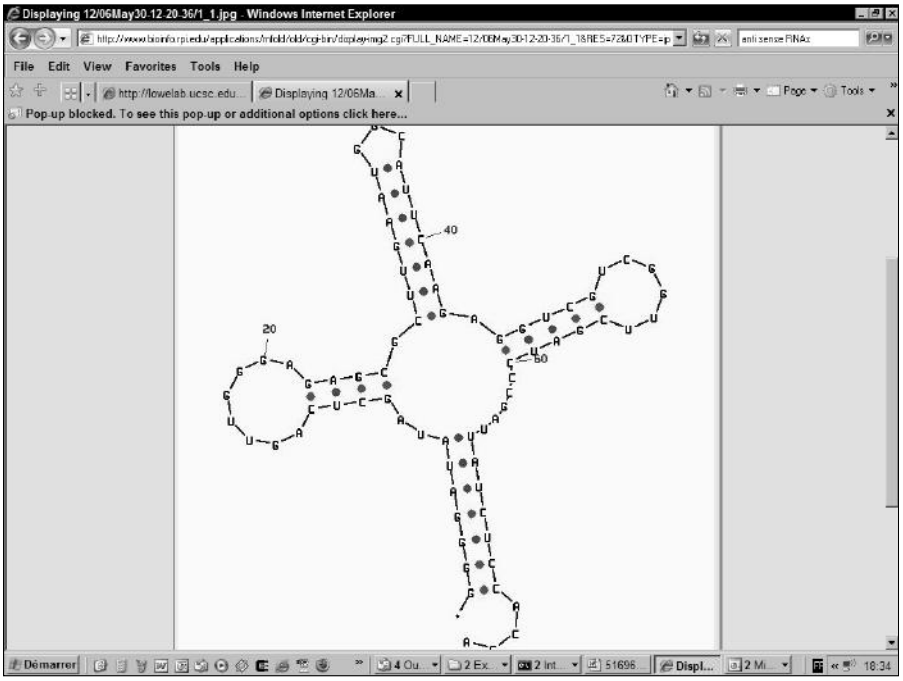
The general rule is that the more stable the conformation mfold predicts, the more likely this interaction is to be correct. To help you get an idea of the nature of this stability, mfold's Output page displays several results related to the stability of your RNA. Among these, you may want to give further attention to the following:

- **The Energy Dot Plot:** The energy dot plot shows the stems that are part of the optimal fold of your sequence.
  - A stem shows up as a black diagonal, perpendicular to the main diagonal.
  - Distal interaction (that is, pairs that involve nucleotides very far from each other) appears as diagonals close to the top-right (or bottom-left) corner.

This section also indicates the free energy of your structure. This energy must be as low as possible. For a molecule that's around 200 bases long, you should expect a free energy lower than –50 Kcal to consider the reported fold as plausible.

- **The Dot Plot Folding Comparisons section:** This very interesting section makes it possible to compare the fold of several suboptimal solutions. In order to do so, follow these steps:

  1. **Choose .jpg from the Image Format drop-down menu.**
  2. **From the Compare Selected Foldings line, select all the alternative structures you want to compare.**
  3. **Click the Do the Comparison button, as shown in Figure 12-6.**


Displaying 12/06May30-12-20-36/1_1.jpg - Windows Internet Explorer
http://www.bioinfo.rpi.edu/applications/mfold/old/cgi-bin/display-img2.cgi?FULL_NAME=12/06May30-12-20-36/1_1&RES=72&DTYPE=p
anti sense RNA:
File
Edit
View
Favorites
Tools
Help
http://lowelab.ucsc.edu...
Displaying 12/06Ma...
Page
Tools
20
40
50
Démarrer
4 Ou...
2 Ex...
2 Int...
51696...
Displ...
2 Mi...
18:34

Pop-up blocked. To see this pop-up or additional options click here...

Figure 12-5: Representation of an RNA secondary structure prediction.

with mfold.

> A graph similar to the one displayed in Figure 12-5 appears. It indicates the elements of predicted secondary structures that are common to the selected suboptimal solutions.
>
> Colored elements correspond to stems that occur only in suboptimal secondary structure predictions.
>
> Finding a high consistency between the optimal solution and the suboptimal ones makes it more likely that the optimal solution is, in fact, biologically correct.

The predictions you make on a single sequence can yield interesting results. RNA is at its best, however, when you make the prediction on a multiple alignment rather than on a single sequence, thanks to what is known as the *covariance phenomenon.*

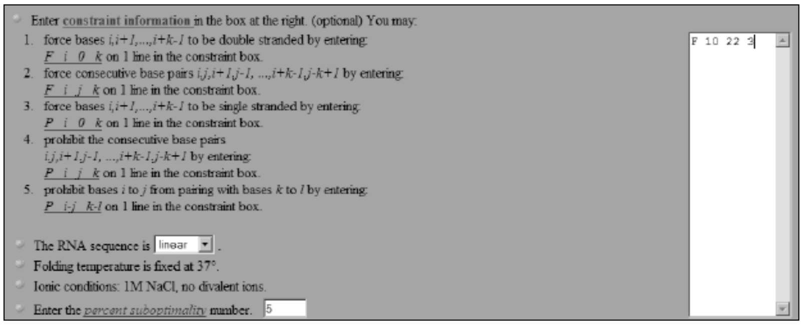
Unfortunately, multiple-sequence-based predictions are not available for free over the Internet, but if you are ready to install some software — and you have a powerful machine — you can use a handy bit of software called Cove, available for download at `www.genetics.wustl.edu/eddy/software/#cove.`

or some of the methods available through registration in Robin Gutell's laboratory at `www.rna.icmb.utexas.edu.`

Forcing interaction in mfold

If you have some experimental data and you know that specific base pairs occur in your RNA sequence, you can force mfold to use this information. For example, you can tell mfold that nucleotides 10, 11, and 12 interact respectively paired with nucleotide 22 — and that you want this interaction to be the beginning of a stem that is 3 nucleotide pairs long (nucleotide 10 with 22, 11 with 21, and 12 with 20).

Figure 12-7 shows you what the interaction form of mfold looks like when it is filled out correctly.



- Enter constraint information in the box at the right. (optional) You may:
  1. force bases *i,i+1,...,i+k-1* to be double stranded by entering:
     F i 0 k on 1 line in the constraint box.
  2. force consecutive base pairs *i,j,i+1,j-1, ...,i+k-1,j-k+1* by entering:
     F i j k on 1 line in the constraint box.
  3. force bases *i,i+1,...,i+k-1* to be single stranded by entering:
     P i 0 k on 1 line in the constraint box.
  4. prohibit the consecutive base pairs
     i,j,i+1,j-1, ...,i+k-1,j-k+1 by entering:
     P i j k on 1 line in the constraint box.
  5. prohibit bases *i* to *j* from pairing with bases *k* to *l* by entering:
     P i-j k-l on 1 line in the constraint box.
- The RNA sequence is linear ▾.
- Folding temperature is fixed at 37°.
- Ionic conditions: 1M NaCl, no divalent ions.
- Enter the *percent suboptimality* number. 5

```
F 10 22 3
```

Figure 12-7: Forcing secondary interaction in mfold.

Searching Databases and Genomes for RNA Sequences

In a cell, all genes are transcribed into RNA molecules so the cell can use them. One thing we all forget sometimes is that there are two kinds of RNAs: those that the cell turns into a protein (using the ribosome machinery) and those the cell uses directly. The latter are known as *non-coding RNAs.*

Non-coding RNAs constitute a rich family, with new members discovered every year. The more we understand the basic mechanisms that govern the life of a cell, the more we realize that non-coding RNAs are everywhere. They help with transcribing, splicing, replicating, and probably many other things we aren't yet aware of. It seems that every basic mechanism of life heavily relies on these little guys — and arguing that a big chunk of biology's future belongs to them is probably the safest bet we'll make in this book.

Unfortunately, with the exception of the ribosomal RNA, these small non-coding RNAs tend to be rather short and poorly conserved. Their most ondary structures. If you have just used mfold to predict the fold of the RNA family you're interested in, you can now use the fold you've discovered to look for new members of the family. In this section, we first show you how to use tRNAscan to look only for tRNAs, and we also introduce PatScan, a method that allows you to look for RNAs with a secondary structure that you can specify yourself.

Finding tRNAs in a genome

tRNAs are small non-coding RNAs that the cell uses to assemble proteins. They constitute one of the best examples of important non-coding genes that are difficult to localize in a genome. All the bacterial and eukaryotic genomes contain tRNAs. Unfortunately, these ubiquitous genes are just as difficult to identify as any small non-coding RNA. They require database search techniques much more sophisticated than BLAST.

The good news here is that there are some very efficient programs out there for predicting tRNA genes in a eukaryotic or a prokaryotic genome. The state-of-the-art method is RNAscan-SE that you can access from the server at Washington University in St. Louis: `selab.janelia.org/tRNAscan-SE/`.

Using PatScan to look for RNA patterns

tRNAscan is perfect if you're interested only in tRNAs. However, if your interest lies in a specific RNA family, you want to use a tool that's more general and enables you to use your own knowledge of the secondary structure of your RNA family. PatScan is ideal for this purpose.

PatScan lets you search databases with patterns that can accurately describe RNA secondary structures. From a computational point of view, this type of search is quite expensive, which is why PatScan requires your e-mail address and returns a result after a few hours. (In our experience, however, PatScan has never taken more than half an hour to return the result.)

Designing a PatScan pattern is the trickiest step when using PatScan. It is something that may require a bit of thought. If you want to know the exact rules that govern the PatScan patterns, you can read the excellent online documentation of this program, available at `www-unix.mcs.anl.gov/compbio/PatScan/HTML/matching_nucleotides.html`.

To help design your pattern, start by identifying the elements you're interested in. Use the following checklist as a guide:

- On the primary structure, report the secondary structure elements.
- Identify the corresponding stems and name them. The name of the left side of the first stem is *p1* and the name of its right side is *~p1*.
- Count the length of each of these elements.
- In each stem element, identify non-canonical Watson and Crick base pairs.
- Write this structure down with your own vocabulary.

  For instance:

  [stem p1 of length 8 to 9] [Loop of length 3 to 8] [stem ~p1 of length 8 to 9, complementary to the left stem p1]
- Turn your secondary structure into a PatScan Pattern.
- p1=8...9 defines the stem p1 that contains 8 to 9 nucleotides. [Note the 3 periods (...) between the 8 and 9.]
- 3...8 defines a pattern of 3 to 8 nucleotides.
- ~p1 defines a stem reverse complementary to p1.

  You can then write your pattern like this: p1=8...9 3...8 ~p1.

  If you want to define several stems, give them different names, such as p1, p2, p3, and so on.

  If you need to, you can have nested stems, like p1=8...9 3...8 p2=5...7 4...5 ~p2 3...8 ~p1.

Table 12-1 RNA-Specific Pattern Rules in PatScan

Rule	*Example*	*Description*
Pairing rule	r1={au,gc} p1=10...12 3...8 r1~p1	Indicates that the p1 pairing only involves au and gc pairs.
Mismatches	p1=10...12 3...8 r1~p1[1,4,2]	The pairing between the two p1 stems can involve at most: 1 mismatch in p1 Vs ~p1 4 unpaired symbols in p1 2 unpaired symbols in ~p1.
Specify patterns	p1=10...12 AUGC~p1	Specifies that the loop MUST contain only AUGC. Note that this pattern must be in uppercase, as opposed to the pairing rules.

With these guidelines, take PatScan on a trial run:

1. **Prepare your PatScan pattern.**
2. **Point your browser to** `www-unix.mcs.anl.gov/compbio/PatScan/HTML/scanner.html`.

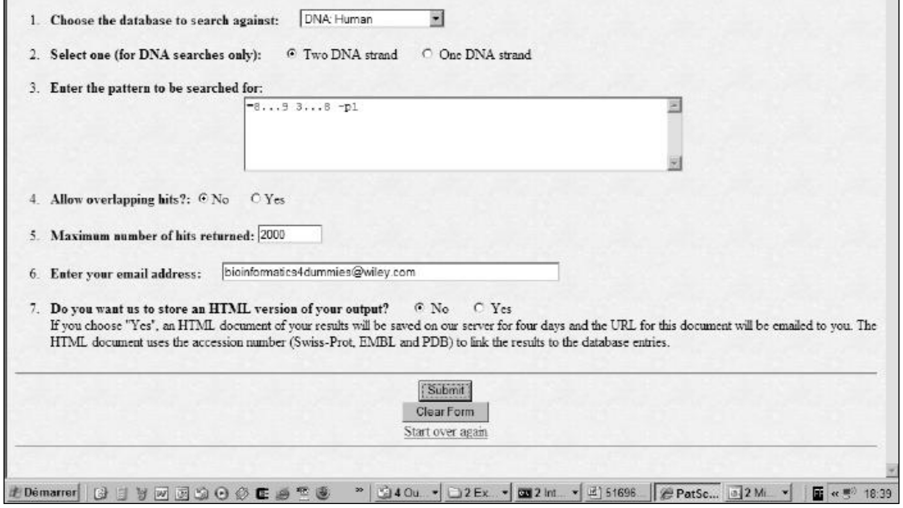
   The PatScan server page appears, as shown in Figure 12-8.
3. **Choose the database you want to search from the Database drop-down menu.**

   In our example, we chose DNA: Human.

   WARNING!

   The default database on this server is Swiss-Prot. In most cases, this database is *not* appropriate for searching RNA secondary structure patterns.

   You must change this database to a DNA database.



1. **Choose the database to search against:** DNA: Human

2. **Select one (for DNA searches only):** ● Two DNA strand ○ One DNA strand

3. **Enter the pattern to be searched for:**

```
=8...9 3...8 ~p1
```

4. **Allow overlapping hits?:** ● No ○ Yes

5. **Maximum number of hits returned:** 2000

6. **Enter your email address:** bioinformatics4dummies@wiley.com

7. **Do you want us to store an HTML version of your output?** ● No ○ Yes
   If you choose "Yes", an HTML document of your results will be saved on our server for four days and the URL for this document will be emailed to you. The HTML document uses the accession number (Swiss-Prot, EMBL and PDB) to link the results to the database entries.

Figure 12-8: Home page of the PatScan server.

4. **Paste or enter your pattern into the Pattern window.**

   We entered p1=8...9 3...8 ~p1.

5. **Enter your e-mail address into the appropriate field.**

6. **Click the Submit button.**

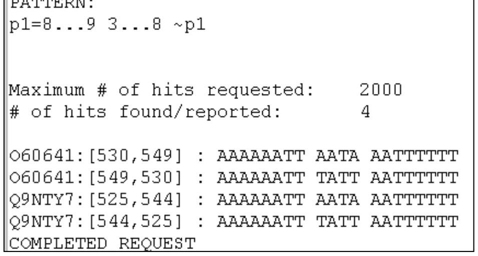
   Depending on the size of the database you have selected — and on the length and complexity of your pattern — your search can take up to an hour.

7. **Analyze your results. (See Figure 12-9.)**

   In Figure 12-9, we have used PatScan to search the Human section of GenBank. Each occurrence of a DNA pattern that matches our request appears on a specific line, which indicates the GenBank/EMBL/DDBJ accession number and the coordinates of this match in the sequences.

WARNING!

The fact that a sequence corresponds to your pattern does *not* necessarily mean that it adopts the secondary structure you had in mind! PatScan checks that the hits are compatible with the specified secondary structure, but it doesn't make sure that this structure is the *most* stable one for these hits.



```
PATTERN:
p1=8...9 3...8 ~p1


Maximum # of hits requested:    2000
# of hits found/reported:       4

O60641:[530,549] : AAAAAATT AATA AATTTTTT
O60641:[549,530] : AAAAAATT TATT AATTTTTT
Q9NTY7:[525,544] : AAAAAATT AATA AATTTTTT
Q9NTY7:[544,525] : AAAAAATT TATT AATTTTTT
COMPLETED REQUEST
```

Figure 12-9: Results of the PatScan server.

Finding the "New" RNAs: miRNAs and siRNAs

One of the most stunning discoveries in biology during recent years was the extent of a phenomenon known as *RNA interference* (*RNAi*) in eukaryotic cells — the possibility for a small, short, non-coding RNA to interfere with a larger, coding RNA — preventing the expression of the corresponding gene. This interference occurs through the binding made possible by complementarity between the small and the long RNA. This is why these small RNAs are sometimes named *antisense RNAs.* (See Chapter 1 and 2 for the basics of nucleic-acid complementarities.) While the possibility of RNA interference had been known for a while in the lab, what stunned everybody was the discovery that nature seems to use it extensively.

Since then, natural short RNAs that can inactivate genes have started popping-up everywhere. They were first discovered in plants and named *silencing RNAs* (*siRNAs*) for their ability to inactivate (silence) genes. These siRNAs are not restricted to plants, and scientists have now found them in most eukaryotic cells. While looking for siRNAs, biologists have started to find a wealth of equally small and unknown RNAs. They have given them the generic name of micro-RNA (miRNA). Nobody really knows what miRNAs really do in the cell, although it's suspected they probably regulate the genes. To make matters worse, not only don't we know exactly what they do, but we also don't know how many of them do it! The reason is that finding miRNAs in a genome is much harder than finding proteins. Nothing makes miRNAs special except for the fact that they are so small. As a consequence, we still don't know how many miRNA genes exist in the human genome. The main distinction between of the most promising (potential) medical innovations on the horizon — provided somebody ever finds a way to efficiently and accurately send these little guys wherever and whenever they are needed in the human body.

Covering miRNAs, siRNAs, and the phenomenon of RNA interference goes well beyond the *For Dummies* scope — but Table 12-2 gives you enough to start hunting for miRNAs in your favorite genome.

Table 12-2 Hunting Micro RNAs (miRNAs) over the Web

Address	*Description*
`sirna.cgb.ki.se/`	An extensive collection of resources on silencing RNAs.
`itb.biologie.hu-berlin.de/~nebulus/sirna/v2/`	A database of all known human silencing RNAs.
`microrna.sanger.ac.uk/sequences`	The home of miRNAs at the Sanger Center in the UK. Probably one of the most extensive resources on micro-RNAs.
`cbit.snu.ac.kr/~ProMiR2/`	A resource for predicting miRNAs using probabilistic methods.
`pictar.bio.nyu.edu`	Prediction of the potential target of your miRNA on complete genomes.
`bibiserv.techfak.uni-bielefeld.de/rnahybrid/`	A resource for predicting the potential target of your miRNA on a user-provided genomic sequence.
`mirna.imbb.forth.gr/microinspector/`	Runs your genomic sequence against an exhaustive database of miRNAs.

Doing RNA Analysis for Free over the Internet

If you're interested in RNA, everything you need is on the Internet. Make your way through this section to get a better sense of the many excellent resources available online for you.

commonplace that any time you find a new bacterium, the first thing you want to do is sequence its ribosomal RNA so you can see where it fits on the big tree of life.

Several laboratories have dedicated their entire work to helping the scientific community use rRNA sequences. Table 12-3 lists some of the most famous of these resources.

Table 12-3 Ribosomal RNA Resources on the Internet

Address	*Description*
`rdp.cme.msu.edu`	Provides data and services, including the possibility to make online phylogenetic analysis.
`www.psb.ugent.be/rRNA/lsu/`	A European database on the larger of the two ribosomal subunits. It contains predicted structures. It is possible to query the database online. Features lots of online software.
`www.psb.ugent.be/rRNA/ssu/`	The "other" European database, this time dedicated to the small ribosomal subunit.

Finding the small, non-coding RNA you need

Small, non-coding RNAs are poorly represented in major databases. Fortunately, excellent alternative databases make it possible to access the genes you're interested in. Table 12-4 is a partial list of these resources.

Table 12-4 Some Non-Coding RNA Resources

Address	*Description*
`condor.bcm.tmc.edu/smallRNA/smallrna.html`	Dedicated to small non-coding RNAs.

(continued)

bighost.area.ba.cnr. it/BIG/UTRHome/	Dedicated to the untranslated regions of genes.
www.indiana.edu/~tmrna/	Dedicated to the recently discovered *tmRNAs* that are both transfer and messenger RNAs. (If you don't yet know what this is, you MUST take a look at this fascinating Web site!)

Generic RNA resources

Well-maintained and usually up to date, these generic RNA resources (see Table 12-5) make it easy for you to locate the server that corresponds to your needs. On these lists of links you can find databases, software, and online servers.

Table 12-5 A List of Generic RNA Resources

Address	***Description***
bioinfo.lifl.fr/rna/	A site dedicated to the detection of non-coding RNAs
www.imb-jena.de/ RNA.html	RNA World, one of the most complete sites currently available
www.rnabase.org/links/	Another very complete list of sites

Building Phylogenetic Trees

In This Chapter

- Finding out what phylogenetic trees can do for you
- Making the right multiple sequence alignment for the right tree
- Estimating distances between your sequences
- Building a Neighbor Joining Tree
- Evaluating the quality of a tree by bootstrapping
- Finding tree resources on the Internet

> *Nothing in biology makes sense except in the light of evolution.*
>
> — Theodosius Dobzhansky (1973)

The purpose of this chapter is to show you how you can use a computer and a few online services to run a *phylogenetic analysis,* the scientific procedure that lets you reconstruct the evolutionary history of a group of organisms or sequences. If you know what phylogenetic trees are all about — but you don't have a clue how to generate one with a computer — this chapter is for you.

Here we show you how you can select sequences, turn them into a multiple sequence alignment, and then turn this alignment into a tree (well, okay, a diagram that *looks* like a tree). We show you three methods:

- **ClustalW:** This is an easy method that you must use as a black box. You cannot really control ClustalW, but it is reassuring to know that it uses Neighbor Joining, a fairly well-established tree method.
- **Phylip:** A more sophisticated method that enables you to control every parameter that plays a part in the making of your tree.
- **phyML:** A very accurate reconstruction method that uses maximum likelihood, a method that specialists often consider to be the most accurate. As its name implies, Maximum Likelihood methods try to reconstruct the tree that more likely to correspond to your alignment.

Finding Out What Phylogenetic Trees Can Do for You

The purpose of *phylogeny* is to reconstruct the history of life and explain the present diversity of living creatures. This can be represented as a huge genealogic tree (the tree of life). The underlying principle of phylogeny is to try to group living creatures according to their level of similarity. In this context, we assume that the more similar two species are (such as human and ape), the closer they are to their common ancestor. Phylogeny is not a new subject, and whether you trace back the birth of modern biology to Darwin and *The Origin of Species* or to Aristotle and his notion of categories, you can't escape this daunting fact: Biology is very much about classifying — and the best means of classification we have is phylogeny.

Phylogenetics is a special kind of phylogeny that relies on the comparison of equivalent genes coming from several species for reconstructing the genealogic tree of these species and finding out who is the closest relative of whom in the family. If necessary, you can also apply phylogenetic methods to the various genes of a gene family to reconstruct the history of the gene family by the same means. (Take note that these trees make sense only if you believe in evolution!)

The molecular stories you can uncover with phylogenetics are incredibly rich. In fact, we find it difficult *not* to see a parallel between the destiny of famous families (such as the Medicis, the Borgias, or the Kennedys) and the fates of gene families. When we unravel the chain of events that makes the story of a protein family, we find tales of mutations and deletions, duplications or speciation, loss and gain of function, inactivation, and all the other traumatic events that shaped the world as it is today. Nothing is ever taken for granted while life evolves!

Phylogenetics is here to let you discover all this. Phylogenetics is not a technique, nor even a discipline; phylogenetics is a science — and a major one. The mere task of laying out the general ideas of phylogenetics is way beyond the scope of this chapter, so if this subject is new to you, we urge you to consult some of the excellent textbooks available on this subject.

puted with all known ribosomal RNAs. This can give you a fairly good idea of who this bacterium really is.

- **Discovering the function of a gene:** If you're studying a gene, you can use phylogenetic trees to be sure that the gene you're interested in is orthologous (more about that in a minute) to another well-characterized gene in another species.
- **Retracing the origin of a gene:** Most genes within a genome travel together through evolutionary time. However, from time to time, individual genes may jump from one species to another — for instance, piggybacking a virus infection. Phylogenetic trees are a great way to reveal such events, which are called *horizontal* (or *lateral*) *transfers.*

Two genes are *orthologous* if they come from two different organisms, derive from a common ancestor gene, and have been separated only by speciation events (as opposed to gene duplications). In theory, two genes that are orthologous often have the same exact function (have similar roles) in the two different organisms they come from. On the other hand, when the two genes arose from gene duplication, we say that they are *paralogous.* Two genes that are paralogous belong to the same family but are more likely to have different functions than they would if they were orthologous.

To give you an image, if the joystick of a Boeing 747 were a gene, it would be orthologous with the steering wheel of your car. In contrast, the left front wheel of your car is paralogous with the right front wheel.

Preparing Your Phylogenetic Data

Biologists have been dealing with phylogeny for many decades now, hence the large number of different techniques available. Things have become a bit simpler since people started doing *phylogenetics* (phylogeny applied to gene data). Nonetheless, there are still many schools of thought around — and ancient fights are still raging in the world of phylogeny. Be aware that when you choose a method, you're also choosing sides! In this chapter, we try to focus on the methods that have attracted the smallest amount of controversy over the past few years.

Phylogenetics uses genes to reconstruct evolutionary history. This type of data is much easier to interpret than the *morphology data* — derived from an analysis of an organism's form and structure — that biologists used to rely on. It's possible to compare gene sequences in a much more objective fashion than it is to compare morphological characteristics.

When you do phylogenetics, good data means *a highly accurate multiple sequence alignment that contains properly chosen sequences.* In the first part of this section, we show you how you can prepare this kind of data.

Choosing the right sequences for the right tree

When you build a phylogenetic tree, you make the assumption that the sequences you are comparing have a common ancestor. If your sequences are similar enough, this is a reasonable hypothesis. You want to use your tree to reconstitute the history of these sequences to find out who is the closest relative to whom.

Of course, this vision implies that evolution is always divergent — thus if two sequences are similar, we rule out the possibility that this similarity may result from convergent evolution. In the vast majority of cases, this rule makes sense. At the molecular level, convergent evolution does exist — but it seems to be the exception rather than the rule.

Using DNA or protein sequences

To establish the relationship between two sequences, you want to measure the time that separates their divergence from their common ancestor. How long ago did they part? To answer this question, compare your sequences and measure a distance or establish an evolutionary scenario. Should you do this on the protein or on the DNA sequence? We have a complex answer to this seemingly simple question:

- **If your DNA sequences are *more* than 70 percent identical:**

  You can make a DNA multiple sequence alignment. If your sequences are coding for proteins, however, this is not recommended.

proteins and to build the multiple sequence alignment with the proteins.

If your sequences are too similar at the protein level, you can thread the DNA sequences back onto the protein alignment. Use pal2nal for that purpose — like this:

```
coot.embl.de/pal2nal
```

If you do not have the DNA sequence that corresponds to your proteins, you can use Protogene to automatically fetch the bona-fide DNA sequences of your proteins. Protogen is a Tcoffee tool, available at `www.tcoffee.org`

After you've completed the DNA to protein threading, you have two possibilities:

- **If most synonymous mutation sites are different:** This is a typical case of saturation. In this case, the distance measurements you make at the protein level are more accurate.
- **If synonymous sites are not saturated:** Distance measurements made at the DNA level are more accurate in theory.

In practice, unless your sequences are almost identical, it is easier to keep working at the protein level. This may not be as accurate as working with DNA sequences, but, in most cases, you can expect the results to be reasonably good.

White T-shirts, spaghetti, evolution, and phylogeny

When we think of evolution, we tend to think of natural selection, survival of the fittest, struggle for life, and all these testosterone-loaded concepts. In molecular terms, gurus name this *neo-Darwinism.* This popular (though ancient) theory states that the only reason mutations stay in a population is because they are helpful and increase the fitness of the individuals that carry them. But this theory doesn't account very well for synonymous mutations or other harmless DNA alterations.

In the 1960s, the increased availability of molecular data led to Motoo Kimura's elaboration of neutralism. *Neutralism,* also known as the neutral theory, challenges neo-Darwinism by stating that most mutations are either lethal or neutral. Everybody agrees that lethal mutations are disposed of right away — but for neutralists, the possibility of a new non-lethal mutations spreading (or not spreading) in a population depends mostly on chance. Even if these mutations are advantageous, chance is the driving

(continued)

out altering its function. Of course, after these harmless-but-useless mutations are fixed, they can also find a niche — and give some advantage to the individuals who carry them — later on. Neutralism is a very powerful theory because it accounts well for the way in which species build their adaptation potential before they need it.

If you're doing phylogeny, these neutral mutations are really the ones you're interested in. You do not want to see adaptation; you want to measure evolutionary distances and the best measure of evolutionary distance you have is counting the number of random and quasi-neutral events that occurred on your sequences since they separated from their ancestor. You can think of these mutations like each tick of a clock that has been working, one mutation at a time, for a million years.

To give you an image, imagine that this morning, like every day, you took a white T-shirt and duplicated it for your two kids (Dave and Benny), and off they went. It was a long day; Dave and Benny both ate their spaghetti bolognese, both proving especially adept at emulating Michelangelo with the verve and passion with which they sketched out — in brilliant red sauce — a few Last Judgments on their white T-shirts. Now suppose that each drop of colored-something landing on the immaculate shirts is a mutation — and the number of spots indicates how far the day has progressed. If you want to know how long Dave and Benny have been wearing these shirts, you can simply count the spots. This is true neutralism, and this is the ideal situation when you compare sequences.

Imagine now that there is a bit of selection going on: Dave and Benny may be very anxious about your reaction when they come back (okay, okay, probably unlikely, but bear with us for the sake of argument), and they carefully

lent of selected mutations. When this goes on, you can no longer tell how long the shirt has been worn.

Non-selected mutations (neutral) are much better for evaluating distances because they accumulate smoothly. In coding DNA, the synonymous mutations are the more typical non-selected mutations. This is why DNA is better for measuring distances than protein sequences whose evolution is much more constrained. (These protein guys have work to do!)

Imagine now that the kids have had a second or third go at the spaghetti bolognese. Drops of sauce have landed on previous spots, and the shirts are now completely saturated with tomato sauce. What can you tell from this? How many spaghetti meals have Benny and Dave had? One? Two? Five? It's impossible to say!

This analogy is typical of how our evolutionary time measure becomes saturated — when there have been one or more mutations on every neutral site of our DNA sequence. Determining how long this has been going on becomes impossible. Of course, the closer you are to saturation, the more imprecise your measurements are going to be.

One more thing before we rush to wash these filthy T-shirts: What if, on a certain day, Benny and Dave have not been doing the same thing? One may come home sparkling clean and the other muddy. Will you assume that they have spent their days in different time dimensions, or would you rather conclude that their shirts have evolved at different paces (or places)? Different paces sound more likely — although, if you ask the kids, you may get a more confusing answer. Such differences also occur with genes — and although we like to think that there is a universal molecular clock, we know it tends to tick at a different pace in every gene.

during which a common ancestor gives birth to two subgroups that slowly drift away from their common genetic makeup to become distinct species. Assuming that the genomes are not rearranged in the two new species, two genes are orthologous when they correspond to the same ancestral gene in the ancestral genome. Biologists usually expect orthologs to have similar functions and structure. In Figure 13-1, A1 and A2 are orthologs, and so are B1 and B2.

- **Paralogs:** Paralogs are homologues separated by a duplication event, meaning that within a genome, a gene was duplicated. One of the duplicates may have kept the original function while the other duplicate could have acquired a new function. You can expect paralogs to have different but related functions. For instance, A1 and B1 are paralogs in Figure 13-1.
- **Xenologs:** *Xeno* is a Greek word that means “foreigner.” Xenologs result from a *lateral transfer* between two organisms — a direct DNA transfer between two species. This means that one of the species contains a gene that does not have the same history as the genome in which it is inserted. A typical case of lateral transfer (or xenologs) is the acquisition of the isoleucyl-tRNA sytnthase from their host by several bacteria. The isoleucyl-tRNA sytnthase is a protein involved in the synthesis of other proteins, and its acquisition by bacteria seems to help them becoming antibiotic resistant. When this happens, the newly acquired isoleucyl-tRNA sytnthase is a xenolog of the other tRNA synthases contained in the bacteria.

When you select a group of homologous genes to make a phylogenetic tree, you always make what biologists call a *gene tree*. It is a tree that tells the story of the genes it contains.

If you select all the *paralogous* members of a large human gene family, your gene tree tells the story of this gene family only. You can only use it to reconstruct the chain of duplications that led from one single ancestral gene to the current situation.

If you select a group of genes that are all *orthologous* from different species, the gene tree you get looks very much like a species tree — which lets you reconstruct the speciations that occurred while the species you’re looking at (or their ancestors) were diverging. The best example of this type of gene tree is the ribosomal RNA phylogenetic tree that biologists use to reconstruct the big tree of life. Ribosomal RNA genes exist in every species and are clearly orthologous between species.

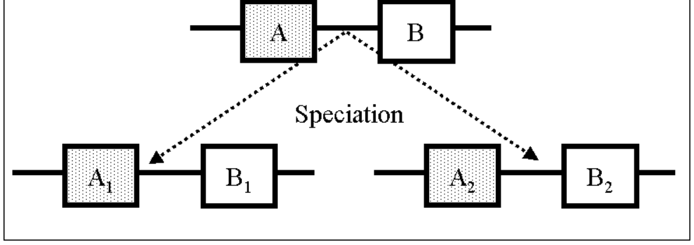


Figure 13-1: Orthology and paralogy.

You can do things the other way round, of course, and use the tree to ask whether your sequences can be called orthologs or paralogs. This is a valid strategy if you know the true species tree.

Two genes that are homologous and come from two different species are not necessarily orthologous. They can also be paralogous — as A1 and B2 are in Figure 13-1. Mixing orthologs and paralogs in a phylogenetic tree is a major source of trouble. Unfortunately, avoiding it is difficult.

In theory, there's no simple solution to the problem of establishing orthology — and demonstrating it is virtually impossible. However, in practice, authors have been using an empirical criterion that works reasonably well: the reciprocal BLAST best match. Here's how it's done:

1. **Choose a sequenceA from genomeA.**
2. **Carry out a BLAST search comparing your sequenceA and every sequence in another complete genomeB. (For more on BLAST searches, see Chapter 7.)**

   The BLAST search returns sequenceB as a top hit.
3. **Carry out a BLAST search comparing sequenceB and every sequence in genomeA.**

   If this search returns sequenceA as a top hit, you can treat sequenceA and sequenceB as orthologs from genomeA and genomeB.

   This result doesn't demonstrate that sequenceA and sequenceB actually *are* orthologous, but it does mean you won't be doing something very wrong if you consider them so.

find it online at www.ncbi.nlm.nih.gov/COG/. Other collections of homologous genes include HOGENOM (former HOBACGEN) and HOVERGEN developed by the Pôle Bioinformatique Lyonnais (http://pbil.univ-lyon1.fr/).

Creating the perfect set

"Perfect" is meant to be ironic here, but we've listed in Table 13-1 a few points you should take into account when assembling your sequence set.

Table 13-1 Preparing a Set of Sequences for Making a Phylogenetic Tree

Problem	*Reason and Solution*
Avoid sequence fragments	Incomplete sequences make multiple sequence alignments and tree reconstruction methods very sick! You want to avoid these at all costs — or you at least want to use the same fragment for all the sequences.
Avoid Xenologs	Unless your purpose is to study them, avoid including genes that result from lateral transfer.
Avoid recombinant sequences	Some proteins result from the combination of several proteins. This is especially common in viruses. In terms of phylogeny, such proteins have two ancestors rather than one — and standard tree methods are not equipped to represent this kind of relationship.
Avoid large complex families	Very large families that contain various domains and repeats can be very tricky to analyze. The ABC transporter is a good example of this kind of troubled family. If you can, stay away from these protein gangs — or try to work on smaller, more uniform subsets.

(continued)

	lar sequences — as many as you want. Unfortunately, it isn't because you need it that you can do it. If you've been through Chapter 9, you know this is not a good thing because most methods find it hard to deal with large datasets. Analyzing large sets means that you may not be able to use the most powerful methods, or that you will need to install these programs on your computer.
Add an outgroup to your dataset	An *outgroup* is a sequence that you know has diverged long ago from the rest of the set. For instance, horse hemoglobin would be an outgroup if you're analyzing primate hemoglobin. This outgroup helps you by adding a root to your tree that symbolizes the first common ancestor of all your sequences. The choice of an outgroup relies mostly on biological criteria, yet you must also make sure that this sequence is similar enough to the rest of the set so your multiple-sequence-alignment method can align it (see Chapter 9).

Living fossils do not exist! Unless you are using fossil DNA sequences, the sequences you are comparing all have the same age. Even if one of the species has retained more similarity with the common ancestor, its sequences have accumulated as many neutral mutations as the most sophisticated member of the family.

Preparing your multiple sequence alignment

Many phylogenetic tree-reconstruction methods require a special table called *distance matrix.* This table contains the distances (or counts the number of evolutionary events) that separate each pair of sequences in your dataset.

In theory, you can build this matrix by comparing all your sequences two by two and then filling up the distance matrix. In practice, that doesn't work — the

2. **Measure the distances on the sequences' pairwise projection.**

   The *pairwise projection* is a pairwise alignment that you *extract* from the multiple sequence alignment. It benefits from the information contained in ALL the other sequences.

The quality of your multiple sequence alignment is the real limiting factor when you make a tree; there is no way you can make a good tree with a bad alignment. (If you aren't sure what a multiple sequence alignment *really* is, make your way through Chapter 9.)

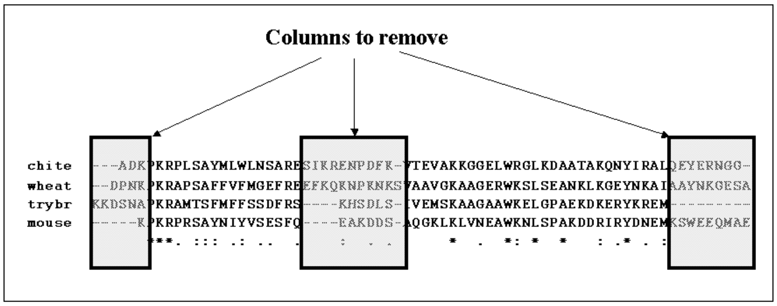
Computing your multiple sequence alignment

To compute your multiple sequence alignment, you can use ANY of the methods we introduce in Chapter 9. These online servers include

- **ClustalW:** `www.ebi.ac.uk/clustalw`
- **MUSCLE:** `phylogenomics.berkeley.edu/cgi-bin/muscle/input_muscle.py`
- **Tcoffee:** `www.tcoffee.org`

Making sure you have the right multiple sequence alignment

Before using your multiple sequence alignment for building a tree, you want to make sure that it's as accurate as possible. In this section, we give you a few criteria — and a list of things to do — to improve the suitability of your multiple alignment. (Figure 13-2 shows you the kinds of regions our handy steps list below targets for removal from your alignment.)



```
chite  ---ADKPKRPLSAYMLWLNSARESIKRENPDFK-VTEVAKKGGELWRGLKDAATAKQNYIRALQEYERNGG-
wheat  --DPNKPKRAPSAFFVFMGEFREEFKQKNPKNKSVAAVGKAAGERWKSLSEANKLKGEYNKAIAAYNKGESA
trybr  KKDSNAPKRAMTSFMFFSSDFRS----KHSDLS-IVEMSKAAGAAWKELGPAEKDKERYKREM---------
mouse  -----KPKRPRSAYNIYVSESFQ----EAKDDS-AQGKLKLVNEAWKNLSPAKDDRIRYDNEMKSWEEQMAE
             ***. ::: .: ..  .    :  . .        *  .  *: *  *   : .* . :
```

Figure 13-2: Preparing a multiple sequence alignment.

of these methods, such as Clustalw, can use the *complete-deletion* techniques and ignore every column that contains a gap.

2. **Remove the extremities of your multiple alignment.**

   The N-terminus and the C-terminus tend to be poorly conserved — and therefore poorly aligned. You can safely remove them (refer to Figure 13-2).

3. **Remove the gap-rich regions of your alignment.**

   Internal, gap-rich regions in a multiple sequence alignment often correspond to loops. Even if your program returns an alignment, it does not mean that this alignment is meaningful.

4. **Be sure to keep the most informative blocks.**

   The ideal multiple alignment for building a tree would be a high-quality alignment of sequences with a low level of identity, so that each position contains a trace of the family history. Here are some pointers:

   - In Chapter 9, we show you what a good block looks like in a multiple sequence alignment. It's typically 20 to 30 amino acids long, and contains a few conserved positions. Such blocks are ideal for producing high-quality trees.
   - If you want, you can use the Tcoffee server to evaluate your multiple sequence alignment and remove columns that are unlikely to be correctly aligned. (See Chapter 9 for more on how to do this.)
   - The best way to edit your multiple alignment is to use Jalview, the multiple alignment editor that we introduce in Chapter 10.

   If you want a quick-and-dirty solution, here's a way to use Microsoft Word to remove blocks very efficiently.

   a. While pressing the Alt key on your keyboard, use the mouse to select entire columns in your alignment.

   b. When you've selected everything you want to remove, press the Delete key to remove the selected block.

Don't even *try* to do anything else to your sequences and alignments while using a word processor! If you need to edit your sequences, use a proper alignment editor like Jalview (Chapter 10):

This chapter shows you how to make distance-based and maximum-likelihood trees. Distance-based trees aren't necessarily the most accurate, but they are generally applicable and easy to set up. You can apply distance-based tree-reconstruction methods on most situations; they tend to generate very reasonable results. Maximum Parsimony and Maximum Likelihood methods are (in theory) the most accurate — but they take more time to run. Experts tend to give a slight hedge to Maximum Likelihood, but the debate is still largely open.

When you decide to use a distance-based method to compute your phylogenetic tree, consider these four major ingredients:

- **Your multiple alignment:** Which method? Which sequences?
- **The distance measure:** Which substitution matrix?
- **The tree-reconstruction algorithm:** UPGMA, NJ, Fitch, or Kitsch?
- **The type of tree you want to display:** A tree without distances (called a *cladogram*) or a tree displaying distances (called a *phenogram*)? A rooted or an unrooted tree?

In this section, we show you how you can use the Internet to compute a tree and to display it in any format you need.

Computing your tree

Combining the right ingredients for making your tree is much like cooking. There is really no such thing as *the best recipe ever;* the best tree you can produce is mostly a matter of taste and the mood of the day.

As when it comes to food, there are two very different ways to produce trees. The first one uses ClustalW; it's quick, hassle-free, and somehow very similar to (good) fast food. We show you in a steps list how you can simply cut and paste your sequences to obtain a tree while avoiding embarrassing questions.

The second step list is for those of you out there who love to buy fresh vegetables on the market and make your own salad dressing — sticklers for detail, if you catch our drift. With these steps, you can control every ingredient that

Producing an (almost) instant phylogenetic tree with ClustalW

If you need a tree and you have firmly decided that you do not want to know *anything* about this barbaric kind of gardening, ClustalW is definitely for you. ClustalW is, above all, a multiple-sequence-alignment package, but some of the ClustalW servers let you use this program to produce phylogenetic trees.

On the EBI ClustalW server — and on most ClustalW Web servers — if you feed ClustalW some unaligned sequences, the program returns a tree. This tree you get, however, is **NOT** a phylogenetic tree; it is a guide tree that ClustalW uses to assemble the multiple sequence alignment. In general, any tree with a `.dnd` extension returned by ClustalW is NOT a phylogenetic tree.

You CANNOT use this tree in place of a phylogenetic tree. Doing so would be a mistake (as in: incorrect results and a false sense of security).

In this section, we show you how you can use the EBI ClustalW server to produce a phylogenetic tree. We assume you have already generated your multiple sequence alignment; if you have not, you can download a dummy alignment from `www.tcoffee.org/dummy_aln2.html`.

Our dummy alignment contains homologous fragments of the gamma fibrinogen. They have been sequenced in several species for comparative studies. We can use fragments rather than entire sequences because these fragments are all homologous on their entire length. We have renamed the fragments with the species names, since trees with species names are much easier to read — unless, of course, you have a formidable memory,

> *O12957 (Sheep), O02672 (Moose), O02683 (Giraffe)*
>
> *O02690 (Chevrotain), O02681 (Beluga) O02687 (Sperm_whale)*
>
> *O02673 (Rorqual), O02688 (Pig), O12959 (Peccary)*
>
> *O02677 (Dromedary), O02689 (Tapir), O02682 (Horse)*
>
> *O02676 (Hyena), O02680 (Coyote), O12954 (Hippopotamus)*

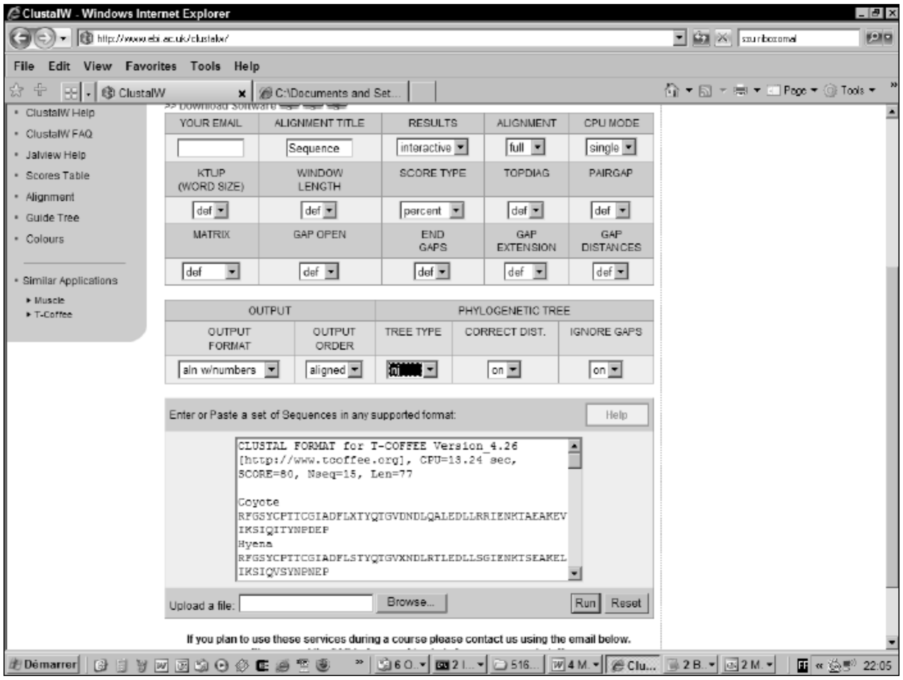
Of course, you could use any set of sequences aligned with any of the servers we described in Chapter 9.

1. **Point your browser to** `www.ebi.ac.uk/clustalw/`**.**

3. **Choose NJ from the Tree Type drop-down menu in the Phylogenetic Tree section, as shown in Figure 13-4.**

   This menu enables you to choose any of several different methods for computing your tree. Specialists generally agree that NJ, the Neighbor Joining method, is the best one for the task at hand.

   If you set the Phylogenetic Tree: Tree Type drop-down menu to anything other than None, you must provide ClustalW with aligned sequences. "Why?" you might ask? Well, when you choose anything but None from the drop-down menu, ClustalW *stops computing alignments* and assumes that you have provided it with aligned sequences. If you don't do as it assumes — and simply give the server unaligned FASTA sequences — it assumes the sequences *are* aligned and computes a tree. Of course (unless your sequences can be aligned without gaps), this tree will be wrong!



YOUR EMAIL	ALIGNMENT TITLE	RESULTS	ALIGNMENT	CPU MODE
	Sequence	interactive	full	single
KTUP (WORD SIZE)	WINDOW LENGTH	SCORE TYPE	TOPDIAG	PAIRGAP
def	def	percent	def	def
MATRIX	GAP OPEN	END GAPS	GAP EXTENSION	GAP DISTANCES
def	def	def	def	def

OUTPUT		PHYLOGENETIC TREE		
OUTPUT FORMAT	OUTPUT ORDER	TREE TYPE	CORRECT DIST.	IGNORE GAPS
aln w/numbers	aligned	nj	on	on

Enter or Paste a set of Sequences in any supported format:

```
CLUSTAL FORMAT for T-COFFEE Version_4.26
[http://www.tcoffee.org], CPU=13.24 sec,
SCORE=80, Nseq=15, Len=77

Coyote
RFGSYCPTTCGIADFLXTYQTGVDNDLQALEDLLRRIENKTAEAKEV
IKSIQITYNPDEP
Hyena
RFGSYCPTTCGIADFLSTYQTGVXNDLRTLEDLLSGIENKTSEAKEL
IKSIQVSYNPNEP
```

Upload a file:

If you plan to use these services during a course please contact us using the email below.

Figure 13-3: Computing a phylogenetic tree with ClustalW.

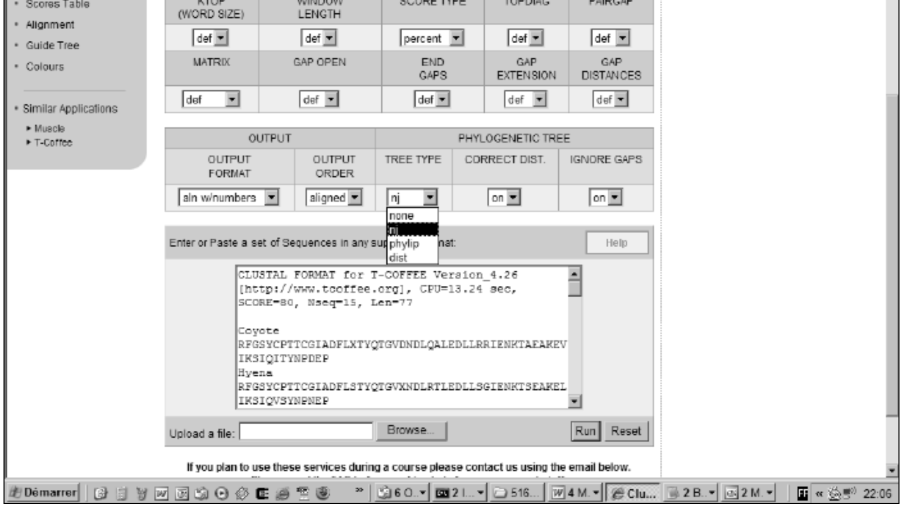


Figure 13-4: The Phylogenetic Tree: Tree Type drop-down menu in ClustalW.

KTUP (WORD SIZE)	WINDOW LENGTH	SCORE TYPE	TOPDIAG	PAIRGAP
def	def	percent	def	def
MATRIX	GAP OPEN	END GAPS	GAP EXTENSION	GAP DISTANCES
def	def	def	def	def

OUTPUT		PHYLOGENETIC TREE		
OUTPUT FORMAT	OUTPUT ORDER	TREE TYPE	CORRECT DIST.	IGNORE GAPS
aln w/numbers	aligned	nj	on	on

Enter or Paste a set of Sequences in any sup nat:

```
CLUSTAL FORMAT for T-COFFEE Version_4.26
[http://www.tcoffee.org], CPU=13.24 sec,
SCORE=80, Nseq=15, Len=77

Coyote
RFGSYCPTTCGIADFLXTYQTGVDNDLQALEDLLRRIENKTAEAKEV
IKSIQITYNPDEP
Hyena
RFGSYCPTTCGIADFLSTYQTGVXNDLRTLEDLLSGIENKTSEAKEL
IKSIQVSYNPNEP
```

Upload a file:

If you plan to use these services during a course please contact us using the email below.

4. **Choose On from the Correct Dist. drop-down menu (located directly to the right of the Phylogenetic Tree: Tree Type drop-down menu).**

   This option makes it possible to correct for multiple substitutions. It tends to elongate the branches of your tree; it's mostly useful for distantly related sequences.

5. **Choose On from the Ignore Gaps drop-down menu.**

   This causes ClustalW to ignore every column of your multiple alignment that contains gaps. Turning this option on ensures that *all* sequences are compared on the same number of residues. Keep in mind that phylogeny gurus usually prefer to ignore positions (columns) that contain gaps.

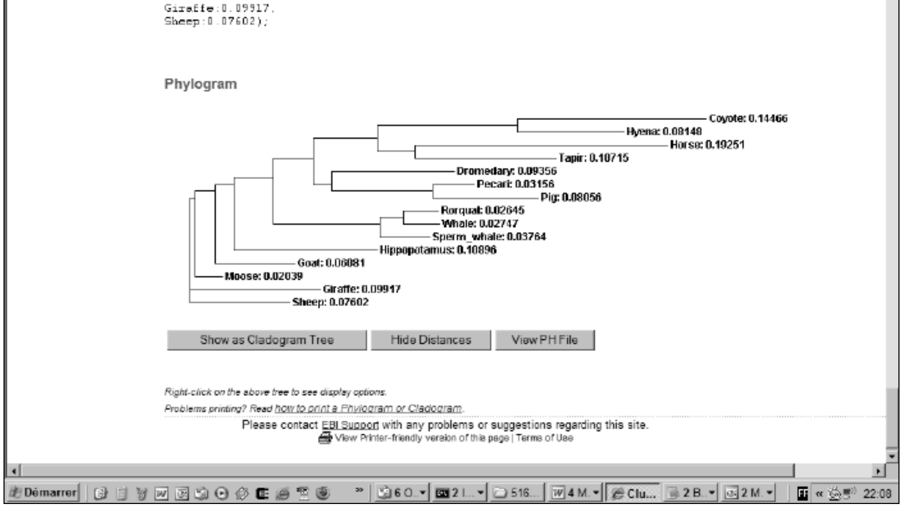
6. **Click the Run button.**

   Wait until ClustalW returns your NJ tree, as shown in Figure 13-5.

   This tree is much more accurate than a guide tree, and this one clearly shows the genetic relationship between the hippopotamus and the whale, as postulated by Higgins and Grauer a few years ago.

7. **Save the graphical representation of your tree.**

   The easiest way to save your tree is to make a screen capture with the print-screen (Prt Sc) key on your keyboard. You can then cut and paste this image into your favorite application (PowerPoint, Paint. etc.).



```
Giraffe:0.09917,
Sheep:0.07602);
```

Right-click on the above tree to see display options.

Problems printing? Read how to print a Phylogram or Cladogram.

Please contact EBI Support with any problems or suggestions regarding this site.

View Printer-friendly version of this page | Terms of Use

Figure 13-5: An NJ tree produced by ClustalW.

8. **Scroll to the bottom of the page and click on View PH File.**

9. **Use the File➪Save As option of your browser to save a text version of your tree.**

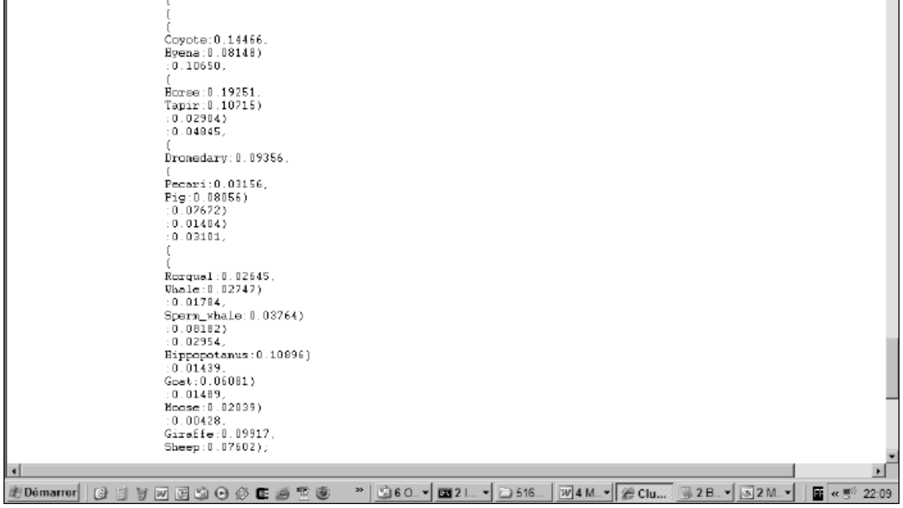
   ClustalW presents your tree in the Newick text format — shown in Figure 13-6 — famous for its many parentheses. When you save the file, you're saving this Newick-formatted version of your tree.

   The Newick format is the most standard way to store phylogenetic trees, and most phylogenetic packages can use this format. If you want to keep only one file related to your tree, keep this one!

Making a tree with Phylip

When it comes to phylogenetic trees, Phylip is a household name. Along with PAUP, it is one of the most widely used resources for computing highly accurate phylogenetic trees.

One of the nice things with Phylip is that it makes it easy for you to *bootstrap* your tree (check its reliability; see the sidebar, "What is bootstrapping?"). What that does for you is make sure that the topology you're looking at is well supported by your multiple sequence alignment.



```
(
(
(
Coyote:0.14466,
Hyena:0.08148)
:0.10650,
(
Horse:0.19251,
Tapir:0.10715)
:0.02904)
:0.04845,
(
Dromedary:0.09356,
(
Pecari:0.03156,
Pig:0.08056)
:0.07672)
:0.01404)
:0.03101,
(
(
Rorqual:0.02645,
Whale:0.02747)
:0.01784,
Sperm_whale:0.03764)
:0.08182)
:0.02954,
Hippopotamus:0.10896)
:0.01439,
Goat:0.06081)
:0.01489,
Moose:0.02039)
:0.00428,
Giraffe:0.09917,
Sheep:0.07602);
```

Figure 13-6: The Newick tree format.

If you decide to become a specialist, Phylip has all you need; you can use it to build almost any kind of tree with any method you like. To demonstrate, here's a set of steps that shows how to produce a phylogenetic tree with bootstrap information:

Before going onto this server, you must have a high-quality multiple sequence alignment ready. For the purpose of these steps, we have prepared one for you, available from `www.tcoffee.org/dummy_aln2.html`.

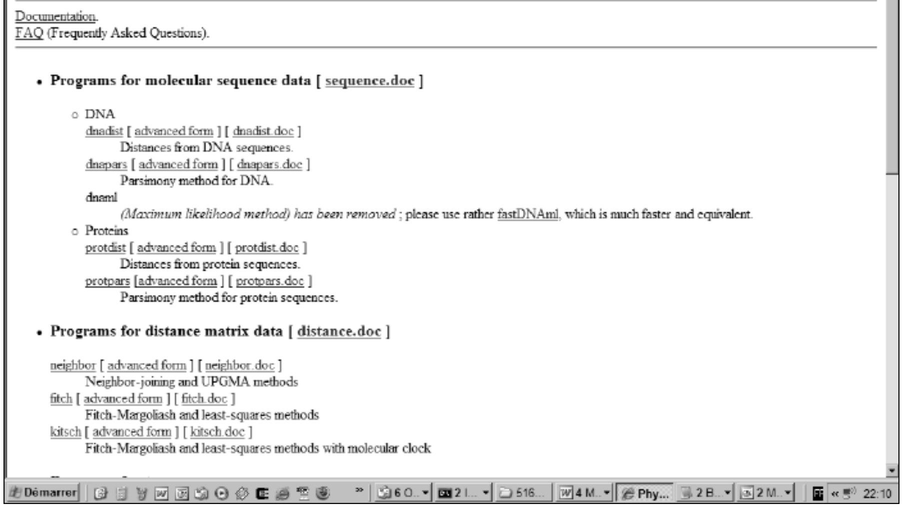
1. **Point your browser to** `bioweb.pasteur.fr/seqanal/phylogeny/phylip-uk.html`**.**

   The Phylogeny: Phylip Programs page appears, as shown in Figure 13-7. This page shows all the Phylip programs installed on this server. You can produce a phylogenetic tree by using these programs one after the other. The only trick is to use them in the right order.

2. **Under the Proteins heading, right next to the** protdist **link, click the** advanced form **link.**

   The Phylip: Protdist page appears. Here's what you need to know:

   - To make a distance-based tree, you first need to generate a distance matrix that contains the pairwise distances between sequences, as measured on the multiple sequence alignment.



Documentation.
FAQ (Frequently Asked Questions).

- **Programs for molecular sequence data [sequence.doc]**
  - DNA
    dnadist [advanced form] [dnadist.doc]
    Distances from DNA sequences.
    dnapars [advanced form] [dnapars.doc]
    Parsimony method for DNA.
    dnaml
    (Maximum likelihood method) has been removed ; please use rather fastDNAml, which is much faster and equivalent.
  - Proteins
    protdist [advanced form] [protdist.doc]
    Distances from protein sequences.
    protpars [advanced form] [protpars.doc]
    Parsimony method for protein sequences.
- **Programs for distance matrix data [distance.doc]**

neighbor [advanced form] [neighbor.doc]
Neighbor-joining and UPGMA methods
fitch [advanced form] [fitch.doc]
Fitch-Margoliash and least-squares methods
kitsch [advanced form] [kitsch.doc]
Fitch-Margoliash and least-squares methods with molecular clock

Figure 13-7: Phylip online at the Pasteur Institute.

- Some methods enable you to measure pairwise distances on DNA or on proteins. There are also different programs for closely or distantly related sequences.
- Protdist is a method that uses a substitution matrix to measure the distance between your aligned sequences.
- Protpars does the same thing, but it tries to evaluate the number of mutations at the DNA level required for the observed amino-acid differences. Protpars works better with closely related sequences.

3. **Enter your e-mail address in the Your E-Mail field of the Phylip: Protdist page.**

   Again, a little briefing is in order:

   - If the server is overloaded or if you are analyzing a large alignment, Phylip sends you your results by e-mail rather than giving them to you online.
   - The e-mail you receive contains a URL. When you open this URL, your browser takes you to a page similar to what's shown in Figure 13-10.

4. **Paste your multiple alignment into the Sequence window, as shown in Figure 13-8.**

   Do not forget to copy the header line that starts with CLUSTALW along with your alignment.

5. **Click the** Bootstrap options **link.**

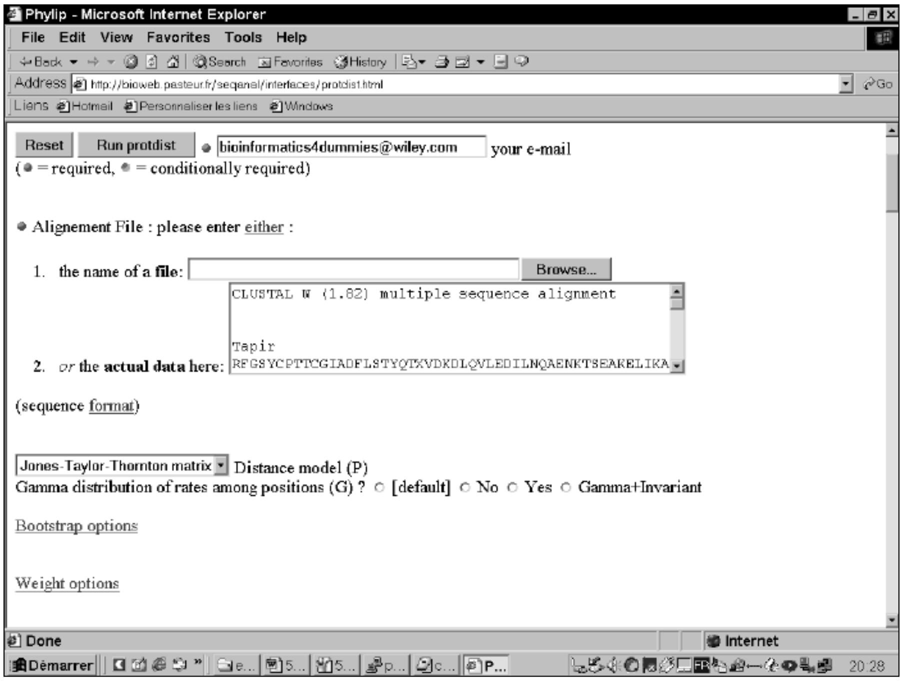
   You are brought to the Bootstrap Options section of the Web page.

6. **Select the Perform a Bootstrap Before Analysis check box, as shown in Figure 13-9.**

   Bootstrapping your data is a way of assessing the quality of your tree; see the "What is bootstrapping?" sidebar for details.

7. **Set the seed to some odd number such as *1* or *3*.**

   The *seed* controls the generation of random numbers during the bootstrap. Different seeds lead to different results.


Phylip - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bioweb.pasteur.fr/seqanal/interfaces/protdist.html
Go
Liens Hotmail Personnaliser les liens Windows
Reset
Run protdist
Browse...
Done
Internet
Démarrer
20:28

● bioinformatics4dummies@wiley.com your e-mail
(● = required, ● = conditionally required)

● Alignement File : please enter either :

1. the name of a file:
2. *or* the actual data here:

```
CLUSTAL W (1.82) multiple sequence alignment


Tapir
RFGSYCPTTCGIADFLSTYQTKVDKDLQVLEDILNQAENKTSEAKELIKA
```

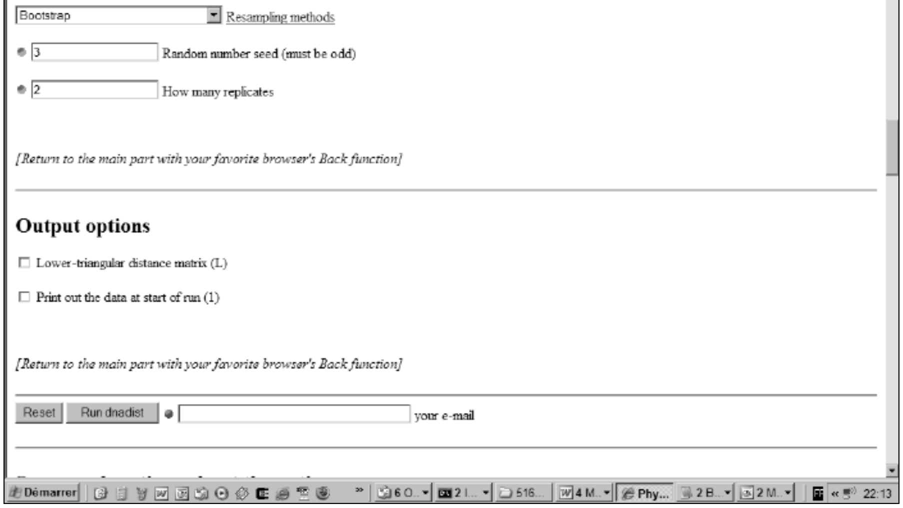
(sequence format)

Jones-Taylor-Thornton matrix Distance model (P)
Gamma distribution of rates among positions (G) ? ○ [default] ○ No ○ Yes ○ Gamma+Invariant

Bootstrap options

Weight options

Figure 13-8: Protdist server.



Bootstrap ▾ Resampling methods

● 3 Random number seed (must be odd)

● 2 How many replicates

[Return to the main part with your favorite browser's Back function]

Output options

□ Lower-triangular distance matrix (L)

□ Print out the data at start of run (1)

[Return to the main part with your favorite browser's Back function]

Reset | Run dnadist | ● your e-mail

Figure 13-9: The Bootstrap options of protdist.

8. **Keep the number of replicates to *100.***

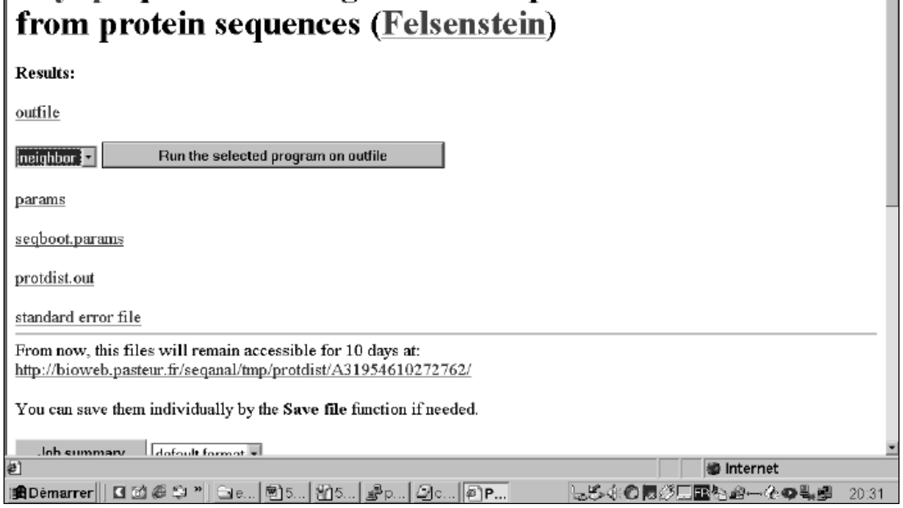
   The number of replicates is the number of bootstrap cycles that you do. Normally, this number should be at least 100.

   Remember how many replicates you request as you will need to re-enter this figure later!

   For real analysis, you need at least 100 replicates, but if you only want to get a feeling of how the server works, you may try values as small as 2. If you do so, it will be fine for the computation but don't expect the bootstrap to be biologically meaningful! For the purposes of this tutorial, we will use 2, so that the results get computed interactively, but if you require a large number of replicates, the server usually prefers returning the results by e-mail.

9. **Scroll back to the top of the page and click the Run Protdist button.**

   Eventually, you'll see results similar to what's shown in Figure 13-10. If your browser times out, the server will send you an e-mail that contains a URL. If you open this URL, you're taken to a page of results similar to what's shown in Figure 13-10.



from protein sequences (Felsenstein)

Results:

outfile

params

seqboot.params

protdist.out

standard error file

From now, this files will remain accessible for 10 days at:
http://bioweb.pasteur.fr/seqanal/tmp/protdist/A31954610272762/

You can save them individually by the **Save file** function if needed.

Figure 13-10: Protdist results.

10. **Choose Neighbor from the first drop-down menu on the result page.**

    Neighbor is a program that turns your distance matrix into a Neighbor Joining tree.

    If you want, you can use another tree-reconstruction method (such as Fitch, which uses the mean-least-square method). These days, people usually agree that NJ provides a very good trade-off between available methods.

11. **Click the Run the Selected Program on Outfile button.**

    The Neighbor page appears.

12. **Select the type of tree you want to produce on the line that says Distance Method. (Neighbor Joining should be selected by default.)**

    Neighbor Joining and UPGMA are related methods, but Neighbor Joining produces trees that are much more accurate.

    UPGMA produces rooted trees, where the root is an attempt to guess the position of the common ancestor (See Figure 13-14).

    Neighbor-Joining does not try to guess where the common ancestor lies and produces unrooted trees that you can root later.

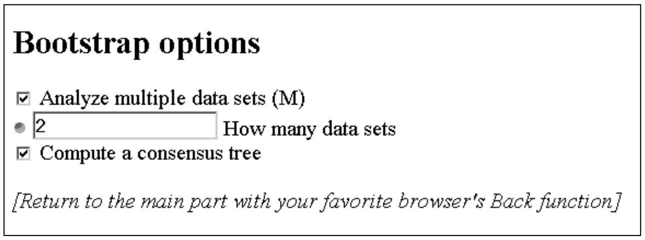
13. **Select the** Bootstrap options **link.**

gram how many cycles it has done!

14. **Select the Analyze Multiple Data Sets option.**

15. **Indicate the number of data sets you're providing, as shown in Figure 13-11.**

    Enter 100 if you have entered 100 in Step 8, or any number you have used in that step.



Bootstrap options

☑ Analyze multiple data sets (M)

● 2 How many data sets

☑ Compute a consensus tree

[Return to the main part with your favorite browser's Back function]

Figure 13-11: The Bootstrap option of Neighbor.

16. **Set the Random Seed to an odd number (such as 1 or 3), and check the Randomize box.**

    No need to use the same number as in Step 7.

17. **Check the Compute a Consensus Tree check box.**

    This option tells Neighbor to return the tree that has the best agreement with the replicate data sets you are providing.

    This option makes sense only if you are using the bootstrap and have chosen values larger than 1 in Steps 8 and 15 (multiple datasets).

18. **Scroll down to the Other Parameters section and choose an outgroup.**

    Here are some handy tips about outgroups:

    - Choosing an outgroup only matters if you selected Neighbor Joining in Step 10.
    - You can specify which outgroup you want to use according to its position in the multiple alignment, and Neighbor uses this outgroup as a root.
    - In your tree, the *root* represents the common ancestor of all the sequences. (Remember, however, that living fossils do not exist!) This outgroup is not the real root, but the real root is necessarily between the outgroup and its closest node.

19. **Scroll to the top of the page and click the Run Neighbor button.**

    Your results appear in a new page as a series of hyperlinks.

20. **Open the consensus tree files.**

    The two first files on the output list are the consensus tree files. They're hyperlinked, so you can simply click them to display their contents.

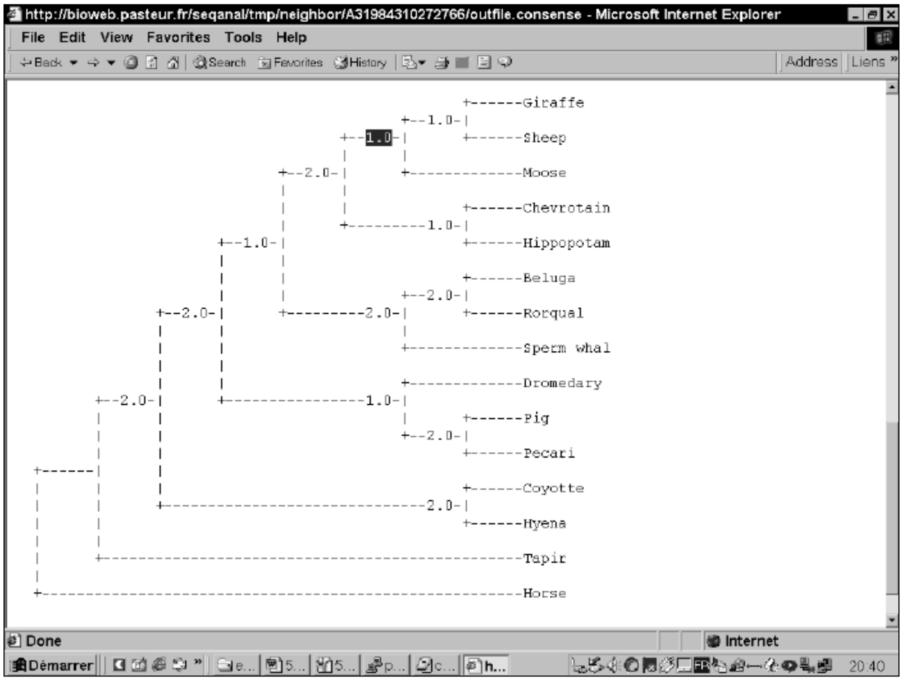
    `outfile.consense` (see Figure 13-12) contains a text version of your consensus phylogenetic tree.

    The consensus tree is always unrooted. The number above each branch indicates its stability, with 50 meaning more stable and 20 meaning less stable.

    `outtree.consense` is the same tree but in the Newick format.

21. **Open the normal trees.**

    Neighbor also outputs the trees it used to build the consensus in `outfile` and `outtree`.


http://bioweb.pasteur.fr/seqanal/tmp/neighbor/A31984310272766/outfile.consense - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address Liens
Done
Internet
Démarrer
20:40

```
                                                 +------Giraffe
                                          +--1.0-|
                                   +--1.0-|      +------Sheep
                                   |      |
                            +--2.0-|      +-------------Moose
                            |      |
                            |      |             +------Chevrotain
                            |      +---------1.0-|
                     +--1.0-|                    +------Hippopotam
                     |      |
                     |      |                    +------Beluga
                     |      |             +--2.0-|
              +--2.0-|      +---------2.0-|      +------Rorqual
              |      |                    |
              |      |                    +-------------Sperm whal
              |      |
              |      |                    +-------------Dromedary
       +--2.0-|      +--------------1.0-|
       |      |                           |      +------Pig
       |      |                           +--2.0-|
       |      |                                  +------Pecari
+------|      |
|      |      |                                  +------Coyotte
|      |      +------------------------------2.0-|
|      |                                         +------Hyena
|      |
|      +------------------------------------------------Tapir
|
+-------------------------------------------------------Horse
```

Figure 13-12: A consensus tree output by Neighbor.

- `outtree` contains the trees in Newick format.

22. **Save and re-display your output.**

    It is difficult to display a tree with bootstrap values using online tools. If you want to keep these values, the best thing to do is to save the `outfile.consense` file into a tree. (This is a standard text file that you can save by using your browser's File➪Save As option.)

    If you want to redisplay your tree using another graphic format, the most convenient thing to do is to save the file `outtree.consense` and to turn it into a GIF or a PDF file with the Phylodendron server. (See the last part of this section for more on how to do this.)

What is bootstrapping?

Just because your computer can produce a tree doesn't necessarily mean that the tree is correct. Many things can go wrong in the tree's construction — including faulty data or an incorrect alignment. In most cases, your tree will be globally correct but contain a few inaccurate branches. This is where *bootstrapping* comes in handy. Tree bootstrapping is a way to check whether your tree is biologically meaningful by assessing its *robustness* — that is, checking whether every portion of the alignment equally supports your tree. The logic behind bootstrapping is that you measure the solidity of each branch in your tree so as to determine whether you can trust it when you draw your conclusions.

Bootstrapping is a multiple-step process. The program starts by sampling columns in your initial alignment. The purpose is to generate a *bootstrap alignment* in which some columns are missing and others are duplicated. Here's how it's done:

1. **Choose a column at random from the initial multiple alignment obtained with your sequences.**

   This column is the first column of your *bootstrap alignment.*

2. **Select a new column in the initial alignment and copy it to extend the bootstrap alignment.**

   By *extend* here, we mean making the alignment one column longer than the column you just copied from the initial alignment.

3. **Repeat Step 2 until your bootstrap alignment contains as many columns as your initial alignment.**

   Note that some of the columns are either never selected or appear several times in the final bootstrap alignment. This is *random sampling.*

(continued)

```
Initial Alignment
Column 1 2 3 4 5 6 7 8 9
seq1   A B C D E F G H I
seq2   A A B B C B A C A
seq3   C C A C B A C A B

Bootstrap Alignment 1
1 1 8 1 2 5 1 8 2
A A H A B E A H B
A A C A B C A C A
C C A C C B C A C

Bootstrap Alignment 2
1 4 5 6 6 3 4 1 7
A D E F F C D A G
A B B C C B B A A
C C B A A A C C C
```

Note that the first bootstrap alignment contains Column #1 of the initial alignment four times.

This procedure generates many multiple alignments that look more or less like the original. The purpose of all this is to check whether all the columns in your initial alignment tell a similar evolutionary story (that should be the case).

During the next step, each random alignment is turned into a distance matrix, and each matrix is turned into a tree. To build the consensus tree, Phylip takes the average of all the trees it has generated from the bootstrap alignments. To assess the quality of each branch in the consensus tree, Phylip counts the number of bootstrap trees that contain this branch. Good branches are those that appear in every

13-12. On this consensus tree, every branch comes along with a number between 1 and 2. This value tells you how solid your branch is. This value falls between 1 and 2 because, in Steps 8 and 15, we chose to have 2 bootstrap cycles. Neighbor generated 2 bootstrap trees with these 2 bootstrap alignments, and it generated a consensus tree that's the average of these 2 bootstrap trees.

In a tree, a branch always separates your data into two groups: sequences on the left and sequences on the right side of the branch. The numbers above the branches in the consensus tree indicate how many branches exist in your two trees that split the data in exactly the same way as the branch you're looking at.

For instance, if you look at the consensus tree in Figure 13-12, you find that the branch containing giraffe and sheep has a value of only 1. The reason for this is that in one of the bootstrap trees, giraffe and moose are in the same group — while in the other tree, giraffe is in the same group as sheep (and moose is alone). In theory, you could conclude that the tree indicates some uncertainty on whether sheep and giraffe are more closely related to one another than each is related to moose. In practice, however, with only 2 bootstraps, you really can't say anything of the kind! You need at least 100 bootstrap cycles before you start convincing farmers to breed giraffes for their wool.

Making a maximum likelihood tree with PhyML

Maximum likelihood trees are considered to be more accurate than other trees because they produce the tree that is most likely (statistically speaking) to explain your alignment. In other words, your alignment is a little story that explains how, starting from one ancestral sequence, a series of mutations

University, France) it is now possible to compute these trees in a much more realistic amount of time. We show you below how to use their PhyML server.

1. **Open the dummy alignment at** `www.tcoffee.org/dummy_aln3.html`**.**

   This alignment is in Phylip format, the only format recognized by the server.

2. **Use the File➪Save As option of your browser to save the alignment as a text file.**

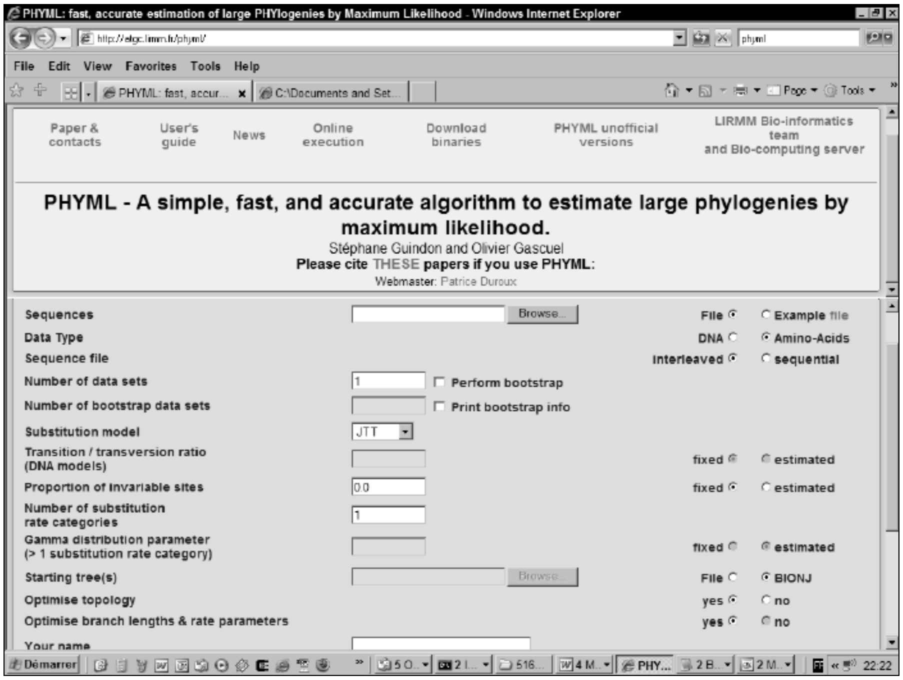
3. **Point your browser to** `atgc.lirmm.fr/phyml/`**.**

4. **Select the File radio button, as shown in Figure 13-13.**

   You can now upload your file with the Browse button.

5. **Select the Amino Acids radio button.**

   Our alignment contains amino acids.


PHYML: fast, accurate estimation of large PHYlogenies by Maximum Likelihood - Windows Internet Explorer
http://atgc.lirmm.fr/phyml/
phyml
File Edit View Favorites Tools Help
PHYML: fast, accur...
C:\Documents and Set...
Page
Tools
Paper & contacts
User's guide
News
Online execution
Download binaries
PHYML unofficial versions
LIRMM Bio-informatics team and Bio-computing server
Démarrer
5 O...
2 I...
516...
4 M...
PHY...
2 B...
2 M...
22:22

PHYML - A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood.

Stéphane Guindon and Olivier Gascuel

Please cite THESE papers if you use PHYML:

Webmaster: Patrice Duroux

Sequences	Browse...	File ●	○ Example file
Data Type		DNA ○	● Amino-Acids
Sequence file		Interleaved ●	○ sequential
Number of data sets	1 □ Perform bootstrap		
Number of bootstrap data sets	□ Print bootstrap info		
Substitution model	JTT		
Transition / transversion ratio (DNA models)		fixed ●	○ estimated
Proportion of invariable sites	0.0	fixed ●	○ estimated
Number of substitution rate categories	1		
Gamma distribution parameter (> 1 substitution rate category)		fixed ○	● estimated
Starting tree(s)	Browse...	File ○	● BIONJ
Optimise topology		yes ●	○ no
Optimise branch lengths & rate parameters		yes ●	○ no
Your name			

Figure 13-13: Setting up the PHYML server.

Number of Bootstrap Data Sets field.

Performing a bootstrap does take more time, but it is a crucial step if you want to reliably estimate the biological meaning of your tree.

In practice, it's better to require 1,000 bootstraps — but for this example to run, we have set the value to 100.

8. **Scroll down to fill in the Name and E-Mail Address fields.**
9. **Click Run.**

When we did a trial run of this example, it took about half an hour for our results to come via e-mail — so be prepared for a bit of a wait. When an e-mail arrives, it contains a tree in Newick format that you can visualize using Phylodendron (we explain how to do this later in this chapter.) The tree will come with bootstrap values indicating how much trust you can have in each node. The higher the bootstrap value, the more reliable the node.

Knowing what's what in your tree

When you work with phylogenetic trees, there's always a bit of special vocabulary involved. Figure 13-14 recapitulates the names of the various components in your tree. Some of these names may sound a bit technical, but it's worth knowing their meaning because phylogenetic jargon is becoming more and more common in mainstream publications.

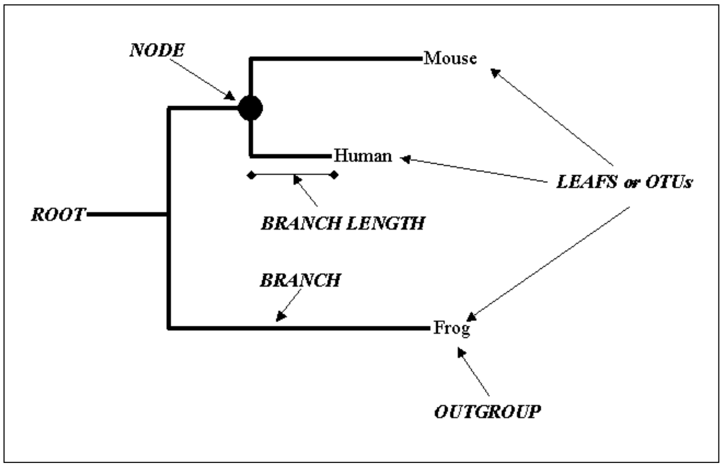


Figure 13-14: Various components of a phylogenetic tree.

tree. A tree can be *scaled* (in which case the branch length means something in terms of evolutionary time) or *unscaled* (in which case only the topology of the tree is informative).

The tree may be rooted or unrooted. *Unrooted* trees do not identify evolutionary paths. If your tree is unrooted, you accept the possibility that any of the nodes or OTUs could be closer to the common ancestor than to any of the other OTUs. If you have an *outgroup,* you know that the root must be inserted in the branch between this outgroup and the other clades.

REMEMBER

The root of a tree corresponds to the most ancient common ancestor. This sounds good, but it is (unfortunately) meaningless in terms of the kinds of trees we're reconstructing here. The problem is that tree-reconstruction methods have no way of telling the direction of evolution. They cannot be used to determine where the most ancient node is. As a consequence, the root you may observe on some trees is usually arbitrary — therefore meaningless from a biological point of view. Biologists like to use what they call *unrooted* trees, to avoid any confusion. Yet, if you need a root, the only proper way to insert it into your tree is to use what gurus call an *outgroup* — an organism so distantly related to the others that it can safely be used as a root. For instance, if you make a tree involving primates, you should include the mouse (with the exception of Mickey Mouse, few mice are considered primates) so you can use it as an outgroup; likewise, if you are comparing mammals, you should include a reptile, and so on.

Displaying your phylogenetic tree

Phylodendron is a powerful Web-based server that lets you turn a text file in Newick format into a graphic display of your tree. You can control every minute aspect of the display and generate images in any format you fancy. In the following steps, we show you how to use Phylodendron to turn a phylogenetic tree into a GIF picture that you can include in any document:

1. **Point your browser to** `iubio.bio.indiana.edu/treeapp/treeprint-form.html`.

   A similar service runs on the Spanish EMBnet node at `www.es.embnet.org/Doc/phylodendron/treeprint-form.html`.

2. **Paste your tree in Newick format in the Input window.**

5. **If you want horizontal distances to have a meaning, select the Use Node Lengths check box in the Tree Growth section.**

6. **Click the Submit button.**

7. **Choose File➪Save As from your browser's main menu to save your results as a GIF file that you can use for inclusion in other documents.**

Doing Phylogeny for Free over the Internet

Conducting phylogeny over the Internet isn't yet as developed as database research or other kinds of sequence analysis methods. Fortunately, over the last few years, the community has made some high-quality resources available. These resources might not run on very powerful servers, but they're perfectly designed if all you need to do is build a tree once in a while.

Finding online resources

A vast amount of phylogeny resources are available over the Internet. Yet most of these programs aren't available online; you need to install them on your own computer. Table 13-2 lists a few resources that you can use online.

Table 13-2 Online Sites for Making Phylogenetic Trees

Address	*Description*
`www.ebi.ac.uk/clustalw/`	You can use ClustalW to build multiple alignments and compute NJ trees. Remember: You cannot do both at the same time!
`www.genebee.msu.ru/clustal/basic.html`	The Genebee server can produce genuine phylogenetic trees in one step.
`www.tcoffee.org`	Tcoffee computes a genuine NJ-phylogenetic tree in one step.

atgc.lirmm.fr/phyml/	A powerful method to compute maximum likelihood trees from Gascuel and his team.
bioweb.pasteur.fr/seqanal/interfaces/bionj-simple.html	An interface to BioNJ, a novel NJ method.
www.up.univ-mrs.fr/evol/figenix/	A powerful Java tool to gather members of a protein family and build the associated tree.
bioweb.pasteur.fr/seqanal/phylogeny/phylip-uk.html	A Web interface for Phylip.
www.genebee.msu.ru/services/phtree_reduced.html	Very powerful interface for a novel tree-reconstruction method.

Finding generic resources

Table 13-3 lists some of the most important resources in phylogeny. In most situations, you should be able to find what you need a few clicks away from these pages. We also include two resources that should cover your immediate need if you want to find out the basics of phylogeny for yourself.

Table 13-3 Generic Phylogenetic Resources on the Internet

Address	*Description*
evolution.genetics.washington.edu/phylip/software.html	Joe Felsenstein's pages, where Phylip lives; it's also one of the most extensive collections of resources available. Truly a legendary site!
www.ucmp.berkeley.edu/subway/phylo/phylosoft.html	A very complete list of phylogeny resources.
paup.csit.fsu.edu/index.html	The home of PAUP, legendary phylogeny package using Parsimony. Although PAUP is a commercial package, it's reasonably priced and worth every penny, according to specialists.

(continued)

phylo.html	
www.techfak.uni-bielefeld.de/bcd/Curric/MathAn/mathan.html	A high-quality course on tree reconstruction methods.

Collections of orthologous genes

Collections of orthologous or homologous genes are very useful if you want to ask functional or phylogenetic questions. Table 13-4 lists some databases that provide this type of information.

Table 13-4 Collections of Orthologous Sequences

Address	*Description*
www.ncbi.nlm.nih.gov/COG/	Cluster of orthologous sequences maintained by the NCBI. Each cluster contains proteins from bacterial genomes.
pbil.univ-lyon1.fr/databases/hovergen.html	A collection of orthologous vertebrate genes.
pbil.univ-lyon1.fr/databases/hogenom.html	A collection of orthologous bacterial genes.
systers.molgen.mpg.de	Another collection of homologous sequences.
rdp.cme.msu.edu/, www.psb.ugent.be/rRNA/ssu/, www.psb.ugent.be/rRNA/lsu/	Three extensive collections of ribosomal RNA sequences, which are very useful for classifying new organisms and come with appropriate phylogenetic tools.