A Software Environment for Sequence Data

by Wolfgang Ludwig and Oliver Strunk

Lehrstuhl für Mikrobiologie

Technische Universität München

D-80290 München

Germany

Contents

Chapter 1 Introduction 1
Section 1 Documentation Conventions 1
Section 2 Authors 1
Section 3 The Manual 1
Section 4 The Concept of ARB 2
Section 5 Databases 2
Section 6 Why use ARB 2
Chapter 2 Installing ARB 3
Section 1 Get ARB 3
Section 2 Install ARB 3
Section 3 Sample Beginner's ARB Installation on a Linux Computer 6
Chapter 3 Working with ARB 7
Section 1 Getting started 7
Section 2 Opening an ARB Data File 8
Section 3 Saving an ARB Data File 8
Section 4 The ARB_NT Main Program 9
Section 5 The Sequence Editors 20
Section 6 Working with Sequences 24
Section 7 Phylogenetic Analyses 31
Section 8 Importing Data 38
Section 9 Saving Database 42
Appendix A Behind the Curtain 43
Section 1 Introduction 43
Section 2 Database 43
Section 3 Database Objects 45
Section 4 NDS & SRT/ACI/REG 46
Section 5 PT_SERVER 49
Section 6 General Conception 50
Section 7 Installation 51

Chapter 1 Introduction

1 Documentation Conventions

- Words in *italics* indicate a special meaning of the term.
- Words in **bold** indicate ARB components, windows, subwindows, buttons.
- Words in **`mono-spaced`** indicate commands that are entered from the keyboard.
- Words in wordbox represent a key on the keyboard.
- Press and click mean pressing and releasing a mouse button (after properly positioning the cursor.
- Drag means moving the cursor while pressing a mouse button.

2 Authors

The ARB program package was developed at the
Lehrstuhl für Mikrobiologie
Technische Universität München
D-80290 München
Germany

Design and programming was done by Oliver Strunk, Wolfgang Ludwig, Oliver Gross, Boris Reichel, Norbert Stuckmann, Michael May, Björn Nonhoff, Michael Lenke, Toni Ginhart, Alexander Vilbig, Ralf Westram.

The ARB program package is still under development.

Foreign software tools were integrated. Copyright notes are accessible via on-line help

3 The Manual

This is a preliminary manual and provides limited information on the most frequently used software tools. For further instruction consult the on-line help facilities. Most ARB windows contain **HELP** button(s). Press these buttons to display the **HELP WINDOW**. This window contains information on the current software tool and provides access to help texts an related topics.

4 The Concept of ARB

ARB (from arbor, Latin: tree) was conceived as a graphically oriented package of software tools for establishing, handling and using hierarchical databases of sequences and associated information. The major concept was to combine access to the data via graphically presented hierarchy (tree) and sequence data analysis. The programs have primarily been designed for rRNA data and data analysis, however, can be used for any nucleic or amino acid sequence data.

The data such as sequence, bibliography, identifiers or user provided text are stored in individual *fields* associated with the respective *species*. The term *species* does not necessarily match that of a biological species but indicates a containment for all data assigned to a sequence. The containments can be hierarchically arranged according to sequence similarity (phylogeny).

Any sequence data and/or associated information can be displayed in the **ARB_NT** main window along with a tree reflecting the hierarchy of the data base. The tree can be used as a guide to walk through the data as well as a tool for selecting (combinations of) data base entries for analysis by using other tools of the package (e.g. for searching, editing, modifying, aligning, treeing, profiling).

The software is fully graphic oriented. Any function can be invoked by mouse click. Generally, the left mouse button has to be used. (In cases where the other buttons are effective short advice is given on the screen).

5 Databases

ARB databases of small and large subunit rRNA sequences are periodically updated and provided at our FTP or WWW servers.

6 Why use ARB

There are hundreds of small and big programs for sequence analysis available. So why use ARB ? Here is my personal checklist:

You should use ARB if you are

- designing many oligonucleid — (16s/23s) probes.
- working with hundreds of sequences.
- doing extensive phylogeny.
- working with 16s/23s/18s rRNA sequences.
- not afraid of UNIX/LINUX

Chapter 2 Installing ARB

1 Get ARB

This shows how to install the ARB Package. First of all check our www or ftp servers and download and read the file 'arb_README'. Then download your machine specific program tar files and some demo databases:

- get the files arb_README from www / ftp-server either by

  netscape http://www.mikro.biologie.tu-muenchen.de
  or ftp ftp.mikro.biologie.tu-muenchen.de /pub/arb
  or netscape http://www.biol.chemie.tu-muenchen.de
  or ftp ftp.biol.chemie.tu-muenchen.de

2 Install ARB

- login as root (recommended, but only necessary for linux systems)
- Read the arb_README:

excerpt from arb_README

```
Welcome to the 'ARB' Sequence Database Tools

/*********** Hardware and System Requirements *************/

 ARB is currently developed on SUN workstations and Linux PCs.
 The most recent version is now available for this machines.

 Release dates / history:

  HP Series 7000 June 95
  PC Linux Jan  96 ( 486dx; >16 Mega Byte RAM)
  SGI Irix June 96
  Digital OSF April97
  SUNOS4.x Mai  92
  SUNOS5.x June 94
```

```
 Hardware Requirements:
    Minimum  Good
 Real Memory  32  64-256
 Free Disc Space  100  1000
 Computer Speed  25Spec92 100Spec92
    = 486dx66 =586dx90
    Sparc 1  Sparc 5/10/20

 Note: Memory is more important than a fast processor, a 486dx
  width 64 mByte of RAM may be much faster than an
  Ultra Sparc with 32 mByte of RAM.


/*********** Files needed to install ARB *************/

 File  FTP server location  // Comment

 'arb_README' pub/ARB/arb_README  // this file
 'arb_install' pub/ARB/$MACH/arb_install // install script
 'arb.tar.gz' pub/ARB/$MACH/arb.tar.gz // ARB program
 'zcat'  pub/ARB/$MACH/zcat  // decompress (gzip)
        ['arb_ale.tar.gz'pub/ARB/$MACH/arb_ale.tar.gz // optional Editor ]
        ['***.arb' pub/ARB/data/*.arb  // optional demo /
                 real rRNA data ]

 Notes:
  - $MACH should be replaced by your system type
    ( type uname -sr to find out your system type )
  - enable binary mode for ftp transfer ( command 'bin' )
  - do not uncompress and untar arb.tar.gz directly, use the
    install script !!!

/*********** Install/Update ARB *************/

 ARB consists of more than 750 files which are installed into a single
 directory. Creating this directory, copying all data into it, and setting
 the permissions correctly is done by the installation script
  'arb_install'

 Goto the directory, where the files

  'arb_install'  //install script
  'arb_README'  //this file
  'arb.tar.gz'  //all the libs and bin
  'zcat'   //decompress
       [ 'arb_ale.tar.gz' //optional sequence editor ]

 are located and type '/bin/sh arb_install'

 On Linux computers become root.
 Answer all questions asked by the script.
 Notes: -The script will ask about a pt_server directory. This is a
  directory where arb will store big index files.
  You should enter a different path as you do not want to
```

```
 recreate those files after an ARB update.
      - Normally pressing enter will be a good choice.
      - You can rerun the script many times, it can be used to
 change an existing installation.

Change your .cshrc/.profile files:
 Set the enviroment variable ARBHOME
 to the ARB installation directory
 Append   $ARBHOME/bin   to your PATH

reread it, (logout+login )

goto a directory with a demo database 'eg demo.arb'
and start 'ARB' with

  'arb'
```

3 Sample Beginner's ARB Installation on a Linux Computer

This is a summary of all keyboard commands necessary to install ARB on a Linux computer. Other computers need similar steps:

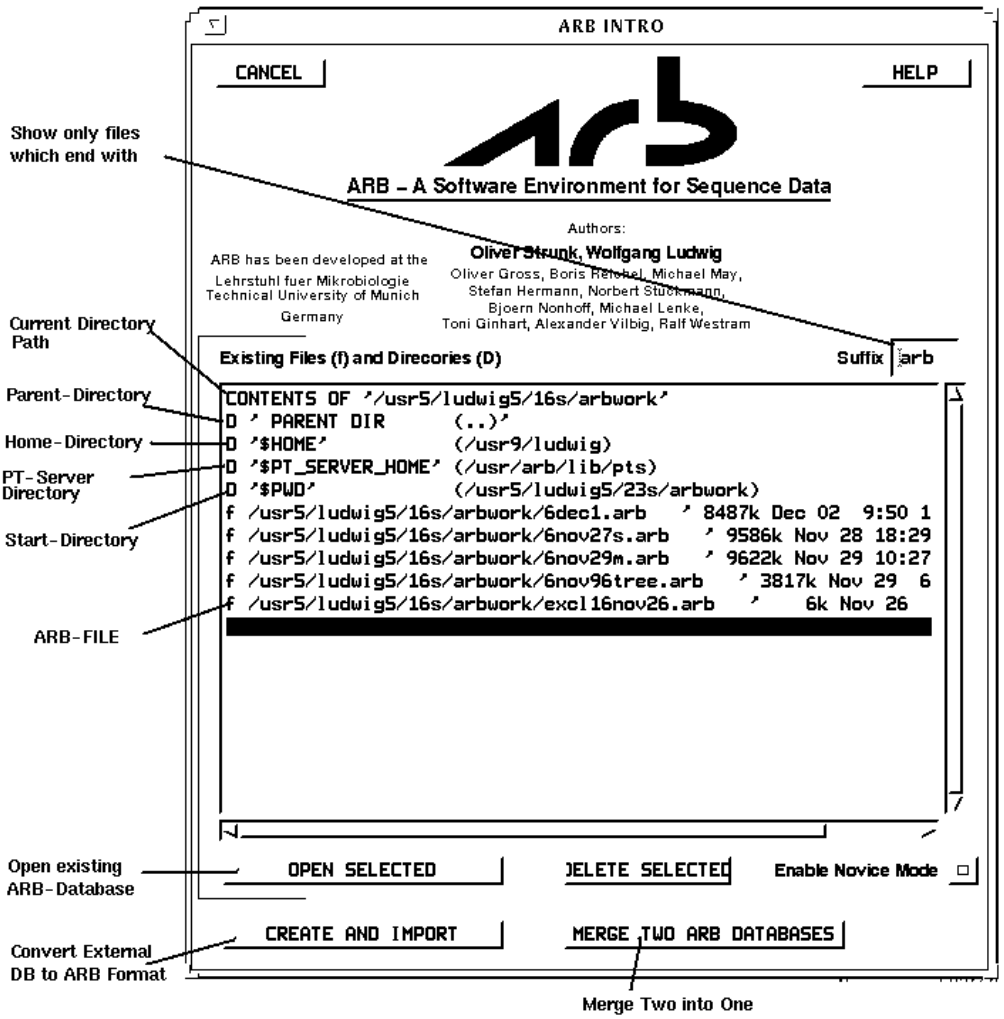
- **login as root** :do this only on linux computers
- **mkdir import** :create import directory
- **cd import** :change current directory to new import dir.
- **ftp 129.187.220.2** :connect to our arb server
- name: **ftp**
- passwd: **your name**
- **bin** :change to binary transfer mode
- **prompt** :disable interactive mode
- **cd ARB**
- **cd linux** :goto source files
- **dir** :check date of files
- **mget *** :download everything
- **cd ../data** : goto data directory
- **mget *** :dowload all datafiles
- **exit** :exit ftp program
- **sh ./arb_install** :start installation script
- **/usr/arb** :path of target installation
- **/usr/pt_server** :path of common data files
- .. everybody may update ..: **y**
- .. trust your users .. **y**
- .. networking.. **s**
- finally: **3**

After starting the X-server (command **startx**) open a command shell, enter arb und be happy.

Chapter 3 Working with ARB

1 Getting started

To start the ARB software type **arb** at a terminal window prompt. The **ARB INTRO** window pops up:



.ARB data files (suffix: **.arb**, marked by **f**) of the current directory (indicated in the first line: **CONTENTS OF .....**) are listed in the **Existing Files (f) and Directories (D)** subwindow. If

desired, move to other directories by mouse click on the lines marked by**D**. The top directories on the list are short cuts to the

- parent directory
- users home directory
- common database and pt_server files directory
- current working directory

folowed by a list of all subdirectories.

2 Opening an ARB Data File

Select an ARB data file by single left mouse click and press the **OPEN SELECTED** button. The main window (**ARB_NT**) appears. The menus and buttons in this window provide access to all other functions of the software package.

3 Saving an ARB Data File

Usually all information of the database is stored into one big file, which is private to each user. At the end of a session, the user has to save all his work to harddisc again. Only explicitly saving makes changes to the database permanent. He can choose between three file formats:

ASCII	A readable ascii representation of the database, which needs an extrem amount of harddisc space. Use this format only in emergency.
BINARY	Normal database format, fast and small
BINARY with FASTLOADFILE	Same as binary, but an additional redundant file is created, which speeds up the database loading time on small computers.

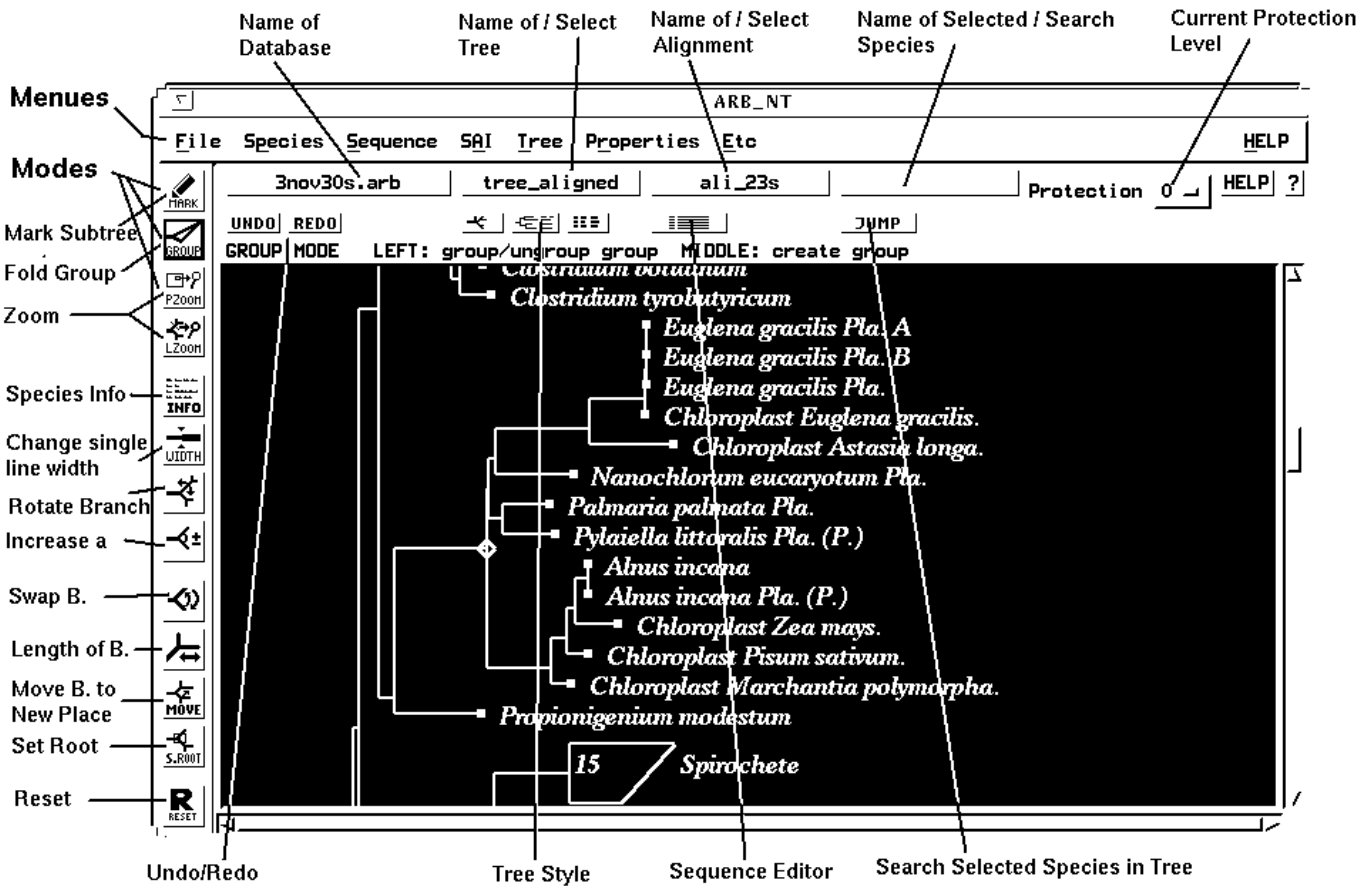
and two types of saving: Normally you cannot do wrong, ARB automatically checks for file

Saving whole database	Create one '.arb' file with all data in it.
Saving changes only	Save only the changes into an extra file.

formats and changes file.

4 The ARB_NT Main Program

The tree displayed in the **ARB_NT** main window gives access to any database entries and allows to select (*mark*) *species* for sequence analyses or processing:



Select a Sequence Type (Alignment)

Different sequence data sets (alignments) e.g. different gene sequences or gene and gene product (protein) sequences assigned to the same *species* can be maintained in an ARB database. Press the button below the **Etc** menu button to bring up the **SELECT AN ALIGNMENT** window and select a dataset from the list. Note: any following operation will be performed on the sequence data stored in the currently selected dataset (alignment).

Select a Tree

The tree to be displayed in the **ARB_NT** main window can be selected from a list displayed in the **SELECT A TREE** window. This window appears after selecting **Select ...** from the **Tree**

menu or pressing the button below the **Tree** menu button. The name of the currently displayed tree is shown in the latter button.

Moving to Selected Species

The *name* of the selected *species* is shown in the button above the **Jump** button. Press the **Jump** button to display that part of the tree which contains the selected *species*.

Tree Layout

Radial tree or dendrogram display mode can be alternatively chosen by pressing the or button, respectively.

The whole tree can be moved by pressing the right mouse button and dragging while the cursor is placed somewhere within the tree display window.

Modifying Tree Layout (Mode Buttons on The Left)

Display Groups

Monophyletic groups (*species* shearing a common root in the currently displayed tree) can be displayed as triangles or four sided figures in a radial tree or dendrogram, respectively. To define those groups press either the

or the

button, position the cursor and use the middle mouse button (as specified in the upper part of the **ARB_NT** main window) to bring up the **ENTER A STRING** window and type a name for the group. This *name* is permanently stored in the database.

The groups can be folded or unfolded pressing the

button, positioning the cursor and using the mouse buttons as specified in the upper part of the **ARB_NT** main window. Further grouping options (**Group All**, **Group All Except Marked** or **Ungroup All**) are accessible from the **Collapse/Expand Tree** submenu which can be selected from the **Tree** menu of the **ARB_NT** main window.

Scaling

Pressing the

button enables you to define a region of the tree to scale up to full size of the tree display window. Use mouse buttons as specified in the upper part of the **ARB_NT** main window

Pressing the

button you can define a subtree to scale up to full size of the tree display window. Use mouse buttons as specified in the upper part of the **ARB_NT** main window.

The original status can be restored by selecting **Reset Logical Zoom** or **Reset Physical Zoom** from the **Etc** menu of the **ARB_NT** main window.

Defining the Root

The position of the root (indicated by a square) can be changed after pressing the

button, positioning the cursor and using mouse buttons as specified in the upper part of the **ARB_NT** main window.

Branch Swapping

The relative order of adjacent branches can be changed by pressing the button, positioning the cursor and using mouse buttons as specified in the upper part of the **ARB_NT** main window.

Rotate Branches

The angles between branches in radial trees can be changed gradually or by mouse dragging. After pressing the button or the button, position the cursor and use the mouse buttons as specified in the upper part of the **ARB_NT** main window.

Finishing, Printing and Exporting Trees

The displayed tree can be exported as Newick formate, printed as it is or further modified using a (foreign) drawing program integrated into the ARB package.

To export the tree, bring up the **TREE ADMIN** window by selecting **Copy_Delete_Rename_Export_Import...** from the **Tree** menu. Select a tree from the list in the **TREE ADMIN** window and press the **EXPORT** button. Select or type a file name to the **TREE SAVE** window.

For printing the displayed tree, select **Print & Export** from the **Tree** menu and **Print Tree View to Printer** from the appearing submenu. The **PRINT GRAPHIC** window pops up. Define whether the whole tree should be printed or only the part of that tree which is displayed on the screen by pressing the respective button in the upper part of the window. Similarly, define whether *handles* (squares indicating root and *marked species* should be printed. Select **Orientation:** of the pages, **Magnification%** and press the **Get Graphic Size** button to calculate the number of printed pages which can be seen in the **Pages** subwindows. Select whether the data should be directly printed or stored (optionally previewed) as postscript file by pressing the corresponding button in the lower part of the window. File name and default printer can be defined by typing to the subwindos in the lower right part of the window.

Note: it depends on the memory of the printer whether large trees can be printed on multiple pages.

For further modifying of the tree, select **Edit Tree View using Xfig** from the **Tree**. The **EXPORT TREE TO XFIG** window pops up. Define whether the whole tree should be exported to file or only that part of the tree which is displayed on the screen by pressing the respective button in the upper part of the window. Similarly, define whether *handles* (squares indicating root and *marked species*) should be shown. Select or type a file name in the **Directories (D) and Files (f)** or **File Name** subwindows, respectively. Press the **GO XFIG** button to start the drawing facility.

Note: There are a few computer systems, where you have to install the xfig program yourself: Digital OSF, Linux, HP, SGI.

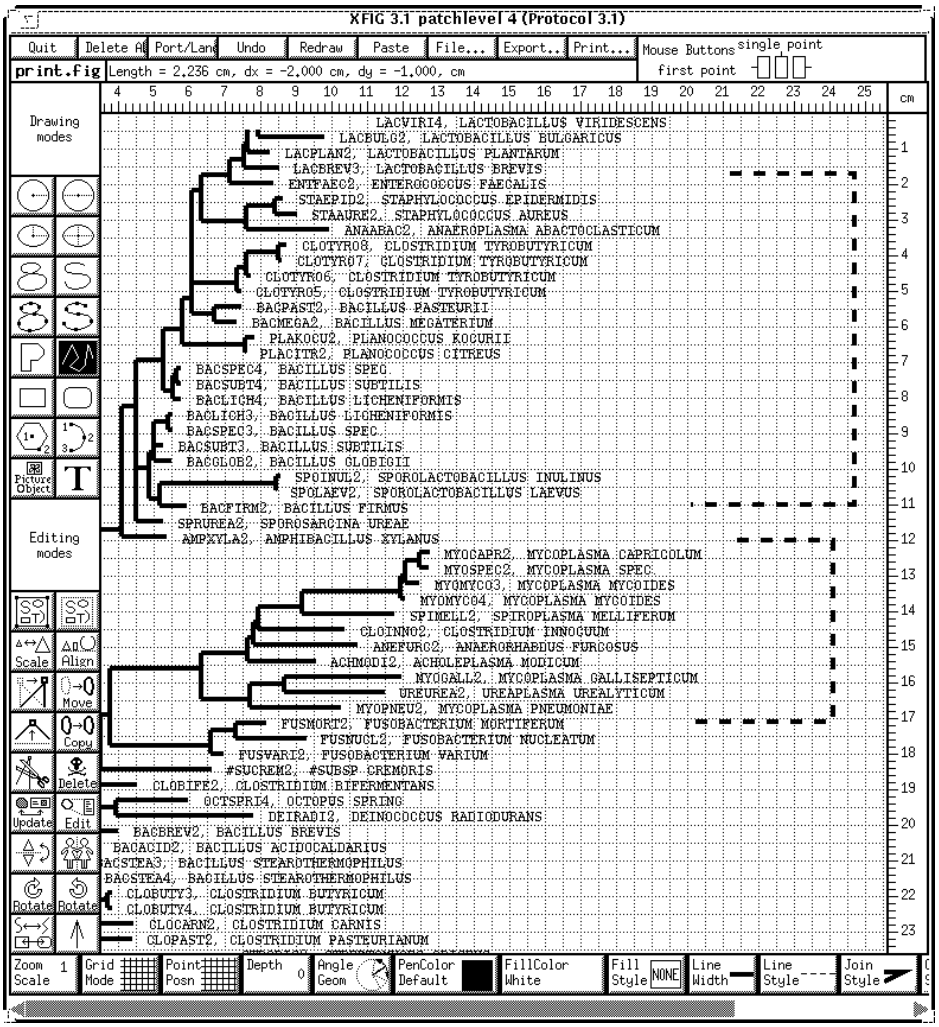


Figure 1 Sample of the xfig drawing program

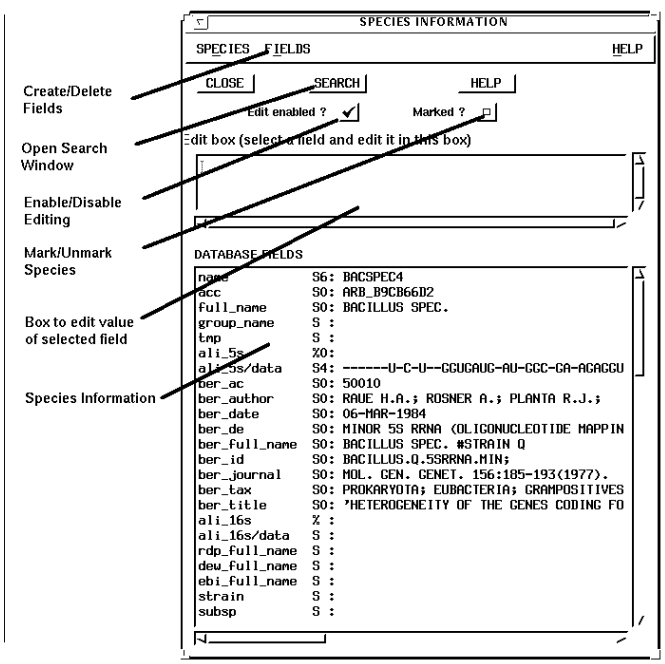
Viewing Database Entries

There are two possible ways to view the database entries: i. display all **field** entries of one **species** in the **SPECIES INFORMATION** window; ii. display a selection of fields for all or a selection of species in the main (**ARB_NT**) window.

Viewing All Fields of one Species

The **SPECIES INFORMATION** window appears after either pressing the

button on the left vertical panel of the main (**ARB_NT**) window, or pressing the **INFO** item in the **Species** menu, or automatically after selecting a species in the **Search and Query** window:



Database Searching

The **SEARCH and QUERY** panel locates strings of text in database *fields*. Select the **Search** option from the **Species** menu of the **ARB_NT** main window or press the **SEARCH** button of the **SPECIES INFORMATION** window to activate the panel.

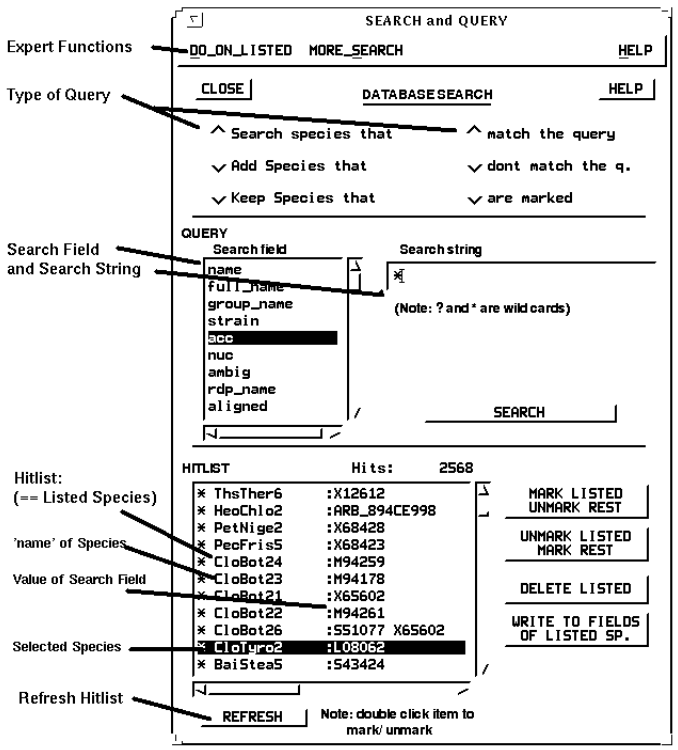


Figure 2 Database search engine

Select a **field** from the **Search field** subwindow and type a string in the **Search string** subwindow. Use `?` and `*` as wild cards for single and multiple characters, respectively.

Database searching is started by pressing the **SEARCH** button as it is by pressing the RETURN key while positioning the cursor in the **Search string** subwindow.

The list of *species* with matching *field* entries is displayed within the **HITLIST** subwindow. Short **name**s and the selected *field* entries are given in the left and right columns, respectively. Alternative *field* entries are displayed after selecting another **Search field** and subsequently pressing the **REFRESH** button. The **HITLIST** can be modified according to the results of further database searches by selecting appropriate combinations of the buttons in the left and right columns of the upper region of the **SEARCH and QUERY** window.

Special search functions are accessible via the items of the **MORE SEARCH** menu. A **HIT LIST** of *species* shearing completely identical entries in the selected *field* is displayed after selecting **Search for Identical Fields ...** while selecting **Search for Identical Words ...** the **HIT LIST** contains *species* shearing single words in their *field*s. Selecting **Search Next Relatives**

... results in a scored **HIT LIST** of species with highest sequence similarity. Note that a correct alignment is not needed to use the latter option.

Defining Selections of *species*

To perform operations on a selection of *species* such as sequence editing, treeing, modifying database entries a set of desired *species* has to be defined by *mark*ing them. This can be done in several windows: the **ARB_NT** main window, the **SPECIES INFORMATION** window, the **ARB_EDIT** window and the **SEARCH and QUERY** window.

To *mark* / un*mark species* in the **ARB_NT** main window invoke the **MARK MODE** by pressing the

button in the left vertical bar and position the cursor to the respective terminal tree node. Press one of the mouse buttons as indicated above the tree display area. There are also *mark*ing options available from the **Species** menu.

To *mark* / un*mark* the *species* edited in the **SPECIES INFORMATION** window press the **MARK** button.

To *mark* / un*mark species* in the **ARB_EDIT** window double click on the respective *name* or use the options available from the **Edit** menu.

To *mark* / un*mark species* in the **SEARCH and QUERY** window double click on the respective *name* or use the appropriate buttons right to the **HIT LIST** window or use the options available from the **DO_ON_LISTED** menu.

Saving Selections of *species*

To store the selection of species under a user defined name, press the **Species/Create Selection from Marked Species**, enter a name and press ok. This selection may be either used to be viewed using the new **ARB_EDIT4** editor (**Sequence/Edit Sequences of Old Selection**), or could be extracted by **Species/Extract Marked Species from Selection**.

Creating, Deleting, Rearranging *Species fields*

Species fields can be created and deleted or their relative order rearranged by selecting **Create ...**, **Delete ...** and **Reorder ...** from the **FIELDS** menu from the **SPECIES INFORMATION** window.

The *fields* can be completely deleted from the database or only their listing be suppressed by choosing the appropriate buttons of the **DELETE FIELD** window which appears after selecting **Delete ...** from the **FIELDS** menu.[1] .

For creating a new *field* type and name of that *field* have to be defined in the **CREATE A NEW FIELD** window which appears after selecting **Create ...** from the **FIELDS** menu.

[1] Note: If some fields have a high security level, you may run into secruity violation while trying to delete them. To overcome this limitation increase your own protection level until it is higher than the highest level of a field to delete.

Modifying Database (*field*) Entries

Database *field* entries can be modified for a selection of *species* or individually.

Field of Individual Species

For modifying *field* entries of individual *species* bring up the **SPECIES INFORMATION** window , ensure that editing is enabled and select a *field* from the **DATABASE FIELDS** subwindow. The *field* entry is displayed in the **EDIT Box** and can be modified by typing to this box.

Set Field of Selection of Species

For modifying *field* entries of a selection of (all) species display this selection in the **SEARCH and QUERY** window and use the **WRITE TO FIELDS OF LISTED SP.** button or the **Modify fields ...** item of the **DO_ON_LISTED** menu. For replacing or appending text to *fields* both approaches can be used.

When the **WRITE TO FIELDS OF LISTED SP.** button is used the **SET MANY FIELDS** window appears. Select a *field* from the list displayed on that window and press the appropriate button for writing to empty *field*s only or to replace the text of the respective *field*s of all selected *species*. The text in *subfield*s can be modified by defining the appropriate *tags*

Parse Field of Selection of Species

The **MODIFY DATABASE FIELD** panel which is displayed after selecting **Modify fields ...** from the **MORE_FUNCTIONS** menu allows more sophisticated modification of *field*s and *subfield*s. **Small programs** provided with the software or **customized** by the user analyze the entries and write the results to the selected *field* of the selected *species*

Select a *field* from the list displayed in the **Destination Field** subwindow and define a *Tag* (if desired). Select an option from the **... predefined program:** subwindow to display the syntax in the **Command ...** subwindow, modify the syntax or type your own if desired. The conventions for the syntax and examples are described in the on-line help text. For replacing or appending text `:` has to be typed followed by the current text separated by `=` from the new text. Use `*` and `?` as wild cards for multiple and single characters, respectively. See on-line help for defining wildcards for words. For analyzing *field* entries e.g. calculating base ratio of the sequence an *ACI* has to be used or typed (see on-line help).

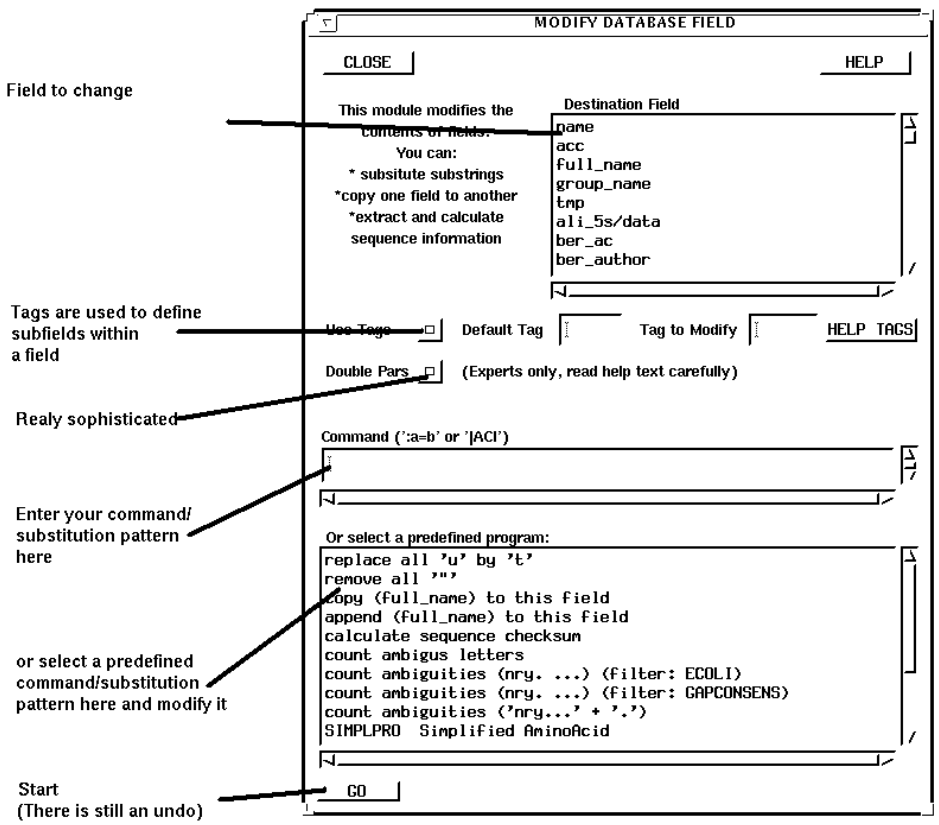
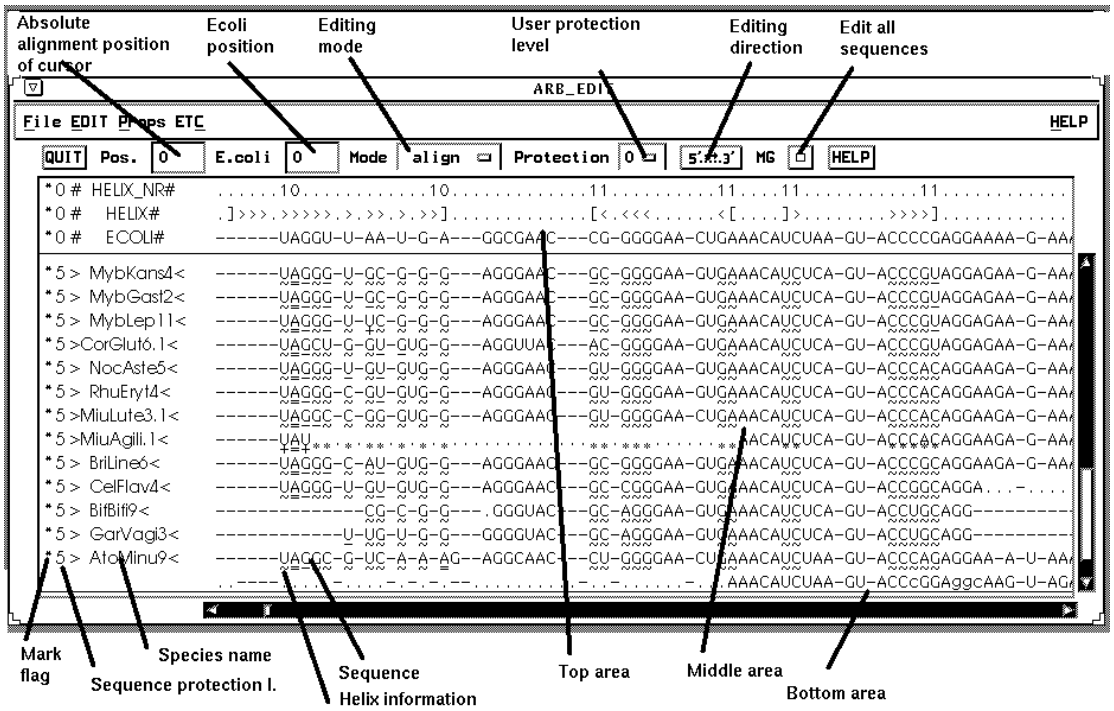


Figure 3 Main window for modifying database fields for a selection of species

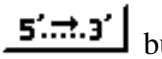
5 The Sequence Editors

The ARB Sequence Editor

The ARB editor is started by selecting **Edit Marked Sequences (ARB)** from the **Sequence** menu or pressing the button of the **ARB_NT** main window. The sequences of the *marked* *species* and all *SAI*s are displayed in the **ARB_EDIT** window. The *name*s of the *species* (>*name*<) and *SAI*s (#*name*#) are listed on the left. The number preceding the *name*s indicates the protection level. The order of the sequences is that of the *species* in the database. It can be changed temporarily by dragging the *name*. There are three regions within the display window. Vertical scrolling is only possible in the middle region. The sequences or *SAI*s can be moved to the upper or lower region by trying to drag them vertically out of the display area.



Three editing modes are available by selecting from the **Mode** menu. For editing, first position the cursor using the mouse. Once positioned it can be moved by the arrow keys also. The **align** mode allows to insert and remove gap symbols (**-** or **.**) and to move characters, however, not to change other characters. Multiple insertion or deletion of gap characters can be achieved by typing a number before typing the symbol (not possible when using the other modes). The **insert** and **replace** modes allow to type and delete any characters.

Depending on the orientation of the arrow in the  button (change by pressing) these operations can be performed in both directions. Single characters (not gap symbols) can be

fetched to the cursor position from both directions by pressing Meta together with left or right arrow key with the arrow facing towards the character to fetch. Characters can be moved from the cursor position to the next neighboring character by performing the same procedure with the arrow facing towards the character to move. Blocks of adjacent characters next to the cursor can be dragged by pressing Control together with the respective arrow key.

Generally, to allow editing the global protection level which is indicated in the second menu bar has to be identical or higher than that of the individual sequence (*SAI*). For modifying the alignment in the **align** mode it is recommended to set the protection level globally, while for modifying the sequence (*SAI*) characters the protection level of the particular sequence (*SAI*) should preferably adjusted by selecting from the **Edit** menu.

Potential secondary structure elements can be indicated by symbols below the characters. The display of the symbols is controlled by the settings defined by the user within the **HELIX PROPERTIES** window which can be invoked by selecting **Helix Symbols** from the **Properties** menu. The display of the symbols is immediately sensitive to any changes of the sequences or alignment. It is controlled by the *SAI*s **HELIX_NR** and **HELIX**. For conventions consult the on-line help.

New *species* (sequences) can be created by selecting **Create Sequence** from the **Edit** menu. The *name* which has to be defined by the user is under control of the *name server*.

GDE Sequence Editor

For users who still want to use the GDE editor designed by Steven Smith we build an interface to it. Unfortunately GDE is available only for some computer systems (no SGI and no HP version). We have changed GDE slightly to bahave like an ARB module. Unfortunately we did not succeed in making it fool proof. Please read the warning message which comes up automatically at GDE startup. This message includes the original GDE man pages.

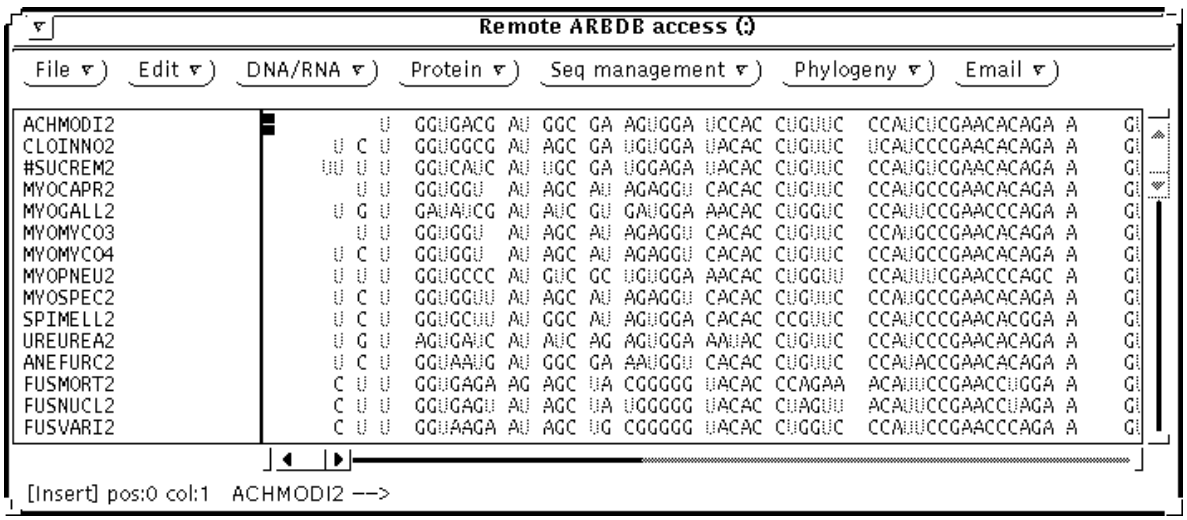


Figure 4 Screenshot of the GDE editor

Prototype of the ALE Sequence Editor

Some years ago the RDP started to develop a new editor called ALE. Unfortunately the editor was never finished. All what was left was a prototype which nevertheless offers excellent alignment editing. Like the GDE editor you should be carefull not to use more than one editor at a time.

The new ARB Editor (Number 4)

The running ARB editor was never planed to be an editor. It was a test program to test the Motif library. In the end it became our main editor, but it's source code looks like spaghetti. Right now (jan 97) we proudly present a prototype of the new editor (ARB_EDIT4), which is basically the old one, but offers much better multiple alignment functions.

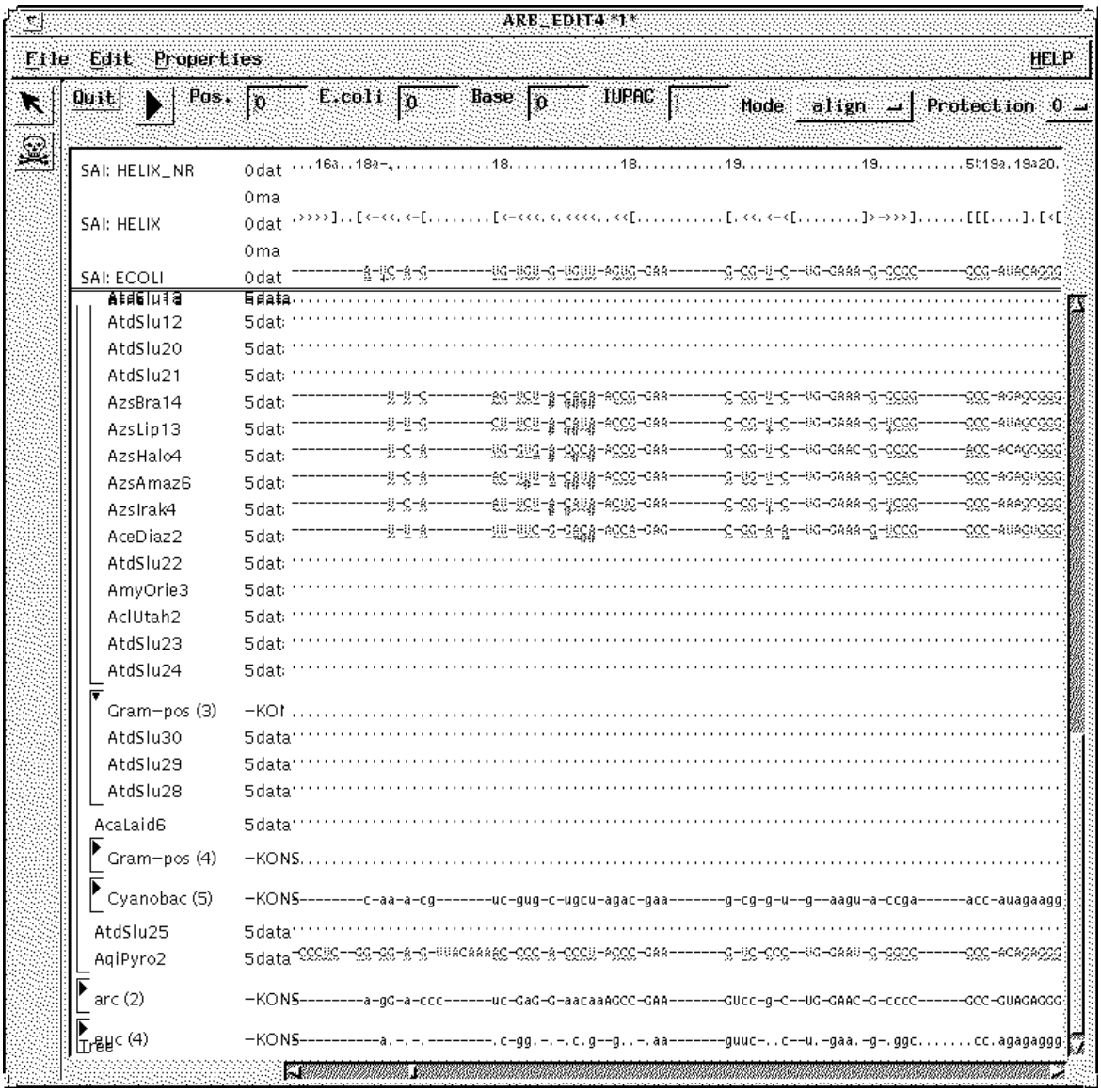


Figure 5 Screenshot of a prototype of ARB_EDIT4

6 Working with Sequences

Aligning Sequences

Aligning a Set of Sequences

In case that you have established a completely new ARB database and no aligned sequences are available, the alignment can be done by hand or by using **CLUSTALV** an integrated foreign program which is accessible from the **Align Sequences** submenu of the **Sequence** menu in the **ARB_NT** main window by pressing **Clustal ...**.

Aligning a New Sequence into a Given Alignment

If you are working with an ARB database of aligned sequences, select **Align Sequence ...** from the **Edit** menu of the **ARB_EDIT** window to invoke the ARB aligner environment. Define in the **AILGNER** window whether the sequences of all *marked* or only the *selected species* (*name* highlighted in the editor) should be aligned and whether the *PT_Server* should search for the most similar sequences to be used as templates or a user defined sequence should be used instead. In the latter case, type the *name* of the species to the **Species by name** subwindow. Select a *PT_Server* from the list which is displayed after pressing the button right to **PT_SERVER:**. Optionally, the closest relatives found by the *PT_Server* can be *marked* .

Note ensure to select a *PT_Server* for homologous sequences. The *PT_Server* does only perform the search for the closest relatives, however, the alignment is done with the sequences from these *species* as they are aligned in the current database.

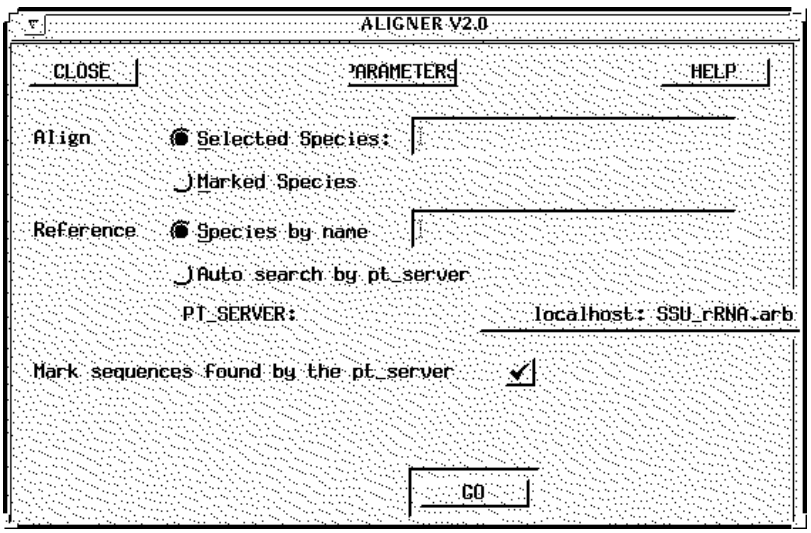
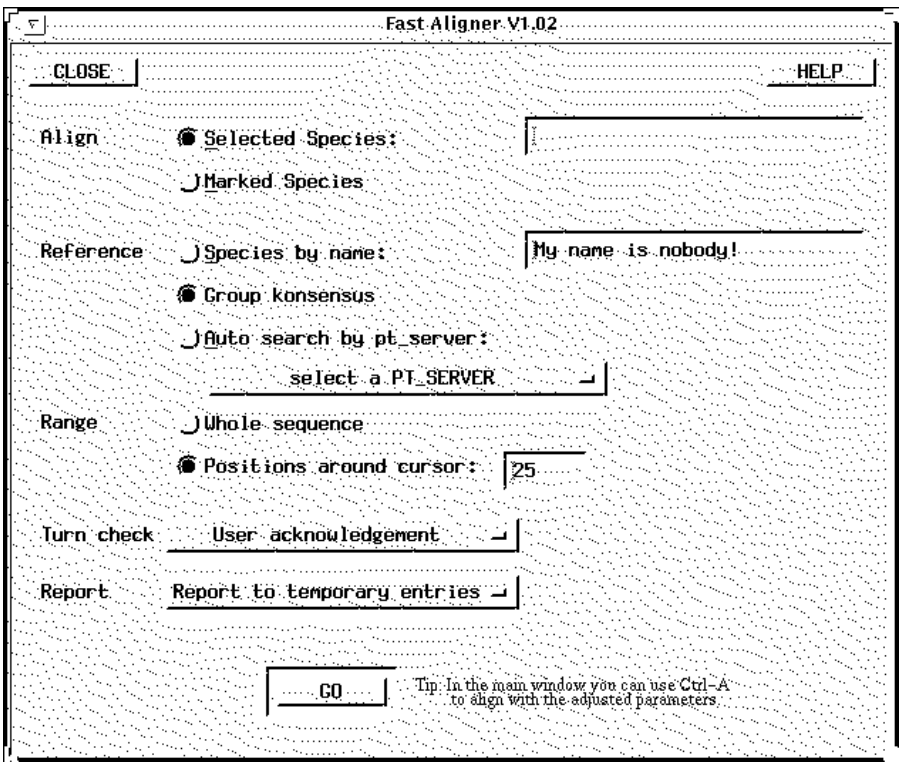


Figure 6 Screenshot of ALIGNER V2 start window

The aligner itself works in the background, you may edit everything except the sequence which is currently aligned. You may even start a second align process.

Note: The current version of the aligner has a small bug: It does not align the ends of short sequences very well.

There is a new aligner available in the new **ARB_EDIT4** editor. It is less sophisticated but much faster than the old aligner (up to 50 times). It does not share the bug with the old one (but has probably new bugs). It also allows to align part of the sequence and align against group consensus.



Filters, and Profiles (*SAI*)

Any sequence or alignment related information which can be displayed as a linear sequence of characters along with (aligned) primary structures can be stored as *SAI* (sequence associated information) and used as filter to in- or exclude alignment columns for analyses such as reconstructing phylogenetic trees . The tools to establish *SAI*s are accessible from the **SAI** menu.

Consensus Sequence

Consensus sequences based on the fraction and frequency of residues at the individual alignment positions of sequences from all or a selection of (*mark*ed) *species* can be established according user defined criteria. Select **Functions: Create ...** from the **SAI** menu of the **ARB_NT** main window to display a submenu. Select **Consensus** from the submenu. The **Expert Window** appears:

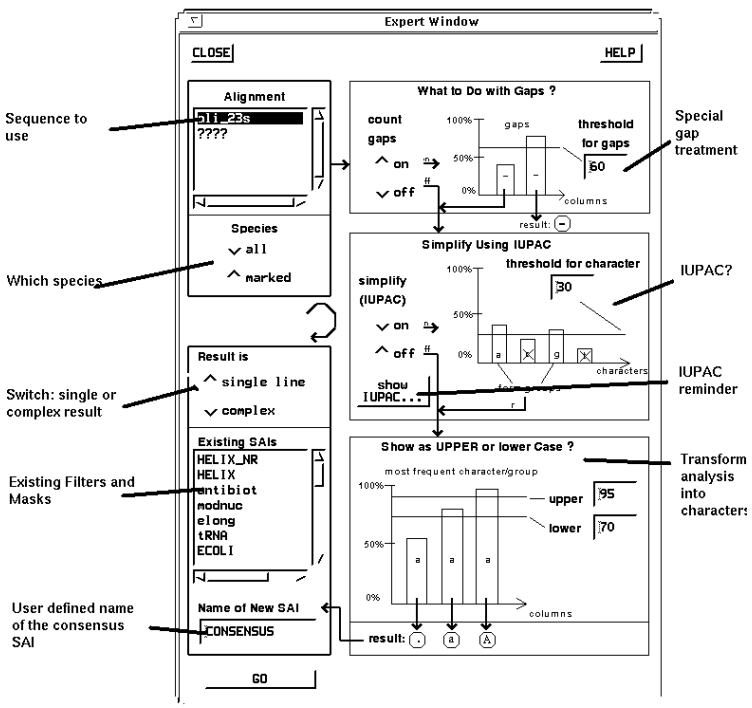


Figure 7 Creating a Consensus

Select an **Alignment** (if you have different sequence data sets in the database;) and define whether the sequences from all or only *mark*ed *species* should be analyzed (upper left subwindow).

Set threshold for in- or exclusion of characters and define whether gap consensus should be calculated and the IUPAC code should be used in the consensus sequence as described in the subwindows of the right part of the **Expert Window**.

Select or type a *name* for the new consensus in the lower left subwindow. Note: always select **single line** (the complex version has not yet been implemented.

Conservation Profiles

The tools for calculation conservation profiles are accessible from the submenu **Functions: Create ...** which is displayed clicking on the **SAI** menu button of the **ARB_NT** main window

Positional variation can be visualized as *SAI*s generated by simply calculating the fraction of the most frequent residues in a set of (*mark*ed) *species*. Select **Maximum Frequency** to display the **Max Frequency** window and define in that window whether gaps should be equivalent to residues or be ignored. Then select or type a *name* for the new *SAI*. The fractions are expressed as characters '1' - '9', '0' indicating 10% - 100%, respectively

A more detailed analysis based on the same criteria can be performed by selecting **Filter by ...**. The **ARB_PHYLO** window:

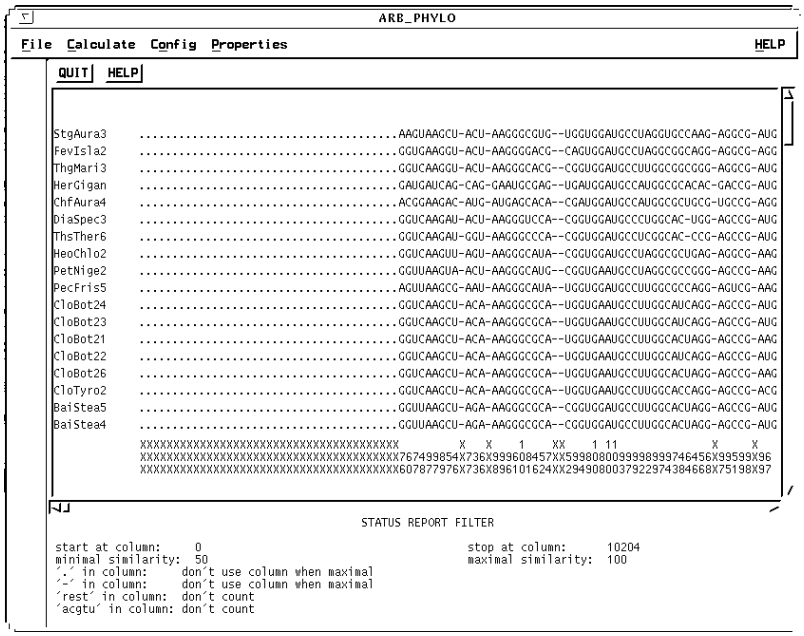


Figure 8 Building a filter using a maximum frequency method

displays the alignment of the sequences from the *mark*ed *species* and the default settings (lower part of the window). Press **Config** to display the **PHYL FILTER** window. Define first and last alignment positions to include as well as lower and upper threshold of positional similarity (fraction of most frequent base) value by typing to the respective subwindows. Select whether real gaps (–), unknown residues (.), ambiguity codes (*rest*) and lower case letters (*acgtu*) should be ignored (the particular position is ecxluded from binary comparison), the column should be excluded completely or only if the majority of sequences contains the character by pressing the corresponding buttons and selecting **dont' count, forget column** or **dont' use column when maximum**. After pressing **Calculate**, the result is displayed at the lower edge of the sequence subwindow. The frequencies (%) are given in columns to read from top to bottom. Alignment positions which are completely excluded are indicated by "**x**". Positions which do or do not fulfill the user defined similarity criteria are indicated by different user defined colors. The

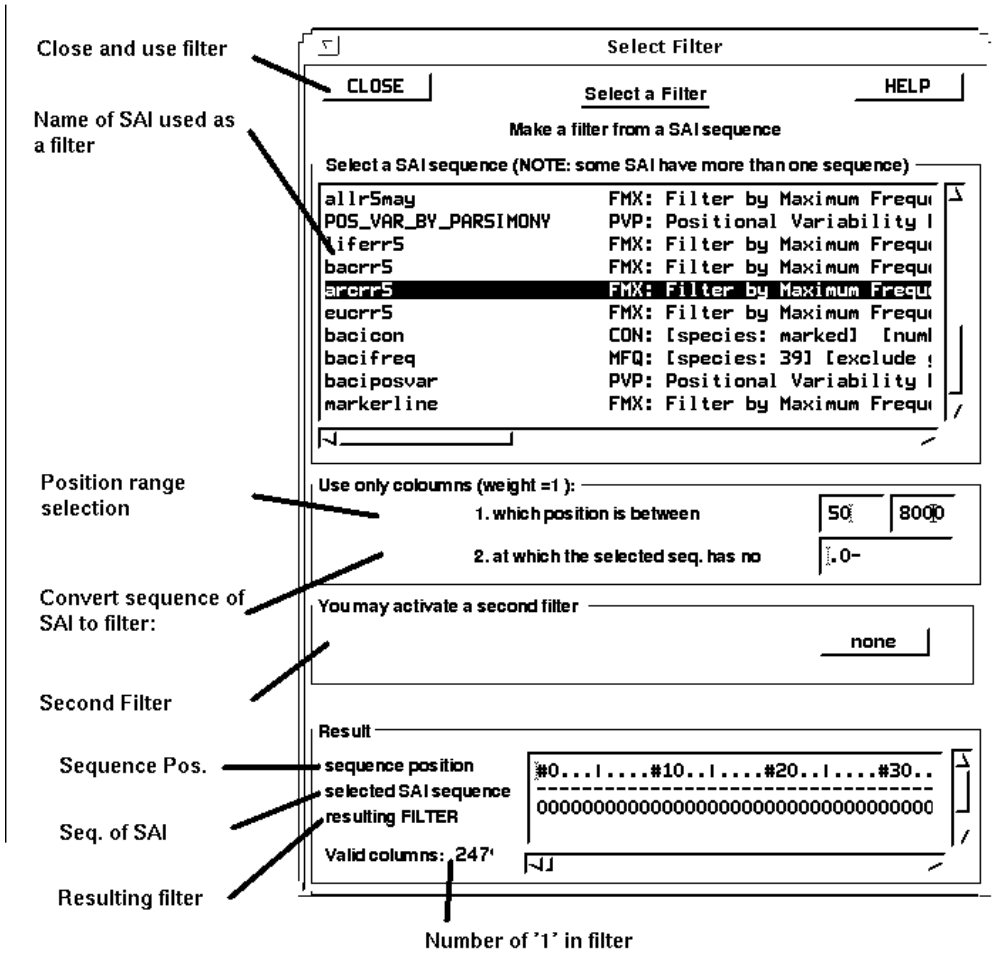
results can be exported to the database as single line *SAI* by selecting **Export Filter** from the **File** menu and subsequently selecting or typing a *name* in the **Export MLine** window. Note the *SAI* contains "**x**" and dots for selected and excluded columns, respectively.

Positional variation can also be visualized as *SAI* generated by a maximum parsimony analysis of the selected data. Select **Positional variability ...** and select a tree in the **Conservation Profile: Parsimony Method** window and select or type a *name* for the new *SAI*. Note: the tree information is needed for the program to enable estimation of positional variability according the parsimony criterion. You should select the most informative (usually the most comprehensive) tree

A more sophisticated (and experimental) method for estimating positional variability is accessible by selecting **ETC** from the **SAI** menu of the **ARB_NT** main window. For further instruction see the on-line manual provided with that tool.

The Use of Filters

The windows providing access to treeing and data exporting facilities contain a **Filter** button. Pressing this button gives access to the **Select Filter** window:



Select a *SAI* from the **Select a SAI ...** subwindow.

Define first and last alignment position to be included (absolute position including gaps as indicated in the ARB editor) by typing to the small subwindows right to **1. .. position .. between**. Specify characters of the *SAI* which should indicate columns to exclude from calculations by typing to the **2. at which ...** subwindow. Note: any character can be used.

The effective filter can be viewed in the **Result** window. In- or excluded positions are indicated by "`1`" and "`0`", respectively.

Multiple filters can be specified by pressing the button in the **You may activate ...** area. A new **Select Filter** window is displayed.

7 Phylogenetic Analyses

The central tool for phylogenetically organizing the database is a maximum parsimony based treeing method developed for the ARB-package (**ARB_PARS**). Another integral ARB component is a neighbor joining distance method which was further developed from the corresponding program from Felsensteins PHYLIP package. For establishing of similarity or distance matrices another ARB-tool was created. In addition foreign software for phylogenetic analyses was included in the ARB package such as PHYLIP, fastdnaml and GDE. For further information consult the respective literature or the original documentation partly provided with the ARB package.

Simple Rules

Often people are interested in simple rules how to get a good phylogeny. Unfortunately it is dangerous to give such rules because there is no perfect method in finding a correct phylogeny.. Nethertheless here are some personal rules (by Oliver Strunk, who is ARB's main programmer but not a biologist):

Which Problem	What to do
Search the next relatives of a single **short** sequence.	• Use netscape and start a blast search • Use the ARB parsimony program and insert your sequence into a big tree, do not use special filters, only generic filters like ECOLI to speed up the calculation process
Calculate a small tree of a single **full** sequence and it's nearest relatives.	• Search the nearest relatives first, • mark them • try different tree algorithms, compare the trees, get a feeling for your data.
Search for a good tree for a small number of **full** sequences.	Try: • Fastdnaml, • Parsimony • and Neighbour-Joining • with a lot of different filters • with different distance corrections and compare the resulting trees.

Figure 9 Simple Rules for Treeing (Continued) . . .

Which Problem	What to do
Search for a good tree for a small number of sequences of **different** length.	Avoid distance methods, rest as before.
Find a good tree of diverged **protein coding** sequences	Always do analysis in the protein sequence !!!! -> translate dna into protein Start neighbourjoining and protpars and compare trees.
What about filters ?	Rule: Take all informativ columns when doing phylogenetic analyses unless • you are interested in the deepest branches, in this case exclude variable positions from the treeing algorithm. • the sequences length varies significantly and you are working with distance methods, in this case exclude all columns where only some sequences have data.
What about weights/rates ?	Always try to use weights/rates for parsimony and fastdnaml algorithms if available.

Figure 9 Simple Rules for Treeing

Finding the Next Relative

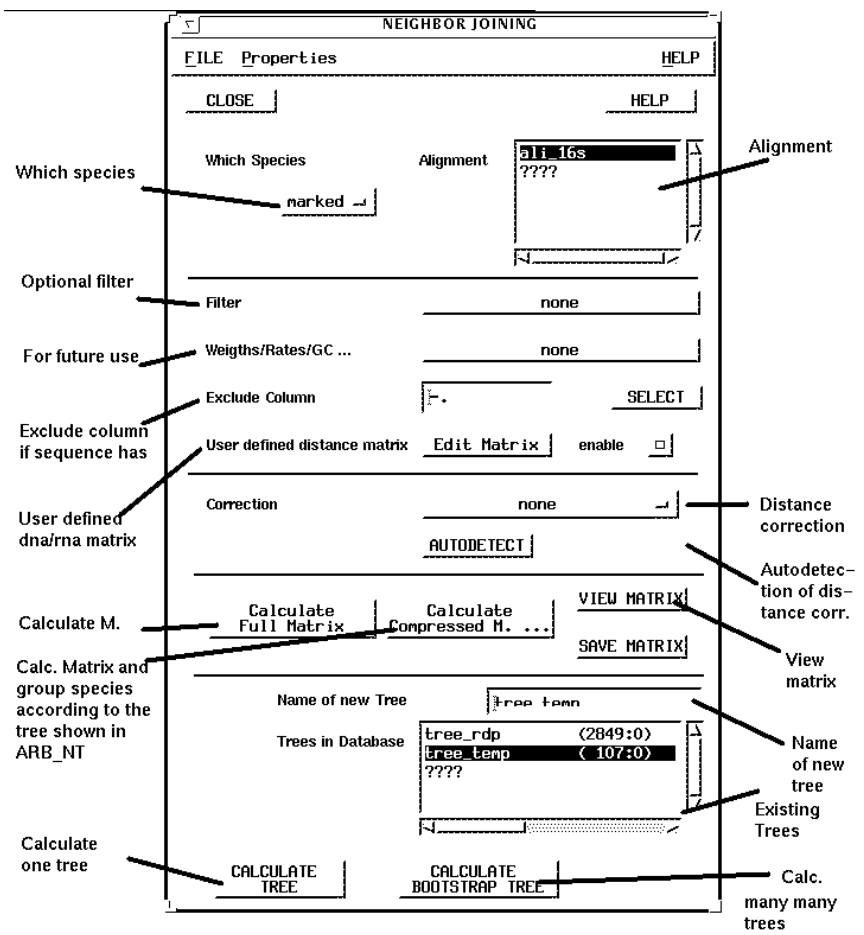
A scored list of next relatives can be obtained for any species without the need of a proper alignment of the sequence from the *species* in question. Display the **SEARCH and QUERY** window by selecting **Search** from the **Species** menu of the **ARB_NT** main window . Select a *species* from the **HIT LIST** (see 2.4.2.) and press **Search Next ...** in the **MORE_SEARCH** menu. The **Search Next Neighbors** window appears. Select a *PT_server* from the list displayed after clicking to the button below **Search Database (PT_server)**. The result will be displayed in the **Hits:** subwindow arranged according scoring values. Higher values indicate closer relationship. Note: an appropriate *PT_server* has to be established before running this program.

Distance and Similarity Matrices

Matrices of similarity, distance and phylogenetically corrected distance values can be generated using the **NEIGHBOR JOINING** window (same as for treeing using neighbor joining) which pops up after selecting **Multiple Sequence Comparison** from the **Sequence** menu of the **ARB_NT** main window and pressing the appearing **Distance Matrix** button.

Select an **Alignment** (if you have different sequence data sets in the database) and define whether the sequences from all or only *mark*ed *species* should be analyzed (upper part of the window).

If desired, define a *SAI* as *filter* (this means to exclude alignment positions according to the filter information) by selecting from a list which pops up after pressing the button right to **Filter**. Note: the *filter* defines which alignment columns are (completely) in- or excluded for the calculations.



Define which positions should be excluded (only) from binary sequence comparison by typing characters to the **Exclude Position** subwindow. For more information press the **SELECT** button.

To get values corrected according to evolutionary models select from the list displayed after pressing the button right to **Correction**. Note: items marked with "(exp)" indicate procedures not yet extensively tested. Selecting "none" means non corrected distances.

The buttons **Calculate Full Matrix** and **Calculate Compressed M.** are used to define whether normal matrix should be calculated or mean values should be given for groups as defined in a tree, respectively. This tree has to be selected from a list displayed in the **SELECT A TREE TO COMPRESS MATRIX** window. Mean values are calculated for those groups which are currently *compressed* in the specified tree.

The results can be viewed or saved as ASCII file by pressing the **VIEW MATRIX** and **SAVE MATRIX** buttons, respectively. In the latter case, name and formate of the file can be defined in the **Save Matrix** window.

Distance Matrix Tree by Neighbor Joining

Distance matrix trees can be reconstructed by automatically combining matrix calculation and the ARB-specifically modified neighbor joining program. Select **Build Tree ...** from the **Tree** menu of the **ARB_NT** main window and subsequently press **Neighbor Joining** from the submenu.

Select an **Alignment** (if you have different sequence data sets in the database) and define whether the sequences from all or only *mark*ed *species* should be analyzed (upper part of the window).

If desired, define a *SAI* as *filter* (this means to exclude alignment positions according to the filter information) by selecting an SAI from a list which pops up after pressing the button right to **Filter**. Note: the *filter* defines which alignment columns are (completely) in- or excluded for the calculations.

Define which positions should be excluded (only) from binary sequence comparison by typing characters to the **Exclude Position** subwindow. For more information press the **SELECT** button.

To get values corrected according to evolutionary models select from the list displayed after pressing the button right to **Correction**. Note: items marked with "(exp)" indicate procedures not yet extensively tested. Selecting "none" means non corrected distances.

Select a tree from the **Trees in Database** subwindow or type a name to the **Name of New Tree** subwindow. Note: selecting an existing tree results in overwriting. A new name typed to the latter window has to follow the convention: "**tree_....**.

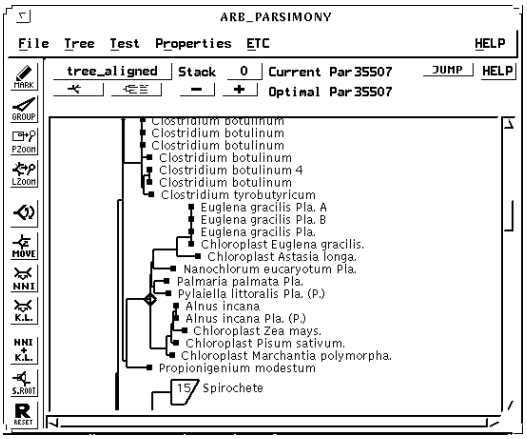
The new or updated tree is stored in the database and can be viewed in the **ARB_NT** main window.

Maximum Parsimony Tree by ARB_PARS

A special treeing program (**ARB_PARS**) has been developed to update and optimize trees based on large datasets according to the maximum parsimony criteria. To invoke the **ARB_PARS** environment select **Add Species ...** from the **Tree** menu of the **ARB_NT** main window and press the appearing **Parsimony** button. The **SET PARSIMONY OPTIONS** window pops up.

Select a tree to modify from the list in the **Tree** subwindow as well as an **Alignment** if there are different sequence sets in the database. Then select a filter by pressing the button right to **Filter** and define the proper selections in the appearing **Select Filter** window. **If you want to add species to large trees, always specify a filter to reduce the length of the alignment and therefor the computational ressources needed**.

After invoking the **ARB_PARS** environment the **ARB_PARSIMONY** window is displayed which resembles the **ARB_NT** main window. Some of the **ARB_NT** functions for modifying the tree layout (buttons in the left column, **Tree** and **ETC** menus) are available here:



Adding *species* to a Tree

Marked species or *selected species* can be added to the tree displayed in the **ARB_PARSIMONY** window by selecting **Add Species to Tree** from the **Tree** menu and pressing **Add Marked Species** or **Add Selected Species** from the submenu, respectively. This function uses the aligned sequence data to place the *marked/selected species* into the tree according to parsimony criteria without changing the tree topology (!). Note: the position of *marked/selected species* already present in the current tree is not changed.

Marked species or *selected species* can be added to the tree displayed in the **ARB_PARSIMONY** window by selecting **Add Species to Tree** from the **Tree** menu and pressing **Add Marked Species + Local Optimization** or **Add Selected Species + Local Optimization** from the submenu, respectively. This function uses the aligned sequence data to place the *marked/selected species* into the tree according to parsimony criteria and performs local tree optimization (see next topic). Note: (only) local optimization is also performed on *marked/selected species* already present in the current tree. Note: misplaced *species* may remain misplaced.

Marked species can be removed (if already placed in the tree) and (newly) inserted into the tree displayed in the **ARB_PARSIMONY** window according to parsimony criteria with or without local optimization by selecting **Add Species to Tree** from the **Tree** menu and pressing **Remove & Add Marked Species** or **Remove & Add Marked Species + Local Optimization** from the submenu, respectively.

Tree Optimization

Performing tree operations, the different versions of the resulting trees can be temporarily stored in a *stack* by pressing the [+] button in the upper part of the **ARB_PARSIMONY** window. A number is assigned to that version which is displayed right to **Stack**. Previous versions can be displayed by pressing [-] button. Note: only the tree currently displayed will be stored in the data base when quitting the **ARB_PARSIMONY** window. Note: after performing tree optimizations, the branch lengths have to be recalculated by selecting **Calculate Branch Lengths** from the **Tree** menu. The program sometimes changes the position of the root. If necessary, reset the root by pressing the

button in the left column of the **ARB_PARSIMONY** window, placing the cursor and pressing the left mouse button (as indicated in the upper part of the window).

Optimization of tree topologies according to parsimony criteria can be obtained by performing swapping of neighboring branches (nearest neighbor interchange, **NNI**) and determining parsimony values. Only tree topologies are maintained which are characterized by a better parsimony value. Also only branches will be modified which are not within a folded group. To perform this operation on the whole tree displayed in the **ARB_PARSIMONY** window select **Tree Optimization** from the **Tree** menu and press **Local Optimization NNI** in the submenu which pops up.

A more sophisticated optimization procedure which allows also swapping of distant branches under the control of a distance penalty is K.L. This procedure can be started on the whole tree by selecting **Tree Optimization** from the **Tree** menu and pressing **Global Optimization** in the submenu which pops up.

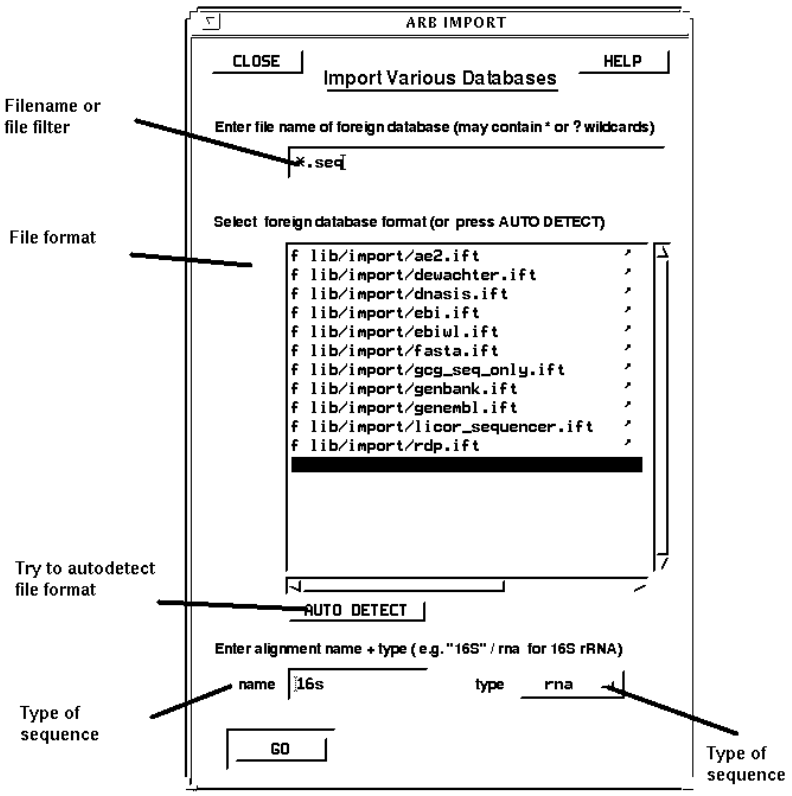
8 Importing Data

There are three ways to get new sequences into an existing ARB database:

- Sequences can be imported by typing in the ARB editor , by copying to the **Edit box** of the**SECIES INFORMATION** window.
- Simple import using the **ARB_NT/File/Import Sequences** function: OK if you import only some sequences (<20).
- Or convert your foreign format to an arb format first (start **`arb foreignformat`** and select import), store data as an ARB file, and merge it with your old data: OK if you want to import many sequences safely.

Converting Sequences

Start the ARB software by typing **`arb`** at a terminal window prompt. The **ARB INTRO** window pops up. ARB data files (suffix: **.arb**, marked by **f**) of the current directory (indicated in the first line: **CONTENTS OF .....**) are listed in the **Existing Files (f) and Directories (D)** subwindow. If desired, move to other directories by mouse click on the lines marked by **D**.



Press the **CREATE AND IMPORT** button to bring up the **ARB IMPORT** window. Specify the data file(s) in the **Enter file name ...** subwindow, select a formate from the **Select foreign**

database formate ... window or press the **AUTO DETECT** button. Type a name of the dataset (*alignment*) to the **name** subwindow and specify the sequence type by using the **type** button (the default type is indicated on the button).

Note: **If no file formats fits your sequences, there is always a chance of using the universal filter, which is never autodetected.**

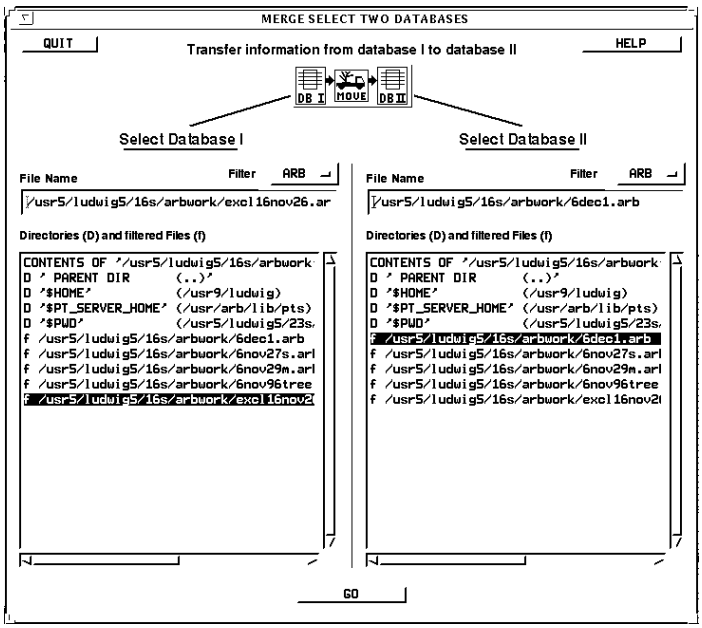
After reading the file(s), the **ARB_NT** main window comes up. Use the options of the **File** menu to save the new database.

Note: at the end of the procedure, the program asks whether new names should be generated from accession numbers and *full_names*. It is recommended to perform this operation to ensure consistency of ARB databases at your site. Therefore, this information should be present in the files to import. If this is not the case, provide these data using the corresponding tools when the **ARB_NT** main window is displayed and rerun the naming procedure by selecting **ETC** from the **Species** menu.

Merging ARB Databases

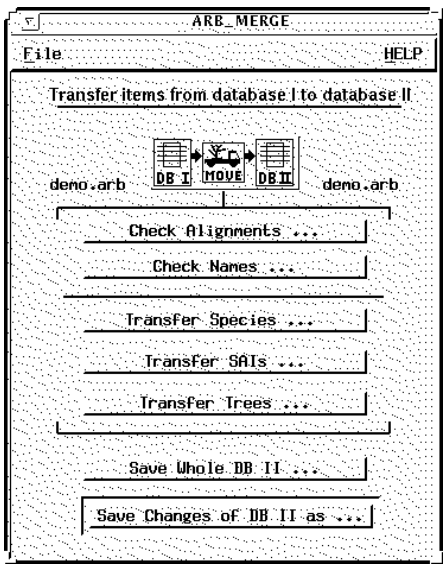
Start the ARB software by typing **`arb`** at a terminal window prompt. The **ARB INTRO** window pops up. ARB data files (suffix: **.arb**, marked by **f**) of the current directory (indicated in the first line: **CONTENTS OF .....**) are listed in the **Existing Files (f) and Directories (D)** subwindow. If desired, move to other directories by mouse click on the lines marked by **D**.

Press the **MERGE TWO ARB DATABASES** button. The **MERGE SELECT TWO DATABASES** window appears:



Select source (**Database I**) and destination (**Database 2**) database from the corresponding **Directories (D) and Files (f)** subwindows. Note: merging is an one way operation.

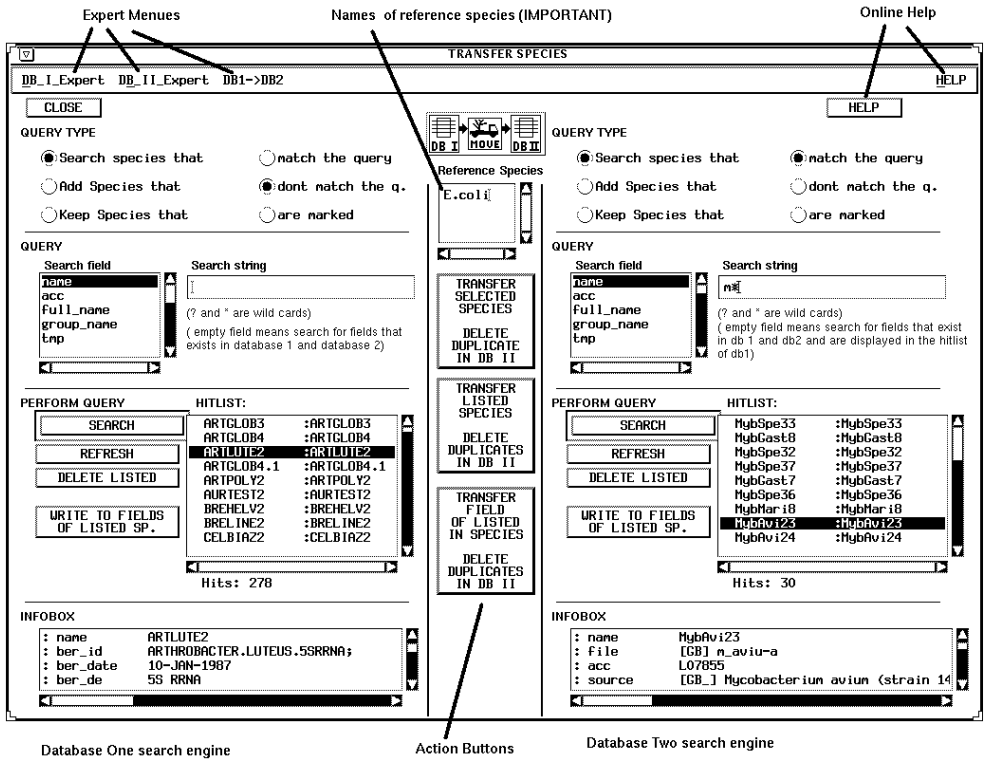
Press the **GO** button to invoke the merge environment. The **ARB_MERGE** window appears which allows to transfer all or part of database *species field* entries , *SAI*s, and trees.



First press **Check Alignments** to ensure that the names of the datasets of the two databases (*alignments*) containing homologous or equivalent sequences are identical. Note: if the names differ, the data will be stored as separate datasets in the same database. The names can be changed by using the **MODIFY** utility accessible in the **MERGE ALIGNMENTS** window.

Press **Check Names** to ensure identical *names* for identical *species* in both databases. Renaming of all (!) *species* in one or both databases using the *name_server* can be achieved by pressing the corresponding buttons of the **SYNCHRONIZE NAMES** window.

To transfer *species field* information press **Transfer Species** to bring up the **TRANSFER SPECIES** window. Select *species* of the source database from which data should be transferred by using the search facilities in the left part of the window. Note: if no search string is typed to the **Search string** subwindow, those *species* and *fields* are searched and listed which according to the settings in the upper half of the left window are identical or different in the destination database. Performing the same in the right part of the window results in a list of *species* and *fields* which are listed for the source database and are identical in the destination database.



Before transfering any kind of sequence, please enter the names of some reference species in the **Reference Species** input field. The two species should be in both databases and it's sequences should be equal in both databases (except the alignment may differ). These sequences are further on used to **preserve the alignment** when transferring data from left to right. **This species should be carefully chosen: They should**

- be important sequences (eg. E.Coli).
- be full sequences.
- cover the whole phylogenetic tree, meaning that each major branch of the tree is represented by it's sequence.

For the transfer of full information or single *field* entries use the appropriate buttons in the middle (**Transfer ...**).

For transferring trees and *SAI*s press the corresponding button of the **ARB_MERGE** window and select them in the **MERGE TREES** or **MERGE SAI** windows, respectively.

9 Saving Database

The ARB database can be saved from the **ARB_NT** main window as well as from the **ARB_MERGE** window by using the **File** menus or the **Save ...** buttons of the **ARB_MERGE** window. Either the complete database or only the recent changes are written to file. The ARB environment can be quitted from the **File** menu of the **ARB_NT** main window.

Appendix A Behind the Curtain

1 Introduction

It is most usefull to know how the program works, how data is stored and what kind of data is available. So we have collected a list of things to know, each item split into a beginner and advanced part. If you are starting to work with arb, read the beginner parts.

2 Database

Beginner

- Any database is saved into and loaded from one huge file. Only explicitly saving to that file makes changes permanent.
- All kind of data (trees, filters, sequences ...) are stored in an ARB database.
- Each user has his private database consisting of one or multiple files (like a word document) and each modification is only private. From time to time the system administrator may collect all private files and merge them into a new database release, which itself is redistributed to the users. Therefore all private changes should be well documented.
- Simply start arb by typing 'arb <return>' and select a database.
- Never delete or copy database files by hand, use arb instead.
- From time to time optimize the database. Rule: do optimization after:
  - ☐ a great number of sequences is added to the database.
  - ☐ a great number of sequences has been changed.
  - ☐ a global alignment insert has been performed.
- You may save the entire database or only the changes. Rule: Always save only the changes unless
  - ☐ you made major database changes like inserting a gap in the whole alignment.
  - ☐ you want to transfer data via ftp.
  - ☐ you reoptimized the database.

Advanced

- A database normally consists of different files:
  - database.arb: The basic database (important)
  - database.a##: where # is a numerical value. (important)
    As storing the entire database takes a while, ARB allows you to store only the changes made to the database into a changes file. The last 5 changes files are not deleted, so you may roll back to the last 5 database states. All changes files are independent and old changes files may be deleted without any harm. ARB itself searches for the latest changes file (the file with the highest number) automatically.
  - database.ARF A list of references to this database (optional)
  - database.ARM An optional fast load file. If you have a computer with only a small amount of memory, it may take very long to open a database. If there is a 'fast load' file available the loading procedure is much faster. If you delete a database.ARM file, no date is lost, but loading the corresponding database may take more time.
- The sequence and other data are stored in a compressed form. Compressing all sequences at one time results in much better compression ratios compared to compressing single sequences. So once a sequence is changed it's good compression is automatically changed to a worse one. If a great portion of the database is badly compressed, doing an optimization step will compress all sequences with the good method.

3 Database Objects

Beginner

There are three main type of data objects:

- Species: Everything which has a real sequence and some documentation information. Example: *E.Coli* with 16s sequence.
  Each species has a unique name.
- SAI: Everything which has a sequence but is not a biological sequence, like a filter, consensus, weights, rates, statistical information, helix information, ...
- Trees: Phylogenetic trees.

Advanced

Each database set (like species SAI, tree etc) has a number of subfields:

- Each subfield has a type:
  - string: Any number of characters except \0
  - integer: a value
  - double: a real value
  - integers: an array of integers
  - doubles: an array of real values
  - container: has no value but a set of subfields
- Each subfield has a key. Example of subfields of species
  - name: The internal name of a species/SAI
  - full_name: The official name of an organism
  - acc: The accession number of a sequence
  - ali_16s: (container) A set of subfields containing the 16s sequence and additional information (alignment quality, etc)
  - ali_16s/data: data is a subfield of ali_16s and holds the species' sequence.
- Each subfield has a protection level.

You may add as many subfields as needed to your species. Because only those fields, which do have a value, are actually generated, creating new and empty fields does not need any computer ressources.

Sequences are treated differently: To create new sequence types do not create new species fields but use the **<sequence/admin>** window instead. There you may define the overall sequence length, default write security, create copies of sequences ...

4 NDS & SRT/ACI/REG

Absolute Beginner

Skip this section

Beginner

All species data can be represented as a string (eg. name full_name sequence ..). ARB offers many many ways to manipulate those strings. The most simple application for this is called **NDS** (Node Display Settings). In practice this means that: **the user can select nearly any kind of information which will be displayed at the tips of the tree.**. This is an extremely powerfull tool and enables such functions as:

- show part of the sequence at the tips
- calculate GC content on the fly
- show name full_name and accession number ...
- and many million more possibilites.

Please read the online help offered in the **<Tree/NDS>** subwindow.

Advanced

Often you want to store the result of the NDS output back into the database. This gives you for example the possibilty to simplify amino acid alignments, calculate the nucleotides of a sequence, store the current date into a list of species, ...

All this is done with the **Modify Fields of Listed Species** in the **Search Species** subwindow. As this manual cannot be up to date, please refer to the online documentation:

This is an excerpt:

```
TITLE  MODIFY FIELDS OF LISTED SPECIES

OCCURRENCE ARB_NT/Species/Search: MODIFY FIELDS OF LISTED SPECIES
  ARB_NT/Tree/NDS

DESCRIPTION Finds and replaces substrings within fields/tagged subfields of
  all listed species. The entries within the selected fields of
  all listed species can be modified either individually or
  globally.

  Two different languages can be used to modify an entry:

  SRT: indicated by a leading ':' character
  ACI: indicated by a leading '|' character
  REG: indicated by sourrounding '/' characters
```

```
  REG: Simple Regular Expressions (not for beginners)
   '/Seach RegExpr/Replace String/'
   See help text for more details

  SRT: Replaces substrings
   Syntax: ':old_string=new_string'
   see SRT help text for more details
   example: remove all spaces -> SRT ': ='

   Different search/replace commands can be performed
   simultanously and have to be seperated by ':'
   ':search1=replace1:search2=replace2: ... :searchn=replacen'.

   * and ? are wild cards for multiple and single
     characters, respectively.

  ACI: More sophisticated string manipulations
   ( Read help text for more information)


NOTES  You may add new commands by editing the file
   $ARBHOME/lib/sellists/mod_fields.sellst
  You should save this file to another location when
  installing new versions of ARB

EXAMPLES ':p?r=p?1w' replaces  par to paw
      pbr to pbw
      pcr to pcw ...
  ':p??r=p?2?1r' swaps the two letters between p and r

  ':a*=b*1'  replaces only the first 'a' by 'b'
  ':?* *=?1. *2' Replaces the first word by its first
    letter + '.'
  ':\:=\n' replaces all ':' by <new_line>
  ':*=*1 *(key1)' appends the database field <key1>
  ':*=*1 *(key1|nothing found)'
    appends the database field <key1>
    if <key1> does not contain entries
    append 'nothing  found'

  1. Global modification: Replace 'spec.' by 'sp.' within
   the field full_name of all listed species:

   Press:  'MODIFY FIELDS OD LISTED SPECIES'

   Select Field: 'full_name'
   Type Command: ':spec.=sp.'
   Press:  'GO'

  2. Individual modification: Append the particular entries
   of fields 'title' and 'journal' to that of the
   fields 'author' of all listed species if there
   are any entries:
```

Press: 'MODIFY FIELDS OD LISTED SPECIES'

Select Field: 'author'
Type Command: ':*=*1 *(title) *(journal)'
Press: 'GO'

NOTE Undo does work.

WARNINGS Be carefull if search or replace string contain special characters (such as ':').

BUGS No bugs known

5 PT_SERVER

Basic

The probe designing and matching tools ('ARB_PROBE') and the aligner of the editor ('ARB_EDIT') rely on 'PT_SERVER' databases and servers. By default there are no PT_SERVER files, but they can be easily created using the **ETC/PT_SERVER Admin/UPDATE SERVER** function. ARB offers some predefined PT_SERVER templates and it's up to the user to fill those templates with the currently loaded data. Please update the 'PT_SERVER' databas when new sequence (species) entries or sequence modifications (base changes) have been introduced and aligned into the current data base.

Advanced

Please please read the online documentation: **PT_SERVER: What Why and How**.

6 General Conception

Beginner

Security Level

All data fields have a security level. This means only users with a higher security level than a data field can change or delete it. The security levels vary between 0 and 7, where a user can select just 0 to 6. Level 7 data never can be changed or deleted. Security levels are (until now) not to be used to disable unauthorizised users to change data, but to reduce the risk of data loss. Any user can raise his security level without any password (until now, future versions may include this option).

Different protection levels can be assigned to different database *field* entries. This can be done using the corresponding facilities of the ARB editor for sequence entries, the respective tools in combination with database searching for any database *field* entries, and the tree administration tool for trees. To modify or delete any protected database entries, the identical or higher protection level has to be selected by using the **Protection** button in the upper right part of the **ARB_NT** main window.

Advanced

System design

Although there exist different modules all they act as one single program as they are all connected over the ARBDB database. ARB_NT is the server, all other programs are clients.

7 Installation

Beginner

Do not care about the installation of ARB.

Advanced

PT_SERVER files

During the installation process s special directory can be assigned to hold the pt_server files. This directory should residue on a huge and fast hard drive. Assigning these directory helps to keep your data during a programm update.

Versions of ARB

ARB is still under construction, do not expect anything perfect, download a new version every 3 month, install it not to the old location but to a new one (maybe the new version has a bug).

Nearly every day I find a minor bug in the program (only 1 major bug every 3 months). But fixing a bug has always the risk of creating a new one. So sometimes it happen

s that the ARB version on our ftp server is a bad one. Sorry. In those cases write me (strunk@pages.de) an email, wait for a new release (1day-2weeks) and install the new version.

PT_SERVER & $(ARBHOME)/lib/arb_tcp.dat

The file '$ARBHOME/lib/arb_tcp' contains entries for communication services used by the different ARB modules. Each line tells an ARB programm where to find its server. Normally there is no need to change this file unless you want to create new PT_SERVER (eg. 18s, special user pt_server ...). The file contains some help text.

This is an example file:

```
#******************* Syntax ***********************************
#
# [USER:]SERVER_ID  HOST:IP_NR [programm args]
# [USER:]SERVER_ID  :SOCKETPATH [programm args]
#  [xxx] means optional field
#  all $(enviroment_variable) are replaced by the value of the 'enviroment_variable'
#
# Existing SERVER_IDs:
#
#  ARB_DB_SERVER   Your private database server. It is automatically started
#  ARB_PT_SERVER  Default PT_SERVER for PB_RETRIEVE (private TU_Munich server)
#  ARB_PT_SERVERn  Global PT_SERVERs. Needed for fast database search (align seq, probe des.)
#     n starts from 0 to max, no number can be excluded !!!
#  ARB_NAME_SERVER  Generates short names for species
#
#*****************************************************************


#*********** Private Servers (for each user) ******************
#
# This is a local server, running on one machine. Each user has his own socket:
# ARB_PID is an Enviroment Variable which is set by the command 'arb' to its own process id
# see programm $ARBHOME/SH/arb_clean to remove the sockets

ARB_DB_SERVER  :/tmp/arb_db_$(USER)_$(ARB_PID)

# If you want to run programms on a workstation cluster, you have to assign your arbdb server
# and one ip-id for each user. To choose an ip_id choose any number between 1000 and 4000 and
# examine /etc/services whether this ip-id has not yet been allocated by another programm
# smith:ARB_DB_SERVER  myhost:4011  // socket for smith ('arb' have to be started on 'myhost')
# ludwig:ARB_DB_SERVER  myhost:4012  // socket for ludwig
# ARB_DB_SERVER  :/tmp/arb_db_$(USER)_$(ARB_PID) // and sockets for all others

#*********** Global Servers (for all users) ******************
#
#*********** Nameserver The server should run on the nfs server ******************

ARB_NAME_SERVER  pop:3029  arb_name_server -d$(ARBHOME)/lib/nas/names.dat

#*********** PT_SERVERS The server should run on the nfs server ******************
#*******     You may add new pt_servers here: (numbers must be continues)   *********

ARB_PT_SERVER  pop:3030  arb_pt_server -D$(ARBHOME)/lib/pts/16s_rrna_aligned.arb
ARB_PT_SERVER0  pop:3030  arb_pt_server -D$(ARBHOME)/lib/pts/16s_rrna_aligned.arb
ARB_PT_SERVER1  pop:3032  arb_pt_server -D$(ARBHOME)/lib/pts/23s_rrna_aligned.arb
ARB_PT_SERVER2  pop:3034  arb_pt_server -D$(ARBHOME)/lib/pts/atp.arb
ARB_PT_SERVER3  pop:3035  arb_pt_server -D$(ARBHOME)/lib/pts/LSU_rRNA.arb
ARB_PT_SERVER4  pop:3036  arb_pt_server -D$(ARBHOME)/lib/pts/rrna.arb
ARB_PT_SERVER5  pop:3037  arb_pt_server -D$(ARBHOME)/lib/pts/SSU_WL.arb
ARB_PT_SERVER6  pop:3038  arb_pt_server -D$(ARBHOME)/lib/pts/user.arb
```