Technical

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (FRANCE)
CENTRO DE REGULACIO GENOMICA (SPAIN)

Cédric Notredame
www.tcoffee.org

T-Coffee: Technical Documentation

T-Coffee Technical Documentation (Version 8.01, July 2009) www.tcoffee.org

T-Coffee, seq_reformat PSI-Coffee, 3D-Coffee, M-Coffee, R-Coffee, APDB, iRMSD, T-RMSD

License and Terms of Use 6
T-Coffee is distributed under the Gnu Public License 6
T-Coffee code can be re-used freely 6
T-Coffee can be incorporated in most pipelines: Plug-in/Plug-out… 6

Addresses and Contacts 7
Contributors 7
Addresses 7

Citations 8
T-Coffee 8
Mocca 10
CORE 10
Other Contributions 10
Bug Reports and Feedback 10

Installation of The T-Coffee Packages 11
Third Party Packages and On Demand Installations 11
Standard Installation of T-Coffee 11
Unix 11
Microsoft Windows/Cygwin 13
MAC osX, Linux 13
CLUSTER Installation 13
If you have PDB installed: 13
Installing BLAST for T-Coffee 14
Why Do I need BLAST with T-Coffee? 14
Using the EBI BLAST Client 14
Using the NCBI BLAST Client 15
Using another Client 15
Using a BLAST local version on UNIX 16
Using a BLAST local version on Windows/cygwin 17
Installing Other Companion Packages 17
Installation of PSI-Coffee and Expresso 19
Installation of M-Coffee 19
Automated Installation 19
Manual Installation 20
Installation of APDB and iRMSD 22
Installation of tRMSD 22
Installation of seq_reformat 22
Installation of extract_from_pdb 22
Installation of 3D-Coffee/Expresso 23
Automated Installation 23
Manual Installation 23
Installing Fugue for T-Coffee 24
Installation of R-Coffee 24
Automated Installation 24
Manual Installation 24
Installing ProbbonsRNA for R-Coffee 25
Installing Consan for R-Coffee 25

Quick Start 26
T-COFFEE 26
M-Coffee 26
Expresso 27
R-Coffee 27

iRMSD and APDB **28**
tRMSD **28**
MOCCA **29**

Recent Modifications 30

Reference Manual 31

Environment Variables **31**
http_proxy_4_TCOFFEE *31*
email_4_TCOFFEE *32*
DIR_4_TCOFFEE *32*
TMP_4_TCOFFEE *32*
CACHE_4_TCOFFEE *32*
PLUGINS_4_TCOFFEE *32*
NO_ERROR_REPORT_4_TCOFFEE *32*
PDB_DIR *32*
NO_WARNING_4_TCOFFEE *32*
UNIQUE_DIR_4_TCOFFEE *32*
Setting up the T-Coffee environment variables **32**
Well Behaved Parameters **33**
Separation *33*
Posix *33*
Entering the right parameters *33*
Parameters Syntax **34**
No Flag *34*
-parameters *34*
-t_coffee_defaults *35*
-mode *35*
-score [Deprecated] *35*
-evaluate *35*
-convert [cw] *36*
-do_align [cw] *36*
Special Parameters **36**
-version *36*
-proxy *36*
-email *36*
-check_configuration *36*
-cache *37*
-update *37*
-full_log *37*
-plugins *37*
-other_pg *37*
Input **38**
Sequence Input *38*
-infile [cw] *38*
-in (Cf –in from the Method and Library Input section) *38*
-get_type *38*
-type [cw] *38*
-seq *38*
-seq_source *38*
Structure Input *39*
-pdb *39*
Tree Input *39*
-usetree *39*
Structures, Sequences Methods and Library Input via the □in Flag *39*
-in *40*
Profile Input *41*

-profile 41
-profile1 [cw] 42
-profile2 [cw] 42
Alignment Computation **42**
Library Computation: Methods 42
-lalign_n_top 42
-align_pdb_param_file 42
-align_pdb_hasch_mode 42
Library Computation: Extension 42
-lib_list [Unsupported] 42
-do_normalise 43
-extend 43
-extend_mode 43
-max_n_pair 43
-seq_name_for_quadruplet 44
-compact 44
-clean 44
-maximise 44
-do_self 44
-seq_name_for_quadruplet 44
-weight 44
Tree Computation 45
-distance_matrix_mode 45
-quicktree [CW] 45
Pair-wise Alignment Computation 46
-dp_mode 46
-ktuple 47
-ndiag 47
-diag_mode 47
-diag_threshold 47
-sim_matrix 47
-matrix [CW] 48
-nomatch 48
-gapopen 48
-gapext 48
-fgapopen 48
-fgapext 48
-cosmetic_penalty 48
-tg_mode 49
Weighting Schemes 49
-seq_weight 49
Multiple Alignment Computation 49
-msa_mode 49
-one2all 49
-profile_comparison 50
-profile_mode 50
Alignment Post-Processing 50
-clean_aln 50
-clean_threshold 51
-clean_iteration 51
-clean_evaluation_mode 51
-iterate 51
CPU Control **51**
Multithreading 51
-multi_core 51
-n_core 52
Limits 52

-mem_mode 52
-ulimit 52
-maxlen 52
Aligning more than 100 sequences with DPA 52
-maxnseq 52
-dpa_master_aln 53
-dpa_maxnseq 53
-dpa_min_score1 53
-dpa_min_score2 53
-dap_tree [NOT IMPLEMENTED] 53
Using Structures **54**
Generic 54
-mode 54
-check_pdb_status 54
3D Coffee: Using SAP 54
Using/finding PDB templates for the Sequences 55
-template_file 55
-struc_to_use 57
Multiple Local Alignments **57**
-domain/-mocca 57
-start 57
-len 58
-scale 58
-domain_interactive [Examples] 58
Output Control **59**
Generic 59
Conventions Regarding Filenames 59
Identifying the Output files automatically 59
-no_warning 59
Alignments 59
-outfile 59
-output 59
-outseqweight 60
-case 60
-cpu 61
-outseqweight 61
-outorder [cw] 61
-inorder [cw] 61
-seqnos 61
Libraries 61
-out_lib 62
-lib_only 62
Trees 62
-newtree 62
Reliability Estimation **62**
CORE Computation 62
-evaluate_mode 62
Generic Output **63**
-run_name 63
-quiet 63
-align [CW] 63
APDB, iRMSD and tRMSD Parameters **63**
-quiet [Same as T-Coffee] 64
-run_name [Same as T-Coffee] 64
-aln 64
-n_excluded_nb 64
-maximum_distance 64

-similarity_threshold *64*
-local_mode *65*
-filter *65*
-print_rapdb [Unsupported] *65*
-outfile [Same as T-Coffee] *65*
-color_mode *65*

Building a Server 66

Environment Variables *66*
Output of the .dnd file. *67*
Permissions *67*
Other Programs *67*

Formats 68

Parameter files **68**
Sequence Name Handling **68**
Automatic Format Recognition **69**
Structures **69**
RNA Structures **69**
Sequences **69**
Alignments **69**
Libraries **70**
T-COFFEE_LIB_FORMAT_01 ..... *70*
T-COFFEE_LIB_FORMAT_02 ..... *71*
Library List **71**
Substitution matrices. **71**
ClustalW Style [Deprecated] *71*
BLAST Format [Recommended] *72*
Sequences Weights **72**

Known Problems 73

Technical Notes 74

Development **74**
Command Line List *74*

To Do... 76

License and Terms of Use

T-Coffee is distributed under the Gnu Public License

Please make sure you have agreed with the terms of the license attached to the package before using the T-Coffee package or its documentation. T-Coffee is a freeware open source distributed under a GPL license. This means that there are very little restrictions to its use, either in an academic or a non academic environment.

T-Coffee code can be re-used freely

Our philosophy is that code is meant to be re-used, including ours. No permission is needed for the cut and paste of a few functions, although we are always happy to receive pieces of improved code.

T-Coffee can be incorporated in most pipelines: Plug-in/Plug-out...

Our philosophy is to insure that as many methods as possible can be used as plug-ins within T-Coffee. Likewise, we will give as much support as possible to anyone wishing to turn T-Coffee into a plug-in for another method. For more details on how to do this, see the plug-in and the plug-out sections of the Tutorial Manual.

Again, you do not need our permission to either use T-Coffee (or your method as a plug-in/out) but if you let us know, we will insure the stability of T-Coffee within your system through future releases.

The current license only allows for the incorporation of T-Coffee in non-commercial pipelines (i.e. where you do not sell the pipeline, or access to it). If your pipeline is commercial, please get in touch with us.

Addresses and Contacts

Contributors

T-coffee is developed, maintained, monitored, used and debugged by a dedicated team that include or have included:

Cédric Notredame, Fabrice Armougom, Des Higgins, Sebastien Moretti, Orla O'Sullivan. Eamon O'Toole, Olivier Poirot, Karsten Suhre, Iain Wallace, Andreas Wilm

Addresses

We are always very eager to get some user feedback. Please do not hesitate to drop us a line at: cedric.notredame@europe.com The latest updates of T-Coffee are always available on: www.tcoffee.org . On this address you will also find a link to some of the online T-Coffee servers, including Tcoffee@igs

T-Coffee can be used to automatically check if an updated version is available, however the program will not update automatically, as this can cause endless reproducibility problems.

```
PROMPT: t_coffee -update
```

Citations

It is important that you cite T-Coffee when you use it. Citing us is (almost) like giving us money: it helps us convincing our institutions that what we do is useful and that they should keep paying our salaries and deliver Donuts to our offices from time to time (Not that they ever did it, but it would be nice anyway).

Cite the server if you used it, otherwise, cite the original paper from 2000 (No, it was never named "T-Coffee 2000").

Notredame C, Higgins DG, Heringa J. — Related Articles, Links

T-Coffee: A novel method for fast and accurate multiple sequence alignment.
J Mol Biol. 2000 Sep 8;302(1):205-17.
PMID: 10964570 [PubMed - indexed for MEDLINE]

Other useful publications include:

T-Coffee

Claude JB, Suhre K, Notredame C, Claverie JM, Abergel C. — Related Articles, Links

CaspR: a web server for automated molecular replacement using homology modelling.
Nucleic Acids Res. 2004 Jul 1;32(Web Server issue):W606-9.
PMID: 15215460 [PubMed - indexed for MEDLINE]

Poirot O, Suhre K, Abergel C, O'Toole E, Notredame C. — Related Articles, Links

3DCoffee@igs: a web server for combining sequences and structures into a multiple sequence alignment.
Nucleic Acids Res. 2004 Jul 1;32(Web Server issue):W37-40.

PMID: 15215345 [PubMed - indexed for MEDLINE]

O'Sullivan O, Suhre K, Abergel C, Higgins DG, Notredame C. Related Articles, Links

3DCoffee: combining protein sequences and structures within multiple sequence alignments.
J Mol Biol. 2004 Jul 2;340(2):385-95.
PMID: 15201059 [PubMed - indexed for MEDLINE]

Poirot O, O'Toole E, Notredame C. Related Articles, Links

Tcoffee@igs: A web server for computing, evaluating and combining multiple sequence alignments.
Nucleic Acids Res. 2003 Jul 1;31(13):3503-6.
PMID: 12824354 [PubMed - indexed for MEDLINE]

Notredame C. Related Articles, Links

Mocca: semi-automatic method for domain hunting.
Bioinformatics. 2001 Apr;17(4):373-4.
PMID: 11301309 [PubMed - indexed for MEDLINE]

Notredame C, Higgins DG, Heringa J. Related Articles, Links

T-Coffee: A novel method for fast and accurate multiple sequence alignment.
J Mol Biol. 2000 Sep 8;302(1):205-17.
PMID: 10964570 [PubMed - indexed for MEDLINE]

Notredame C, Holm L, Higgins DG. Related Articles, Links

COFFEE: an objective function for multiple sequence alignments.
Bioinformatics. 1998 Jun;14(5):407-22.
PMID: 9682054 [PubMed - indexed for MEDLINE]

Mocca

Notredame C. Related Articles, Links

Mocca: semi-automatic method for domain hunting. Bioinformatics. 2001 Apr;17(4):373-4.
PMID: 11301309 [PubMed - indexed for MEDLINE]

CORE

http://www.tcoffee.org/Publications/Pdf/core.pp.pdf

Other Contributions

We do not mean to steal code, but we will always try to re-use pre-existing code whenever that code exists, free of copyright, just like we expect people to do with our code. However, whenever this happens, we make a point at properly citing the source of the original contribution. If ever you recognize a piece of your code improperly cited, please drop us a note and we will be happy to correct that.

In the mean time, here are some important pieces of code from other packages that have been incorporated within the T-Coffee package. These include:

-The Sim algorithm of Huang and Miller that given two sequences computes the N best scoring local alignments.

-The tree reading/computing routines are taken from the ClustalW Package, courtesy of Julie Thompson, Des Higgins and Toby Gibson (Thompson, Higgins, Gibson, 1994, 4673-4680,vol. 22, Nucleic Acid Research).

-The implementation of the algorithm for aligning two sequences in linear space was adapted from Myers and Miller, in CABIOS, 1988, 11-17, vol. 1)

-Various techniques and algorithms have been implemented. Whenever relevant, the source of the code/algorithm/idea is indicated in the corresponding function.

-64 Bits compliance was implemented by Benjamin Sohn, Performance Computing Center Stuttgart (HLRS), Germany

-David Mathog (Caltech) provided many fixes and useful feedback for improving the code and making the whole soft behaving more rationally

Bug Reports and Feedback

-Prof David Jones (UCL) reported and corrected the PDB1K bug (now t_coffee/sap can align PDB sequences longer than 1000 AA).

-Johan Leckner reported several bugs related to the treatment of PDB structures, insuring a consistent behavior between version 1.37 and current ones.

Installation of The T-Coffee Packages

Third Party Packages and On Demand Installations

T-Coffee is a complex package that interacts with many other third part software. If you only want a standalone version of T-Coffee, you may install that package on its own. If you want to use a most sophisticated flavor (3dcoffee, expresso, rcofeee, etc...), the installer will try to install all the third party packages required.

Note that since version 7.56, T-Coffee will use 'on demand' installation and install the third party packages it needs *when* it needs them. This only works for packages not requiring specific licenses and that can be installed by the regular installer. Please let us know if you would like another third party package to be included.

Whenver on-demand installation or automated installation fails because of unforessen system specificities, users should install the third party package manually. This documentation gives some tips we have found useful, but users are encouraged to check the original documentation.

Standard Installation of T-Coffee

Unix

You need to have: gcc, g77, CPAN and an internet connection and your root password (to install SOAP). If you cannot log as root, ask (kindly) your system manager to install SOAP::Lite for you. You may do this before or after the installation of T-Coffee. Even without SOAP you will still be able to use the basic functions of T-Coffee (simplest usage).

```
1. gunzip t_coffee.tar.gz
2. tar -xvf t_coffee.tar
3. cd t_coffee
4. ./install t_coffee
```

This installation will only install the stand alone T-Coffee. If you want to install a

specific mode of T-Coffee, you may try the following commands that will try to gather all the necessary third party packages. Note that a package already found on your system will not be re-installed.

```
./install t_coffee

./install mcoffee

./install 3dcoffee

./install rcoffee

./install psicoffee
```

Or even

```
./install all
```

-All the corresponding executables will be downloaded automatically and installed in

```
$HOME/.t_coffee/plugins
```

-if you executables are in a different location, give it to T-Coffee using the -plugins flag.

-If the installation of any of the companion package fails, you should install it yourself using the provided link (see below) and following the authors instructions.

-If you have not managed to install SOAP::Lite, you can re-install it later (from anywhere) following steps 1-2.

-This procedure attempts 3 things: installing and Compiling T-Coffee (C program), Installing and compiling TMalign (Fortran), Installing and compiling SOAP::Lite(Perl Module).

-If you have never installed SOAP::Lite, CPAN will ask you many questions: say Yes to all

-If everything went well, the procedure has created in the **bin** directory two executables: t_coffee and TMalign (**Make sure these executables are on your $PATH!**)

Microsoft Windows/Cygwin

Install Cygwin

Download The Installer (NOT Cygwin/X)

Click on view to list ALL the packages

Select: gcc-core, make, wget

Optional: ssh, xemacs, nano

Run mkpasswd in Cywin (as requested when you start cygwin)

Install T-Coffee within Cygwin using the Unix procedure

MAC osX, Linux

Make sure you have the Developer's kit installed (compilers and makefile)

Follow the Unix Procedure

CLUSTER Installation

In order to run, T-Coffee must have a value for the http_proxy and for the E-mail. In order to do so you can either:

export the following values:

export http_proxy_4_TCOFFEE="proxy" or "" if no proxy

export EMAIL_4_TCOFFEE="your email"

OR

modify the file ~/.t_coffee/t_coffee_env

OR

add to your command line: t_coffee …. -proxy=<proxy> -email=<email

if you have no proxy: t_coffee … -proxy -email=<email>

If you have PDB installed:

Assuming you have a standard PDB installation in your file system

```
setenv (or export)  PDB_DIR <abs
path>/data/structures/all/pdb/
OR
setenv (or export)  PDB_DIR <abs
path>/structures/divided/pdb/
```

If you do not have PDB installed, don't worry, t_coffee will go and fetch any structure it needs directly from the PDB repository. It will simply be a bit slower than if you had PDB locally.

Installing BLAST for T-Coffee

BLAST is a program that search sequence databases for homologues of a query sequence. It works for proteins and Nucleic Acids. In theory BLAST is just a package like any, but in practice things are a bit more complex. To run well, BLST requires up to date databases (that can be fairly large, like NR or UNIPROT) and a powerful computer.

Fortunately, an increasing number of institutes or companies are now providing BLAST clients that run over the net. It means that all you need is a small program that send your query to the big server and gets the results back. This prevents you from the hassle of installing and maintaining BLAST, but of course it is less private and you rely on the network and the current load of these busy servers.

Thanks to its interaction with BLAST, T-Coffee can gather structures and protein profiles and deliver an alignment significantly more accurate than the default you would get with T-Coffee or any similar method.

Let us go through the various modes available for T-Coffee

Why Do I need BLAST with T-Coffee?

The most accurate modes of T-Coffe scan the databases for templates that they use to align the sequences. There are currently two types of templates for proteins:

structures (PDB) that can be found by a blastp against the PDB database and profiles that can be constructed with eiether a blastp or a psiblast against nr or uniprot.

These templates are automatically built if you use:

```
t_coffee <yourseq> -mode expresso
```

that fetches aand uses pdb templates, or

```
t_coffee <your seq> -mode psicoffee
```

that fetches and uses profile templates, or

```
t_coffee <your seq> -mode accurate
```

that does everything and tries to use the best template. Now that you see why it is useful let's see how to get BLAST up and running, from the easy solution to tailor made ones.

Using the EBI BLAST Client

This is by far the easiest (and the default mode). The perl clients are already incorporated in T-Coffeem and all you need is the SOAP::Lite perl library. In theory, T-Coffee should have already installed this library during the standard

installation. Yet, this requires having toot access. If you did not have it at the time of the installation, or if you need your system administrator to install SOAP::Lite, simply follow the instruction provided on the website:

```
http://search.cpan.org/~byrne/SOAP-Lite-0.60a
```

It really is worth the effort, since the EBI is providing one of the best webservice available around, and most notably, the only public psiblast via a web service.

Another important point is that the EBI requires your E-mail address to process your queries. Normally, T-Coffee should have asked you to provide this address. If you have not, or if you have provided a phony address, you should correct this by directly editing the file

```
~/.t_coffee/email.txt
```

Be Careful! If you provide a fake E-mail, the EBI may suspend the service for all machines associated with your IP address (that could mean your entire lab, or entire institute, or even the entire country or, but I doubt it, the whole universe).

Using the NCBI BLAST Client

The NCBI is the next best alternative. In my hand it was always a bit slower and most of all, it does not incorporate PSI-BLAST (as a web sevice). A big miss. The NCBI web blast client is a small executable that you should install on your system following the instructions given on this link

```
ftp://ftp.ncbi.nih.gov/blast/executables/LATEST
```

Simply go for **netbl,** download the executable that corresponds to your architecture (cygwin users should go for the win executable). Despite all the files that come along the executable blastcl3 is a stand alone executable that you can safely move to your $BIN.

All you will then need to do is to make sure that T-Coffee uses the right client, when you run it.

```
-blast_server=NCBI
```

No need for any E-mail here, but you don't get psiblast, and whenever T-Coffee wants to use it, blastp will be used instead.

Using another Client

You may have your own client (lucky you). If that is so, all you need is to make sure that this client is complient with the blast command line. If your client is named foo.pl, all you need to to is run T-Coffee with

```
-blast_server=CLIENT_foo.pl
```

Foo will be called as if it were blastpgp, and it is your responsability to make sure it can handle the following command line:

```
foo.pl -p <method> -d <db> -i <infile> -o <outfile> -m 7
```

method can either be blastp or psiblast.

infile is a FASTA file

-m7 triggers the XML output. T-Coffee is able to parse both the EBI XML output and the NCBI XML output.

If foo.pl behaves differently, the easiest will probably be to write a wrapper around it so that wrapped_foo.pl behaves like blastpgp

Using a BLAST local version on UNIX

If you have blastpgp installed, you can run it instead of the remote clients by using:

```
-blast_server=LOCAL
```

The documnentation for blastpgp can be found on:

```
www.ncbi.nlm.nih.gov/staff/tao/URLAPI/blastpgp.html
```

and the package is part of the standard BLAST distribution

```
ftp://ftp.ncbi.nih.gov/blast/executables/LATEST
```

Depending on your system, your own skills, your requirements and on more parameters than I have fingers to count, installing a BLAST server suited for your needs can range from a 10 minutes job to an achivement spread over several generations. So at this point, you should roam the NCBI website for suitable information.

If you want to have your own BLAST server to run your own databases, you should know that it is possible to control both the database and the program used by BLAST:

```
-protein_db: will specify the database used by all the psi-blast
modes
-pdb_db: will specify the database used by the pdb modes
```

Using a BLAST local version on Windows/cygwin

For those of you using cygwin, be careful. While cygwin behaves like a UNIX system, the BLAST executable required for cygwin (win32) is expecting WINDOWS path and not UNIX path. This has three important consequences:

1- the ncbi file declaring the Data directory must be:

C:WINDOWS//ncbi.init [at the root of your WINDOWS]

2- the address mentionned with this file must be WINDOWS formated, for instance, on my system:

Data=C:\cygwin\home\notredame\blast\data

3- When you pass database addresses to BLAST, these must be in Windows format:

-protein_db="c:/somewhere/somewhereelse/database"

(using the slash (/) or the andtislash (\) does not matter on new systems but I would reommand against incorporating white spaces.

Installing Other Companion Packages

T-Coffee is meant to interact with as many packages as possible, either for aligning or using predictions. If you type

```
t_coffee
```

You will receive a list of supported packages that looks like the next table. In theory, most of these packages can be installed by T-Coffee

```
****** Pairwise Sequence Alignment Methods:
-----------------------------------------
fast_pair          built_in
exon3_pair         built_in
exon2_pair         built_in
exon_pair          built_in
slow_pair          built_in
proba_pair         built_in
lalign_id_pair     built_in
seq_pair           built_in
externprofile_pair built_in
hh_pair            built_in
profile_pair       built_in
cdna_fast_pair     built_in
cdna_cfast_pair    built_in
clustalw_pair      ftp://www.ebi.ac.uk/pub/clustalw
mafft_pair                                  http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
mafftjtt_pair                               http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
mafftgins_pair                              http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
dialigntx_pair     http://dialign-tx.gobics.de/
dialignt_pair      http://dialign-t.gobics.de/
poa_pair           http://www.bioinformatics.ucla.edu/poa/
probcons_pair      http://probcons.stanford.edu/
```

```
muscle_pair         http://www.drive5.com/muscle/
t_coffee_pair       http://www.tcoffee.org
pcma_pair           ftp://iole.swmed.edu/pub/PCMA/
kalign_pair         http://msa.cgb.ki.se
amap_pair           http://bio.math.berkeley.edu/amap/
proda_pair          http://bio.math.berkeley.edu/proda/
prank_pair          http://www.ebi.ac.uk/goldman-srv/prank/
consan_pair         http://selab.janelia.org/software/consan/

****** Pairwise Structural Alignment Methods:
-------------------------------------------
align_pdbpair       built_in
lalign_pdbpair      built_in
extern_pdbpair      built_in
thread_pair         built_in
fugue_pair                                                  http://www-
cryst.bioc.cam.ac.uk/fugue/download.html
pdb_pair            built_in
sap_pair                                                    http://www-
cryst.bioc.cam.ac.uk/fugue/download.html
mustang_pair        http://www.cs.mu.oz.au/~arun/mustang/
tmalign_pair        http://zhang.bioinformatics.ku.edu/TM-align/

****** Multiple Sequence Alignment Methods:
-------------------------------------------
clustalw_msa        ftp://www.ebi.ac.uk/pub/clustalw
mafft_msa                                   http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
mafftjtt_msa                                http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
mafftgins_msa                               http://www.biophys.kyoto-
u.ac.jp/~katoh/programs/align/mafft/
dialigntx_msa       http://dialign-tx.gobics.de/
dialignt_msa        http://dialign-t.gobics.de/
poa_msa             http://www.bioinformatics.ucla.edu/poa/
probcons_msa        http://probcons.stanford.edu/
muscle_msa          http://www.drive5.com/muscle/
t_coffee_msa        http://www.tcoffee.org
pcma_msa            ftp://iole.swmed.edu/pub/PCMA/
kalign_msa          http://msa.cgb.ki.se
amap_msa            http://bio.math.berkeley.edu/amap/
proda_msa           http://bio.math.berkeley.edu/proda/
prank_msa           http://www.ebi.ac.uk/goldman-srv/prank/

#######   Prediction Methods available to generate Templates
-----------------------------------------------------------
RNAplfold           http://www.tbi.univie.ac.at/~ivo/RNA/
HMMtop              www.enzim.hu/hmmtop/
GOR4                http://mig.jouy.inra.fr/logiciels/gorIV/
wublast_client
http://www.ebi.ac.uk/Tools/webservices/services/wublast
blastpgp_client
http://www.ebi.ac.uk/Tools/webservices/services/blastpgp
===========================================================
```

Installation of PSI-Coffee and Expresso

PSI-Coffee is a mode of T-Coffee that runs a a Psi-BLAST on each of your sequences and makes a multiple profile alignment. If you do not have any structural information, it is by far the most accurate mode of T-Coffee. To use it, you must have SOAP installed so that the EBI BLAST client can run on your system.

It is a bit slow, but really worth it if your sequences are hard to align and if the accuracy of your alignment is important.

To use this mode, try:

```
t_coffee <yoursequence> -mode psicoffee
```

Note that because PSI-BLAST is time consuming, T-Coffee stores the runs in its cache (./tcoffee/cache) so that it does not need to be re-run. It means that if you re-align your sequences (or add a few extra sequences), things will be considerably faster.

If your installation procedure has managed to compile TMalign, and if T-Coffee has access to the EBI BLAST server (or any other server) you can also do the following:

```
t_coffee <yoursequence> -mode expresso
```

That will look for structural templates. And if both these modes are running fine, then you are ready for the best, the "crème de la crème":

```
t_coffee <yoursequence> -mode accurate
```

Installation of M-Coffee

M-Coffee is a special mode of T-Coffee that makes it possible to combine the output of many multiple sequence alignment packages.

Automated Installation

In the T-Coffee distribution, type:

```
./install mcoffee
```

In theory, this command should download and install every required package. If, however, it fails, you should switch to the manual installation (see next).

By default these packages will be in

```
$HOME/.t_coffee/plugins
```

If you want to have these companion packages in a different directory, you can either set the environement variable

```
PLUGINS_4_TCOFFEE=<plugins dir>
```

Or use the command line flag -plugin (over-rides every other setting)

```
t_coffee ... -plugins=<plugins dir>
```

Manual Installation

M-Coffee requires a standard T-Coffee installation (c.f. previous section) and the following packages to be installed on your system:

Package	Where From
ClustalW	can interact with t_coffee
Poa	http://www.bioinformatics.ucla.edu/poa/
Muscle	http://www.drive5.com
ProbCons	http://probcons.stanford.edu/
ProbConsRNA	http://probcons.stanford.edu/
MAFFT	http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/
Dialign-T	http://dialign-t.gobics.de/
Dialign-TX	http://dialign-tx.gobics.de/
PCMA	ftp://iole.swmed.edu/pub/PCMA/
kalign	http://msa.cgb.ki.se
amap	http://bio.math.berkeley.edu/amap/
proda_msa	http://bio.math.berkeley.edu/proda/
prank_msa	http://www.ebi.ac.uk/goldman-srv/prank/

In our hands all these packages where very straightforward to compile and install on a standard cygwin or Linux configuration. Just make sure you have gcc, the C compiler, properly installed.

Once the package is compiled and ready to use, make sure that the executable is on your path, so that t_coffee can find it automatically. Our favorite procedure is to create a bin directory in the home. If you do so, make sure this bin is in your path and fill it with all your executables (this is a standard Unix practice).

If for some reason, you do not want this directory to be on your path, or you want to specify a precise directory containing the executables, you can use:

```
export PLUGINS_4_TCOFFEE=<dir>
```

By default this directory is set to $HOME/.t_coffee/plugins/$OS, but you can override it with the environement variable or using the flag:

```
t_coffee ...-plugins=<dir>
```

If you cannot, or do not want to use a single bin directory, you can set the following environment variables to the absolute path values of the executable you want to use. Whenever they are set, these variables will supersede any other declaration. This is a convenient way to experiment with multiple package versions.

```
POA_4_TCOOFFEE
CLUSTALW_4_TCOFFEE
POA_4_TCOFFEE
TCOFFEE_4_TCOFFEE
MAFFT_4_TCOFFEE
MUSCLE_4_TCOFFEE
DIALIGNT_4_TCOFFEE
PRANK_4_TCOFFEE
DIALIGNTX_4_TCOFFEE
```

For three of these packages, you will need to copy some of the files in a special T-Coffee directory.

```
cp POA_DIR/* ~/.t_coffee/mcoffee/

cp DIALIGN-T/conf/*  ~/.t_coffee/mcoffee

cp DIALIGN-TX/conf/*  ~/.t_coffee/mcoffee
```

Note that the following files are enough for default usage:

```
BLOSUM.diag_prob_t10   BLOSUM75.scr  blosum80_trunc.mat

dna_diag_prob_100_exp_330000  dna_diag_prob_200_exp_110000

BLOSUM.scr             BLOSUM90.scr  dna_diag_prob_100_exp_110000

dna_diag_prob_100_exp_550000  dna_diag_prob_250_exp_110000

BLOSUM75.diag_prob_t2  blosum80.mat  dna_diag_prob_100_exp_220000
```

```
dna_diag_prob_150_exp_110000  dna_matrix.scr
```

If you would rather have the mcoffee directory in some other location, set the MCOFFEE_4_TCOFFEE environement variable to the propoer directory:

```
setenv MCOFFEE_4_TCOFFEE <directory containing mcoffee files>
```

Installation of APDB and iRMSD

APDB and iRMSD are incorporated in T-Coffee. Once t_coffee is installed, you can invoque these programs by typing:

```
t_coffee -other_pg apdb
t_coffee -other_pg irmsd
```

Installation of tRMSD

tRMSD comes along with t_coffee but it also requires the package phylip in order to be functional. Phylip can be obtained from:

```
Package           Function
=================================================
-------------------------------------------------
Phylip            Phylogenetic tree computation
                  evolution.genetics.washington.edu/phylip.html
-------------------------------------------------
```

```
t_coffee -other_pg trmsd
```

Installation of seq_reformat

Seq_reformat is a reformatting package that is part of t_coffee. To use it (and see the available options), type:

```
t_coffee -other_pg seq_reformat
```

Installation of extract_from_pdb

Extract_from_pdb is a PDB reformatting package that is part of t_coffee. To use it (and see the available options), type.

```
t_coffee -other_pg apdb -h
```

Extract_from_pdb requires wget in order to automatically fetch PDB structures.

Installation of 3D-Coffee/Expresso

3D-Coffee/Expresso is a special mode of T-Coffee that makes it possible to combine sequences and structures. The main difference between Expresso and 3D-Coffee is that Expresso fetches the structures itself.

Automated Installation

In the T-Coffee distribution, type:

```
./install expresso

OR

./install 3dcoffee
```

In theory, this command should download and install every required package (**except fugue**). If, however, it fails, you should switch to the manual installation (see next).

Manual Installation

In order to make the most out of T-Coffee, you will need to install the following packages (make sure the executable is named as indicated below):

```
Package           Function
================================================
------------------------------------------------
wget              3DCoffee
                  Automatic Downloading of Structures
------------------------------------------------
sap               structure/structure comparisons
(obtain it from W. Taylor, NIMR-MRC).
------------------------------------------------
TMalign           zhang.bioinformatics.ku.edu/TM-align/
------------------------------------------------
mustang           www.cs.mu.oz.au/~arun/mustang/
------------------------------------------------
wublastclient     www.ebi.ac.uk/Tools/webservices/clients/wublast
------------------------------------------------
Blast             www.ncbi.nih.nlm.gov
------------------------------------------------
Fugue*            protein to structure alignment program
                  http://www-cryst.bioc.cam.ac.uk/fugue/download.html
                  ***NOT COMPULSORY***
```

Once the package is installed, make sure make sure that the executable is on your path, so that t_coffee can find it automatically.

The wublast client makes it possible to run BLAST at the EBI without having to install any database locally. It is an ideal solution if you are only using expresso occasionally.

Installing Fugue for T-Coffee

Uses a standard fugue installation. You only need to install the following packages:

joy, melody, fugueali, sstruc, hbond

If you have root privileges, you can install the common data in:

cp fugue/classdef.dat /data/fugue/SUBST/classdef.dat

otherwise

Setenv MELODY_CLASSDEF=<location>

Setenv MELODY_SUBST=fugue/allmat.dat

All the other configuration files must be in the right location.

Installation of R-Coffee

R-Coffee is a special mode able to align RNA sequences while taking into account their secondary structure.

Automated Installation

In the T-Coffee distribution, type:

```
./install rcoffee
```

In theory, this command should download and install every required package (except **consan**). If, however, it fails, you should switch to the manual installation (see next).

Manual Installation

R-Coffee only requires the package Vienna to be installed, in order to compute multiple sequence alignments. To make the best out of it, you should also have all the packages required by M-Coffee

```
Package           Function
================================================
------------------------------------------------
consan            R-Coffee
                  Computes highly accurate pairwise Alignments
                  ***NOT COMPULSORY***
                  selab.janelia.org/software/consan/
```

```
-------------------------------------------------
RNAplfold         Computes RNA secondary Structures
                  www.tbi.univie.ac.at/~ivo/RNA/
-------------------------------------------------
probconsRNA       probcons.stanford.edu/

-------------------------------------------------
M-Coffee          T-Coffee and the most common MSA Packages
                  (cf M-Coffee in this installation guide)
```

Installing ProbbonsRNA for R-Coffee

Follow the installation procedure, but make sure you rename the probcons executable into probconsRNA.

Installing Consan for R-Coffee

In order to insure a proper interface beween consan and R-Coffee, you must make sure that the file mix80.mod is in the directory ~/.t_coffee/mcoffee or in the mcoffee directory otherwise declared.

Quick Start

We only give you the very basics here. Please use the Tutorial for more detailed information on how to use our tools.

IMPORTANT: All the files mentionned here (sampe_seq...) can be found in the example directory of the distribution.

T-COFFEE

Write your sequences in the same file (Swiss-prot, Fasta or Pir) and type.

```
PROMPT: t_coffee sample_seq1.fasta
```

This will output two files:

sample_seq1.aln: your Multiple Sequence Alignment

sample_seq1.dnd: The Guide tree (newick Format)

IMPORTANT:

In theory nucleic acids should be automatically detected and the default methods should be adapted appropriately. However, sometimes this may fail, either because the sequences are too short or contain too many ambiguity codes.

When this happens, you are advised to explicitly set the type of your sequences

NOTE: the –mode=dna is not needed or supported anymore

```
PROMPT: t_coffee sample_dnaseq1.fasta -type=dna
```

M-Coffee

M-Coffee is a Meta version of T-Coffee that makes it possible to combine the output of at least eight packages (Muscle, probcons, poa, dialignT, mafft, clustalw, PCMA and T-Coffee).

If all these packages are already installed on your machine. You must:

1-set the following environment variables

```
export POA_DIR=[absolute path of the POA installation dir]

export DIALIGNT_DIR=[Absolute path of the DIALIGN-T/conf
```

Once this is done, write your sequences in a file and run: same file (Swiss-prot, Fasta or Pir) and type.

```
PROMPT: t_coffee sample_seq1.fasta -mode mcoffee
```

If the program starts complaining one package or the other is missing, this means you will have to go the hard way and install all these packages yourself... Proceed to the M-Coffee section for more detailed instructions.

Expresso

If you have installed the EBI wublast.pl client, Expresso will BLAST your sequences against PDB, identify the best targets and use these to align your proteins.

```
PROMPT: t_coffee sample_seq1.fasta -mode expresso
```

If you did not manage to install all the required structural packages for Expresso, like Fugue or Sap, you can still run expresso by selecting yourself the structural packages you want to use. For instance, if you'd rather use TM-Align than sap, try:

```
PROMPT: t_coffee sample_seq1.fasta -template_file EXPRESSO -method
TMalign_pair
```

R-Coffee

R-Coffee can be used to align RNA sequences, using their RNApfold predicted secondary structures. The best results are obtained by using the consan pairwise method. If you have consan installed:

```
t_coffee sample_rnaseq1.fasta -special_mode rcoffee_consan
```

This will only work if your sequences are short enough (less than 200 nucleotides).

A good alternative is the rmcoffee mode that will run Muscle, Probcons4RNA and MAfft and then use the secondary structures predicted by RNApfold.

```
PROMPT: t_coffee sample_rnaseq1.fasta –mode mrcoffee
```

If you want to decide yourself which methods should be combined by R-Coffee, run:

```
PROMPT: t_coffee sample_rnaseq1.fasta –mode rcoffee -method
lalign_id_pair slow_pair
```

iRMSD and APDB

All you need is a file containing the alignment of sequences with a known structure. These sequences must be named according to their PDB ID, followed by the chain index (1aabA for instance). All the sequences do not need to have a known structure, but at least two need to have it.

Given the alignment:

```
PROMPT: t_coffee –other_pg irmsd -aln 3d_sample4.aln
```

tRMSD

tRMSD is a structure based clustering method using the iRMSD to drive the clustering. The T-RMSD supports all the parameters supported by iRMSD or APDB.

```
PROMPT: t_coffee –other_pg trmsd -aln 3d_sample5.aln -template_file
3d_sample5.template_list
```

3d_sample5.aln is a multiple alignment in which each sequence has a known structure. The file 3d_sample5.template_list is a fasta like file declaring the structure associated with each sequence, in the form:

```
> <seq_name> _P_ <PDB structure file or name>
```

```
******* 3d_sample5.template_list ********
>2UWI-3A _P_ 2UWI-3.pdb
```

```
>2UWI-2A _P_ 2UWI-2.pdb
>2UWI-1A _P_ 2UWI-1.pdb
>2HEY-4R _P_ 2HEY-4.pdb
...
************************************
```

The program then outputs a series of files

```
Template Type: [3d_sample5.template_list] Mode Or File: [3d_sample5.template_list]
[Start]
         [Sample Columns][TOT=   51][100 %][ELAPSED TIME:    0 sec.]
         [Tree Cmp][TOT=   13][ 92 %][ELAPSED TIME:    0 sec.]
 #### File Type=   TreeList Format=     newick Name= 3d_sample5.tot_pos_list
 #### File Type=       Tree Format=     newick Name= 3d_sample5.struc_tree10
 #### File Type=       Tree Format=     newick Name= 3d_sample5.struc_tree50
 #### File Type=       Tree Format=     newick Name= 3d_sample5.struc_tree100
 #### File Type= Colored MSA Format= score_html Name= 3d_sample5.struc_tree.html
```

3d_sample5.tot_pos_list is a list of the tRMSD tree associated with every position.

3d_sample5.struc_tree100 is a consensus tree (phylip/consense) of the trees contained in the previous file. **This file is the default output**

3d_sample5.struc_tree10 is a consensus tree (phylip/consense) of the 10% trees having the higest average agreement with the rest

3d_sample5.struc_tree10 is a consensus tree (phylip/consense) of the 50% trees having the higest average agreement with the rest

3d_sample5.html is a colored version of the output showing in red the positions that give the highest support to 3d_sample5.struc_tree100

MOCCA

Write your sequences in the same file (Swiss-prot, Fasta or Pir) and type.

```
PROMPT: t_coffee -other_pg mocca sample_seq1.fasta
```

This command output one files (<your sequences>.mocca_lib) and starts an interactive menu.

Recent Modifications

Warning: This log of recent modifications is not as thorough and accurate as it should be.

-5.80 Novel assembly algorithm (linked_pair_wise) and the primary library is now made of probcons style pairwise alignments (proba_pair)

-4.30 and upward: the FAQ has moved into a new tutorial document

-4.30 and upward: -in has will be deprecated and replaced by the flags: -profile,-method,-aln,-seq,-pdb

-4.02: -mode=dna is still available but not any more needed or supported. Use type=protein or dna if you need to force things

-3.28: corrected a bug that prevents short sequences from being correctly aligned

-Use of @ as a separator when specifying methods parameters

-The most notable modifications have to do with the structure of the input. From version 2.20, all files must be tagged to indicate their nature (A: alignment, S: Sequence, L: Library…). We are becoming stricter, but that's for your own good…

Another important modification has to do with the flag -matrix: it now controls the matrix being used for the computation

Reference Manual

This reference manual gives a list of all the flags that can be used to modify the behavior of T-Coffee. For your convenience, we have grouped them according to their nature. To display a list of all the flags used in the version of T-Coffee you are using (along with their default value), type:

```
PROMPT: t_coffee
```

Or

```
PROMPT: t_coffee -help
```

Or

```
PROMPT: t_coffee -help -in
```

Or any other parameter

Environment Variables

It is possible to modify T-Coffee's behavior by setting any of the following environement variables. On the bash shell, use export VAR="value". On the cshell, use set $VAR="xxx"

http_proxy_4_TCOFFEE

Sets the http_proxy and HTTP_proxy values used by T-Coffee.

These values get supersede http_proxy and HTTP_proxy. http_proxy_4_TCOFFEE gets superseded by the command line values (-proxy and -email)

If you have no proxy, just set this value to an empty string.

email_4_TCOFFEE

Set the E-mail values provided to web services called upon by T-Coffee. Can be over-riden by the flag *-email*.

DIR_4_TCOFFEE

By default this variable is set to $HOME/.t_coffee. This is where T-Coffee expects to find its cache, tmp dir and possibly any temporary data stored by the program.

TMP_4_TCOFFEE

By default this variable is set to $HOME/.t_coffee/tmp. This is where T-Coffee stores temporary files.

CACHE_4_TCOFFEE

By default this variable is set to $HOME/.t_coffee/cache. This is where T-Coffee stores any data expensive to obtain: pdb files, sap alignments....

PLUGINS_4_TCOFFEE

By default all the companion packages are searched in the directory DIR_4_TCOFFEE/plugins/<OS>. This variable overrides the default. This variable can also be overriden by the *-plugins* T-Coffee flag

NO_ERROR_REPORT_4_TCOFFEE

By default this variable is no set. Set it if you do not want the program to generate a verbose error output file (useful for running a server).

PDB_DIR

Indicate the location of your local PDB installation.

NO_WARNING_4_TCOFFEE

Suppresses all the warnings.

UNIQUE_DIR_4_TCOFFEE

Sets:

DIR_4_TCOFFEE

CACHE_4_TCOFFEE

TMP_4_TCOFFEE

PLUGINS_4_TCOFFEE

To the same unique value. The string MUST be a valid directory

Setting up the T-Coffee environment variables

T-Coffee can have its own environment file. This environment is kept in a file named $HOME/.t_coffee/t_coffee_env and can be edited. The value of any legal variable can be

modified through that file. For instance, here is an example of a configuration file when not requiring a proxy.

```
http_proxy_4_TCOFFEE=
EMAIL_4_TCOFFEE=cedric.notredame@europe.com
```

IMPORTANT:

-proxy, -email >> t_coffee_env >> env

Well Behaved Parameters

Separation

You can use any kind of separator you want (i.e. ,; <space>=). The syntax used in this document is meant to be consistent with that of ClustalW. However, in order to take advantage of the automatic filename compleation provided by many shells, you can replace "=" and "," with a space.

Posix

T-Coffee is not POSIX compliant.

Entering the right parameters

There are many ways to enter parameters in T-Coffee, see the -parameter flag in

Parameters Priority

In general you will not need to use these complicated parameters. Yet, if you find yourself typing long command lines on a regular basis, it may be worth reading this section.

One may easily feel confused with the various manners in which the parameters can be passed to t_coffee. The reason for these many mechanisms is that they allow several levels of intervention. For instance, you may install t_coffee for all the users and decide that the defaults we provide are not the proper ones… In this case, you will need to make your own t_coffee_default file.

Later on, a user may find that he/she needs to keep re-using a specific set of parameters, different from those in t_coffee_default, hence the possibility to write an extra parameter file: parameters. In summary:

-parameters > prompt parameters > -t_coffee_defaults > -mode

This means that *-parameters* supersede all the others, while parameters provided via *-special mode* are the weakest.

Parameters Syntax

No Flag

If no flag is used <*your sequence*> must be the first argument. See format for further information.

```
PROMPT: t_coffee sample_seq1.fasta
```

Which is equivalent to

```
PROMPT: t_coffee Ssample_seq1.fasta
```

When you do so, **sample_seq1** is used as a name prefix for every file the program outputs.

-parameters

Usage: -parameters=parameters_file

Default: no parameters file

Indicates a file containing extra parameters. Parameters read this way behave as if they had been added on the right end of the command line that they either supersede(one value parameter) or complete (list of values). For instance, the following file (parameter.file) could be used

```
*******sample_param_file.param********
      -in=Ssample_seq1.fasta,Mfast_pair
      -output=msf_aln
************************************
```

Note: This is one of the exceptions (with –infile) where the identifier tag (S,A,L,M…) can be omitted. Any dataset provided this way will be assumed to be a sequence (S). These exceptions have been designed to keep the program compatible with ClustalW.

Note: This parameter file can ONLY contain valid parameters. Comments are not allowed. Parameters passed this way will be checked like normal parameters.

Used with:

```
PROMPT: t_coffee -parameters=sample_param_file.param
```

Will cause t_coffee to apply the fast_pair method onto to the sequences contained in sample_seq.fasta. If you wish, you can also pipe these arguments into t_coffee, by naming the parameter file "stdin" (as a rule, any file named stdin is expected to receive its content via the stdin)

```
cat sample_param_file.param  | t_coffee -parameters=stdin
```

-t_coffee_defaults

Usage: -t_coffee_defaults=<file_name>

Default: not used.

This flag tells the program to use some default parameter file for t_coffee. The format of that file is the same as the one used with -parameters. The file used is either:

1. <file name> if a name has been specified
2. **~/.t_coffee_defaults** if no file was specified
3. The file indicated by the environment variable **TCOFFEE_DEFAULTS**

-mode

Usage: -mode= hard coded mode

Default: not used.

It indicates that t_coffee will use some hard coded parameters. These include:

quickaln: very fast approximate alignment

dali: a mode used to combine dali pairwise alignments

evaluate: defaults for evaluating an alignment

3dcoffee: runs t_coffee with the 3dcoffee parameterization

Other modes exist that are not yet fully supported

-score [Deprecated]

Usage: -score

Default: not used

Toggles on the evaluate mode and causes t_coffee to evaluates a precomputed alignment provided via **-infile=<alignment>**. The flag **-output** must be set to an appropriate format (i.e. -output=score_ascii, score_html or score_pdf). A better default parameterization is obtained when using the flag **-mode=evaluate.**

-evaluate

Usage: -evaluate

Default: not used

Replaces –score. This flag toggles on the evaluate mode and causes t_coffee to evaluates a pre-computed alignment provided via **-infile=<alignment>**. The flag **-output** must be set to an appropriate format (i.e. -output=score_ascii, score_html or score_pdf).

The main purpose of –evaluate is to let you control every aspect of the evaluation. Yet it is advisable to use pre-defined parameterization: **mode=evaluate.**

```
PROMPT: t_coffee –infile=sample_aln1.aln -mode=evaluate
```

```
PROMPT: t_coffee -infile=sample_seq1.aln -in  Lsample_lib1.tc_lib
-mode=evaluate
```

-convert [cw]

Usage: -convert

Default: turned off

Toggles on the conversion mode and causes T-Coffee to convert the sequences, alignments, libraries or structures provided via the **-infile** and **-in** flags. The output format must be set via the **-output** flag. This flag can also be used if you simply want to compute a library (i.e. you have an alignment and you want to turn it into a library).

This flag is ClustalW compliant.

-do_align [cw]

Usage: -do_align

Default: turned on

Special Parameters

-version

Usage: -version

Default: not used

Returns the current version number

-proxy

Usage: -proxy=<proxy>

Default: not used

Sets the proxy used by HTTP_proxy AND http_proxy. Setting with the propmpt supersedes ANY other setting.

Note that if you use no proxy, you should set

-proxy

-email

Usage: -email=<email>

Default: not used

Sets your email value as provided to web services

-check_configuration

Usage: -check_configuration

Default: not used

Checks your system to determine whether all the programs T-Coffee can interact with are installed.

-cache

Usage: -cache=<use, update, ignore, <filename>>

Default: -cache=use

By default, t_coffee stores in a cache directory, the results of computationally expensive (structural alignment) or network intensive (BLAST search) operations.

-update

Usage: -update

Default: turned off

Causes a ***wget*** access that checks whether the t_coffee version you are using needs updating.

-full_log

Usage: -full_log=<filename>

Default: turned off

Causes t_coffee to output a full log file that contains all the input/output files.

-plugins

Usage: -plugins=<dir>

Default: default

Specifies the directory in which the companion packages (other multiple aligners used by M-Coffee, structural aligners, etc…) are kept as an alternative, you can also set the environment variable PLUGINS_4_TCOFFEE

The default is ~/.t_coffee/plugins/

-other_pg

Usage: -other_pg=<filename>

Default: turned off

Some rumours claim that Tetris is embedded within T-Coffee and could be ran using some special set of commands. We wish to deny these rumours, although we may admit that several interesting reformatting programs are now embedded in t_coffee and can be ran through the –other_pg flag.

```
PROMPT: t_coffee -other_pg=seq_reformat

PROMPT: t_coffee -other_pg=unpack_all

PROMPT: t_coffee -other_pg=unpack_extract_from_pdb
```

Input

Sequence Input

-infile [cw]

To remain compatible with ClustalW, it is possible to indicate the sequences with this flag

```
PROMPT: t_coffee -infile=sample_seq1.fasta
```

Note: Common multiple sequence alignments format constitute a valid input format.

Note: T-Coffee automatically removes the gaps *before* doing the alignment. This behaviour is different from that of ClustalW where the gaps are kept.

-in (Cf –in from the Method and Library Input section)

-get_type

Usage: -get_type

Default: turned off

Forces t_coffee to identify the sequences type (PROTEIN, DNA).

-type [cw]

Usage: -type=DNA ¦ PROTEIN¦ DNA_PROTEIN

Default: -type=<automatically set>

This flag sets the type of the sequences. If omitted, the type is guessed automatically. This flag is compatible with ClustalW.

Warning: In case of low complexity or short sequences, it is recommended to set the type manually.

-seq

Usage: -seq=[<P,S><name>,]

Default: none

-seq is now the recommended flag to provide your sequences. It behaves mostly like the -in flag.

-seq_source

Usage: -seq_source=<ANY or _LS or LS >

Default: ANY.

You may not want to combine all the provided sequences into a single sequence list. You can do by specifying that you do not want to treat all the –in files as potential sequence sources.

-seq_source=_LA indicates that neither sequences provided via the A (Alignment) flag or via the L (Library flag) should be added to the sequence list.

-seq_source=S means that only sequences provided via the S tag will be considered. All the other sequences will be ignored.

Note: This flag is mostly designed for interactions between T-Coffee and T-CoffeeDPA (the large scale version of T-Coffee).

Structure Input

-pdb

Usage: -pdb=<pdbid1>,<pdbid2>…[Max 200]

Default: None

Reads or fetch a pdb file. It is possible to specify a chain or even a sub-chain:

```
PDBID(PDB_CHAIN)[opt] (FIRST,LAST)[opt]
```

It is also possible to input structures via the –in flag. In that case, you will need to use the TAG identifier:

```
-in Ppdb1 Ppdb2…
```

Tree Input

-usetree

Usage: -usetree=<tree file>

Default: No file specified

Format: newick tree format (ClustalW Style)

This flag indicates that rather than computing a new dendrogram, t_coffee must use a pre-computed one. The tree files are in phylips format and compatible with ClustalW. In most cases, using a pre-computed tree will halve the computation time required by t_coffee. It is also possible to use trees output by ClustalW, Phylips and any other program.

Structures, Sequences Methods and Library Input via the -in Flag

The -in Flag and its Identifier TAGS

<-in> is the real grinder of T-Coffee. Sequences, methods and alignments all pass through so that T-Coffee can turn it all into a single list of constraints (the library). Everything is done automatically with T-Coffee going through each file to extract the sequences it contains. The methods are then applied to the sequences. Pre-compiled constraint list can also be provided. Each file provided via this flag must be preceded with a symbol (Identifier TAG) that indicates its nature to T-Coffee. The TAGs currently supported are the following:

P PDB structure
S for sequences (use it as well to treat an MSA as unaligned sequences)

M	**Methods used to build the library**
L	**Pre-computed T-Coffee library**
A	**Multiple Alignments that must be turned into a Library**
X	**Substitution matrices.**
R	**Profiles. This is a legal multiple alignments that will be treated as single sequences (the sequences it contains will not be realigned).**

If you do not want to use the TAGS, you will need to use the following flags in replacement of -in. Do not use the TAGS when using these flags:

-aln	**Alignments**	**(A)**
-profile	**Profiles**	**(R)**
-method	**Method**	**(M)**
-seq	**Sequences**	**(S)**
-lib	**Libraries**	**(L)**

-in

Usage: -in=[<P,S,A,L,M,X><name>,]

Default: -in=Mlalign_id_pair,Mclustalw_pair

Note: -in can be replaced with the combined usage of -aln, iprofile, .pdb, .lib, -method.

See the box for an explanation of the -in flag. The following argument passed via -in

```
PROMPT: t_coffee -
in=Ssample_seq1.fasta,Asample_aln1.aln,Asample_aln2.msf,Mlalign_id
_pair,Lsample_lib1.tc_lib -outfile=outaln
```

This command will trigger the following chain of events:

1-Gather all the sequences

Sequences within all the provided files are pooled together. Format recognition is automatic. Duplicates are removed (if they have the same name). Duplicates in a single file are only tolerated in FASTA format file, although they will cause sequences to be renamed.

In the above case, the total set of sequences will be made of sequences contained in sequences1.seq, alignment1.aln, alignment2.msf and library.lib, plus the sequences initially gathered by -infile.

2-Turn alignments into libraries

alignment1.aln and alignment2.msf will be read and turned into libraries. Another library will be produced by applying the method lalign_id_pair to the set of sequences previously obtained (1). The final library used for the alignment will be the combination of all this information.

Note as well the following rules:

1-Order: The order in which sequences, methods, alignments and libraries are fed in is irrelevant.

2-Heterogeneity: There is no need for each element (A, S, L) to contain the same sequences.

3-No Duplicate: Each file should contain only one copy of each sequence. Duplicates are only allowed in FASTA files but will cause the sequences to be renamed.

4-Reconciliation: If two files (for instance two alignments) contain different versions of the same sequence due to an indel, a new sequence will be reconstructed and used instead:

```
aln 1:hgab1    AAAAABAAAAA

aln 2:hgab1    AAAAAAAAAACCC
```

will cause the program to reconstruct and use the following sequence

```
hgab1    AAAAABAAAAACCC
```

This can be useful if you are trying to combine several runs of blast, or structural information where residues may have been deleted. However substitutions are forbidden. If two sequences with the same name cannot be merged, they will cause the program to exit with an information message.

5-Methods: The method describer can either be built in (See ### for a list of all the available methods) or be a file describing the method to be used. The exact syntax is provided in part 4 of this manual.

6-Substitution Matrices: If the method is a substitution matrix (X) then no other type of information should be provided. For instance:

```
PROMPT: t_coffee sample_seq1.fasta -in=Xpam250mt  -gapopen=-10  -
gapext=-1
```

This command results in a progressive alignment carried out on the sequences in seqfile. The procedure does not use any more the T-Coffee concistency based algorithm, but switches to a standard progressive alignment algorithm (like ClustalW or Pileup) much less accurate. In this context, appropriate gap penalties should be provided. The matrices are in the file source/matrices.h. Add-Hoc matrices can also be provided by the user (see the matrices format section at the end of this manual).

Warning: Xmatrix does not have the same effect as using the -matrix flag. The -matrix defines the matrix that will be used while compiling the library while the Xmatrix defines the matrix used when assembling the final alignment.

Profile Input

-profile

Usage: -profile=[<name>,] maximum of 200 profiles.

Default: no default

This flag causes T-Coffee to treat multiple alignments as a single sequences, thus making it possible to make multiple profile alignments. The profile-profile alignment is controlled by -profile_mode and -profile_comparison. When provided with the **-in** flag, profiles must be preceded with the letter R.

```
PROMPT: t_coffee -profile sample_aln1.aln,sample_aln2.aln -
outfile=profile_aln

PROMPT: t_coffee -in
Rsample_aln1.aln,Rsample_aln2.aln,Mslow_pair,Mlalign_id_pair -
outfile=profile_aln
```

Note that when using –template_file, the program will also look for the templates associated with the profiles, even if the profiles have been provided as templates themselves (however it will not look for the template of the profile templates of the profile templates…)

-profile1 [cw]

Usage: -profile1=[<name>], one name only

Default: no default

Similar to the previous one and was provided for compatibility with ClustalW.

-profile2 [cw]

Usage: -profile1=[<name>], one name only

Default: no default

Similar to the previous one and was provided for compatibility with ClustalW.

Alignment Computation

Library Computation: Methods

-lalign_n_top

Usage: -lalign_n_top=<Integer>

Default: -lalign_n_top=10

Number of alignment reported by the local method (lalign).

-align_pdb_param_file

Unsuported

-align_pdb_hasch_mode

Unsuported

Library Computation: Extension

-lib_list [Unsupported]

Usage: -lib_list=<filename>

Default:unset

Use this flag if you do not want the library computation to take into account all the possible pairs in your dataset. For instance

Format:

```
2 Name1 name2
2 Name1 name4
3 Name1 Name2 Name3…
```

(the line 3 would be used by a multiple alignment method).

-do_normalise

Usage: -do_normalise=<0 or a positive value>

Default:-do_normalise=1000

Development Only

When using a value different from 0, this flag sets the score of the highest scoring pair to 1000.

-extend

Usage: -extend=<0,1 or a positive value>

Default:-extend=1

Development Only

When turned on, this flag indicates that the library extension should be carried out when performing the multiple alignment. If **-extend =0**, the extension is not made, if it is set to 1, the extension is made on all the pairs in the library. If the extension is set to another positive value, the extension is only carried out on pairs having a weight value superior to the specified limit.

-extend_mode

Usage: -extend=<string>

Default:-extend=very_fast_triplet

Warning: Development Only

Controls the algorithm for matrix extension. Available modes include:

relative_triplet	Unsupported
g_coffee	Unsupported
g_coffee_quadruplets	Unsupported
fast_triplet	Fast triplet extension
very_fast_triplet	slow triplet extension, limited to the **-max_n_pair** best sequence pairs when aligning two profiles
slow_triplet	Exhaustive use of all the triplets
mixt	Unsupported
quadruplet	Unsupported
test	Unsupported
matrix	Use of the matrix **-matrix**
fast_matrix	Use of the matrix **-matrix**. Profiles are turned into consensus

-max_n_pair

Usage: -max_n_pair=<integer>

Default:-extend=10

Development Only

Controls the number of pairs considered by the **-extend_mode**=very_fast_triplet. Setting it to 0 forces all the pairs to be considered (equivalent to **-extend_mode**=slow_triplet).

-seq_name_for_quadruplet

Usage: Unsupported

-compact

Usage: Unsupported

-clean

Usage: Unsupported

-maximise

Usage: Unsupported

-do_self

Usage: Flag -do_self

Default: No

This flag causes the extension to carried out within the sequences (as opposed to between sequences). This is necessary when looking for internal repeats with Mocca.

-seq_name_for_quadruplet

Usage: Unsupported

-weight

Usage: -weight=<winsimN, sim or sim_<matrix_name or matrix_file> or <integer value>

Default: -weight=sim

Weight defines the way alignments are weighted when turned into a library. Overweighting can be obtained with the OW<X> weight mode.

winsimN indicates that the weight assigned to a given pair will be equal to the percent identity within a window of 2N+1 length centered on that pair. For instance winsim10 defines a window of 10 residues around the pair being considered. This gives its own weight to each residue in the output library. In our hands, this type of weighting scheme has not provided any significant improvement over the standard sim value.

```
PROMPT: t_coffee sample_seq1.fasta -weight=winsim10 -
out_lib=test.tc_lib
```

sim indicates that the weight equals the average identity within the sequences containing the matched residues.

OW<X> Will cause the sim weight to be multiplied by X

sim_matrix_name indicates the average identity with two residues regarded as identical when their substitution value is positive. The valid matrices names are in *matrices.h (pam250mt)* .Matrices not found in this header are considered to be filenames. See the format section for matrices. For instance, *-weight=sim_pam250mt* indicates that the grouping used for similarity will be the set of classes with positive substitutions.

```
PROMPT: t_coffee sample_seq1.fasta -weight=winsim10 -
out_lib=test.tc_lib
```

Other groups include

sim_clustalw_col (categories of clustalw marked with :)

sim_clustalw_dot (categories of clustalw marked with .)

Value indicates that all the pairs found in the alignments must be given the same weight equal to value. This is useful when the alignment one wishes to turn into a library must be given a pre-specified score (for instance if they come from a structure super-imposition program). Value is an integer:

```
PROMPT: t_coffee sample_seq1.fasta -weight=1000 -
out_lib=test.tc_lib
```

Tree Computation

-distance_matrix_mode

Usage: -distance_matrix_mode=<slow, fast, very_fast>

Default: very_fast

This flag indicates the method used for computing the distance matrix (distance between every pair of sequences) required for the computation of the dendrogram.

Slow The chosen dp_mode using the extended library,

fast: The fasta dp_mode using the extended library.

very_fast The fasta dp_mode using blosum62mt.

ktup Ktup matching (Muscle kind)

aln Read the distances on a precomputed MSA

-quicktree [CW]

Usage: -quicktree

Description: Causes T-Coffee to compute a fast approximate

guide tree

This flag is kept for compatibility with ClustalW. It indicates that:

```
PROMPT: t_coffee sample_seq1.fasta -distance_matrix_mode=very_fast

PROMPT: t_coffee sample_seq1.fasta -quicktree
```

Controlling Alignment Computation

Most parameters in this section refer to the alignment mode fasta_pair_wise and cfatsa_pair_wise. When using these alignment modes, things proceed as follow:
1-Sequences are recoded using a degenerated alphabet provided with <-sim_matrix>
2-Recoded sequences are then hashed into ktuples of size <-ktup>
3-Dynamic programming runs on the <-ndiag> best diagonals whose score is higher than <-diag_threshold>, the way diagonals are scored is controlled via <-diag_mode> .
4-The Dynamic computation is made to optimize either the library scoring scheme (as defined by the -in flag) or a substitution matrix as provided via the -matrix flag. The penalty scheme is defined by -gapopen and -gapext. If -gapopen is undefined, the value defined in -cosmetic_penalty is used instead.
5-Terminal gaps are scored according to -tg_mode

-dp_mode

Usage: -dp_mode=<string>

Default: -dp_mode=cfasta_fair_wise

This flag indicates the type of dynamic programming used by the program:

```
PROMPT: t_coffee sample_seq1.fasta -dp_mode myers_miller_pair_wise
```

gotoh_pair_wise: implementation of the gotoh algorithm (quadratic in memory and time)

myers_miller_pair_wise: implementation of the Myers and Miller dynamic programming algorithm (quadratic in time and linear in space). This algorithm is recommended for very long sequences. It is about 2 times slower than gotoh and only accepts *tg_mode=1or 2 (i.e. gaps penalized for opening)*.

fasta_pair_wise: implementation of the fasta algorithm. The sequence is hashed, looking for *ktuples* words. Dynamic programming is only carried out on the *ndiag* best scoring diagonals. This is much faster but less accurate than the two previous. This mode is controlled by the parameters **-ktuple, -diag_mode and -ndiag**

cfasta_pair_wise: c stands for checked. It is the same algorithm. The dynamic programming is made on the *ndiag* best diagonals, and then on the 2*ndiags, and so on until the scores converge. Complexity will depend on the level of divergence of the sequences, but will usually be L*log(L), with an accuracy comparable to the two first mode (this was checked on BaliBase). This mode is controlled by the parameters **-ktuple, -diag_mode and –ndiag**

Note: Users may find by looking into the code that other modes with fancy names exists (viterby_pair_wise…) Unless mentioned in this documentation, these modes are not supported.

-ktuple

Usage: -ktuple=<value>

Default: -ktuple=1 or 2

Indicates the ktuple size for cfasta_pair_wise dp_mode and fasta_pair_wise. It is set to 1 for proteins, and 2 for DNA. The alphabet used for protein can be a degenerated version, set with **-sim_matrix.**.

-ndiag

Usage: -ndiag=<value>

Default: -ndiag=0

Indicates the number of diagonals used by the *fasta_pair_wise* algorithm (cf **-dp_mode**). When **-ndiag=0**, n_diag=Log (length of the smallest sequence)+1.

When –ndiag and –diag_threshold are set, diagonals are selected if and only if they fulfill both conditions.

-diag_mode

Usage: -diag_mode=<value>

Default: -diag_mode=0

Indicates the manner in which diagonals are scored during the fasta hashing.

0: indicates that the score of a diagonal is equal to the sum of the scores of the exact matches it contains.

1 indicates that this score is set equal to the score of the best uninterrupted segment (useful when dealing with fragments of sequences).

-diag_threshold

Usage: -diag_threshold=<value>

Default: -diag_threshold=0

Sets the value of the threshold when selecting diagonals.

0: indicates that –ndiag should be used to select the diagonals (cf –ndiag section).

-sim_matrix

Usage: -sim_matrix=<string>

Default: -sim_matrix=vasiliky

Indicates the manner in which the amino acid alphabet is degenerated when hashing in the fasta_pairwise dynamic programming. Standard ClustalW matrices are all valid. They are used to define groups of amino acids having positive substitution values. In T-Coffee, the default is a 13 letter grouping named Vasiliky, with residues grouped as follows:

```
rk, de, qh, vilm, fy (other residues kept alone).
```

This alphabet is set with the flag **-sim_matrix=vasiliky**. In order to keep the alphabet non degenerated, **-sim_matrix=idmat** can be used to retain the standard alphabet.

-matrix [CW]

Usage: -matrix=<blosum62mt>

Default: -matrix=blosum62mt

The usage of this flag has been modified from previous versions, due to frequent mistakes in its usage. This flag sets the matrix that will be used by alignment methods within t_coffee (slow_pair, lalign_id_pair). It does not affect external methods (like clustal_pair, clustal_aln…).

Users can also provide their own matrices, using the matrix format described in the appendix.

-nomatch

Usage: -nomatch=<positive value>

Default: -nomatch=0

Indicates the penalty to associate with a match. When using a library, all matches are positive or equal to 0. Matches equal to 0 are unsupported by the library but non-penalized. Setting nomatch to a non-negative value makes it possible to penalize these null matches and prevent unrelated sequences from being aligned (this can be useful when the alignments are meant to be used for structural modeling).

-gapopen

Usage: -gapopen=<negative value>

Default: -gapopen=0

Indicates the penalty applied for opening a gap. The penalty must be negative. If no value is provided when using a substitution matrix, a value will be automatically computed.

Here are some guidelines regarding the tuning of gapopen and gapext. In T-Coffee matches get a score between 0 (match) and 1000 (match perfectly consistent with the library). The default cosmetic penalty is set to -50 (5% of a perfect match). If you want to tune -gapoen and see a strong effect, you should therefore consider values between 0 and -1000.

-gapext

Usage: -gapext=<negative value>

Default: -gapext=0

Indicates the penalty applied for extending a gap (cf **-gapopen**)

-fgapopen

Unsupported

-fgapext

Unsupported

-cosmetic_penalty

Usage: -cosmetic_penalty=<negative value>

Default: -cosmetic_penalty=-50

Indicates the penalty applied for opening a gap. This penalty is set to a very low value. It will only have an influence on the portions of the alignment that are unalignable. It will not

make them more correct, but only more pleasing to the eye (i.e. Avoid stretches of lonely residues).

The cosmetic penalty is automatically turned off if a substitution matrix is used rather than a library.

-tg_mode

Usage: -tg_mode=<0, 1, or 2>

Default: -tg_mode=1

0: terminal gaps penalized with ***-gapopen*** + ***-gapext*len***

1: terminal gaps penalized with a ***-gapext*len***

2: terminal gaps unpenalized.

Weighting Schemes

-seq_weight

Usage: -seq_weight=<t_coffee or <file_name>>

Default: -seq_weight=t_coffee

These are the individual weights assigned to each sequence. The t_coffee weights try to compensate the bias in consistency caused by redundancy in the sequences.

sim(A,B)=%similarity between A and B, between 0 and 1.

weight(A)=1/sum(sim(A,X)^3)

Weights are normalized so that their sum equals the number of sequences. They are applied onto the primary library in the following manner:

res_score(Ax,By)=Min(weight(A), weight(B))*res_score(Ax, By)

These are very simple weights. Their main goal is to prevent a single sequence present in many copies to dominate the alignment.

Note: The library output by -out_lib is the un-weighted library.

Note: Weights can be output using the -outseqweight flag.

Note: You can use your own weights (see the format section).

Multiple Alignment Computation

-msa_mode

Usage: -msa_mode=<tree,graph,precomputed>

Default: -evaluate_mode=tree

Unsupported

-one2all

Usage: -one2all=<name>

Default: not used

Will generate a one to all library with respect to the specified sequence and will then align all the sequences in turn to that sequence, in a sequence determined by the order in which the sequences were provided.

–profile_comparison =profile, the MSAs provided via –profile are vectorized and the function specified by –profile_comparison is used to make profile profile alignments. In that case, the complexity is NL^2

-profile_comparison

Usage: -profile_mode=<fullN,profile>

Default: -profile_mode=full50

The profile mode flag controls the multiple profile alignments in T-Coffee. There are two instances where t_coffee can make multiple profile alignments:

1-When N, the number of sequences is higher than **–maxnseq,** the program switches to its multiple profile alignment mode (t_coffee_dpa).

2-When MSAs are provided via the **–profile** flag or via **–profile1** and **–profile2**.

In these situations, the –profile_mode value influences the alignment computation, these values are:

–profile_comparison =profile, the MSAs provided via –profile are vectorized and the function specified by –profile_comparison is used to make profile profile alignments. In that case, the complexity is NL^2

-profile_comparison=fullN, N is an integer value that can omitted. *Full* indicates that given two profiles, the alignment will be based on a library that includes every possible pair of sequences between the two profiles. If N is set, then the library will be restricted to the N most similar pairs of sequences between the two profiles, as judged from a measure made on a pairwise alignment of these two profiles.

-profile_mode

Usage: -profile_mode=<cw_profile_profile, muscle_profile_profile, multi_channel>

Default: -profile_mode=cw_profile_profile

When **–profile_comparison=profile**, this flag selects a profile scoring function.

Alignment Post-Processing

-clean_aln

Usage: -clean_aln

Default:-clean_aln

This flag causes T-Coffee to post-process the multiple alignment. Residues that have a reliability score smaller or equal to -clean_threshold (as given by an evaluation that uses -clean_evaluate_mode) are realigned to the rest of the alignment. Residues with a score higher than the threshold constitute a rigid framework that cannot be altered.

The cleaning algorithm is greedy. It starts from the top left segment of low constituency residues and works its way left to right, top to bottom along the alignment. You can require this operation to be carried out for several cycles using the -clean_iterations flag.

The rationale behind this operation is mostly cosmetic. In order to ensure a decent looking

alignment, the gop is set to -20 and the gep to -1. There is no penalty for terminal gaps, and the matrix is blosum62mt.

Note: Gaps are always considered to have a reliability score of 0.

Note: The use of the cleaning option can result in memory overflow when aligning large sequences,

-clean_threshold

Usage: -clean_threshold=<0-9>

Default:-clean_aln=1

See -clean_aln for details.

-clean_iteration

Usage: -clean_iteration=<value between 1 and >

Default:-clean_iteration=1

See -clean_aln for details.

-clean_evaluation_mode

Usage: -clean_iteration=<evaluation_mode >

Default:-clean_iteration=t_coffee_non_extended

Indicates the mode used for the evaluation that will indicate the segments that should be realigned. See -evaluation_mode for the list of accepted modes.

-iterate

Usage: -iterate=<integer>

Default: -iterate=0

Sequences are extracted in turn and realigned to the MSA. If iterate is set to -1, each sequence is realigned, otherwise the number of iterations is set by –iterate.

CPU Control

Multithreading

-multi_core

Usage: -multi_core= templates_jobs_relax_msa

Default: 0

template: fetch the templates in a parallel way

jobs: compute the library

relax: extend the library in a parallel way

msa: compute the msa in a parallel way

Specifies that the steps of T-Coffee that should be multi threaded. by default all relevant

steps are parallelized.

```
PROMPT: t_coffee sample_seq2.fasta -multi_core jobs
```

In order to prevent the use of the parallel mode it is possible to use:

```
PROMPT: t_coffee sample_seq2.fasta -multi_core no
```

-n_core

Usage: -n_core= <number of cores>

Default: 0

Default indicates that all cores will be used, as indicated by the environment via:

```
PROMPT: t_coffee sample_seq2.fasta -multi_core jobs
```

Limits

-mem_mode

Usage: deprecated

-ulimit

Usage: -ulimit=<value>

Default: -ulimit=0

Specifies the upper limit of memory usage (in Megabytes). Processes exceeding this limit will automatically exit. A value 0 indicates that no limit applies.

-maxlen

Usage: -maxlen=<value, 0=nolimit>

Default: -maxlen=1000

Indicates the maximum length of the sequences.

Aligning more than 100 sequences with DPA

-maxnseq

Usage: -maxnseq=<value, 0=nolimit>

Default: -maxnseq=50

Indicates the maximum number of sequences before triggering the use of t_coffee_dpa.

-dpa_master_aln

Usage: -dpa_master_aln=<File, method>

Default: -dpa_master_aln=NO

When using dpa, t_coffee needs a seed alignment that can be computed using any appropriate method. By default, t_coffee computes a fast approximate alignment.

A pre-alignment can be provided through this flag, as well as any program using the following syntax:

```
your_script –in <fasta_file> -out <file_name>
```

-dpa_maxnseq

Usage: -dpa_maxnseq=<integer value>

Default: -dpa_maxnseq=30

Maximum number of sequences aligned simultaneously when DPA is ran. Given the tree computed from the master alignment, a node is sent to computation if it controls more than –**dpa_maxnseq** OR if it controls a pair of sequences having less than **–dpa_min_score2** percent ID.

-dpa_min_score1

Usage: -dpa_min_score1=<integer value>

Default: -dpa_min_score1=95

Threshold for not realigning the sequences within the master alignment. Given this alignment and the associated tree, sequences below a node are not realigned if none of them has less than **–dpa_min_score1** % identity.

-dpa_min_score2

Usage: -dpa_min_score2

Default: -dpa_min_score2

Maximum number of sequences aligned simultaneously when DPA is ran. Given the tree computed from the master alignment, a node is sent to computation if it controls more than –**dpa_maxnseq** OR if it controls a pair of sequences having less than **–dpa_min_score2** percent ID.

-dap_tree [NOT IMPLEMENTED]

Usage: -dpa_tree=<filename>

Default: -unset

Guide tree used in DPA. This is a newick tree where the distance associated with each node is set to the minimum pairwise distance among all considered sequences.

Using Structures

Generic

-mode

Usage: -mode=3dcoffee

Default: turned off

Runs t_coffee with the 3dcoffee mode (cf next section).

-check_pdb_status

Usage: -check_pdb_status

Default: turned off

Forces t_coffee to run extract_from_pdb to check the pdb status of each sequence. This can considerably slow down the program.

3D Coffee: Using SAP

It is possible to use t_coffee to compute multiple structural alignments. To do so, ensure that you have the sap program installed.

```
PROMPT: t_coffee –pdb=struc1.pdb,struc2.pdb,struc3.pdb -method
sap_pair
```

Will combine the pairwise alignments produced by SAP. There are currently four methods that can be interfaced with t_coffee:

sap_pair: that uses the sap algorithm

align_pdb: uses a t_coffee implementation of sap, not as accurate.

tmaliagn_pair (http://zhang.bioinformatics.ku.edu/TM-align/)

mustang_pair (http://www.cs.mu.oz.au/~arun/mustang)

When providing a PDB file, the computation is only carried out on the first chain of this file. If your original file contains several chain, you should extract the chain you want to work on. You can use **t_coffee –other_pg extract_from_pdb** or any pdb handling program.

If you are working with public PDB files, you can use the PDB identifier and specify the chain by adding its index to the identifier (i.e. 1pdbC). If your structure is an NMR structure, you are advised to provide the program with one structure only.

If you wish to align only a portion of the structure, you should extract it yourself from the pdb file, using **t_coffee –other_pg extract_from_pdb** or any pdb handling program.

You can provide t_coffee with a mixture of sequences and structure. In this case, you should use the special mode:

```
PROMPT: t_coffee –mode 3dcoffee –seq 3d_sample3.fasta -
template_file template_file.template
```

Using/finding PDB templates for the Sequences

-template_file

Usage: -template_file =
<filename,
SCRIPT_scriptame,
SELF_TAG
SEQFILE_TAG_filename,
no>

Default: no

This flag instructs t_coffee on the templates that will be used when combining several types of information. For instance, when using structural information, this file will indicate the structural template that corresponds to your sequences. The identifier T indicates that the file should be a FASTA like file, formatted as follows. There are several ways to pass the templates:

Predefined Modes

EXPRESSO: will use the EBI server to find _P_ templates

PSIBLAST: will use the EBI sever to find profiles

File name

This file contains the sequence/template association it uses a FASTA-like format, as follows:

```
><sequence name> _P_ <pdb template>
><sequence name> _G_ <gene template>
><sequence name> _R_ <MSA template>
><sequence name> _F_ <RNA Secondary Structure>
><sequence name> _T_ <Transmembrane Secondary Structure>
><sequence name> _E_ <Protein Secondary Structure>
```

Each template will be used in place of the sequence with the appropriate method. For instance, structural templates will be aligned with sap_pair and the information thus generated will be transferred onto the alignment.

Note the following rule:

-Each sequence can have one template of each type (structural, genomics…)

-Each sequence can only have one template of a given type

-Several sequences can share the same template

-All the sequences do not need to have a template

The type of template on which a method works is declared with the SEQ_TYPE parameter in the method configuration file:

SEQ_TYPE S: a method that uses sequences

SEQ_TYPE PS: a pairwise method that aligns sequences and structures

SEQ_TYPE P: a method that aligns structures (sap for instance)

There are 4 tags identifying the template type:

_P_ Structural templates: a pdb identifier OR a pdb file

_G_ Genomic templates: a protein sequence where boundary amino-acid have been

recoded with (o:0, i:1, j:2)

_R_ Profile Templates: a file containing a multiple sequence alignment

_F_ RNA secondary Structures

More than one template file can be provided. There is no need to have one template for every sequence in the dataset.

_P_, _G_, and _R_ are known as **template TAGS**

2-SCRIPT_<scriptname>

Indicates that filename is a script that will be used to generate a valid template file. The script will run on a file containing all your sequences using the following syntax:

```
scriptname –infile=<your sequences> -

outfile=<template_file>
```

It is also possible to pass some parameters, use @ as a separator and # in place of the = sign. For instance, if you want to call the a script named blast.pl with the foloowing parameters;

```
blast.pl -db=pdb -dir=/local/test
```

Use

```
SCRIPT_blast.pl@db#pdb@dir#/local/test
```

Bear in mind that the input output flags will then be concatenated to this command line so that t_coffee ends up calling the program using the following system call:

```
blast.pl -db=pdb -dir=/local/test -infile=<some tmp file> -

outfile=<another tmp file>
```

3-SELF_TAG

TAG can take the value of any of the known TAGS (_S_, _G_, _P_). SELF indicates that the original name of the sequence will be used to fetch the template:

```
PROMPT: t_coffee 3d_sample2.fasta –template_file SELF_P_
```

The previous command will work because the sequences in 3d_sample3 are named

4-SEQFILE_TAG_filename

Use this flag if your templates are in filename, and are named according to the sequences. For instance, if your protein sequences have been recoded with Exon/Intron information, you should have the recoded sequences names according to the original:

```
SEQFILE_G_recodedprotein.fasta
```

-struc_to_use

Usage: -struc_to_use=<struc1, struc2…>

Default: -struc_to_use=NULL

Restricts the 3Dcoffee to a set of pre-defined structures.

Multiple Local Alignments

It is possible to compute multiple local alignments, using the moca routine. MOCA is a routine that allows extracting all the local alignments that show some similarity with another predefined fragment.

'mocca' is a perl script that calls t-coffee and provides it with the appropriate parameters.

-domain/-mocca

Usage: -domain

Default: not set

This flag indicates that t_coffee will run using the domain mode. All the sequences will be concatenated, and the resulting sequence will be compared to itself using lalign_rs_s_pair mode (lalign of the sequence against itself using keeping the lalign raw score). This step is the most computer intensive, and it is advisable to save the resulting file.

```
PROMPT: t_coffee -in Ssample_seq1.fasta,Mlalign_rs_s_pair -
out_lib=sample_lib1.mocca_lib -domain -start=100 -len=50
```

This instruction will use the fragment 100-150 on the concatenated sequences, as a template for the extracted repeats. The extraction will only be made once. The library will be placed in the file <lib name>.

If you want, you can test other coordinates for the repeat, such as

```
PROMPT: t_coffee -in sample_lib1.mocca_lib -domain -start=100 -
len=60
```

This run will use the fragment 100-160, and will be much faster because it does not need to re-compute the lalign library.

-start

Usage: -start=<int value>

Default: not set

This flag indicates the starting position of the portion of sequence that will be used as a template for the repeat extraction. The value assumes that all the sequences have been concatenated, and is given on the resulting sequence.

-len

Usage: -len=<int value>

Default: not set

This flag indicates the length of the portion of sequence that will be used as a template.

-scale

Usage: -scale=<int value>

Default: -scale=-100

This flag indicates the value of the threshold for extracting the repeats. The actual threshold is equal to:

motif_len*scale

Increase the scale ⇔Increase sensitivity ⇔ More alignments(i.e. -50).

-domain_interactive [Examples]

Usage: -domain_interactive

Default: unset

Launches an interactive mocca session.

```
PROMPT: t_coffee -in Lsample_lib3.tc_lib,Mlalign_rs_s_pair -domain
-start=100 -len=60
```

```
TOLB_ECOLI_212_26                        211      SKLAYVTFESGR--SALVIQTLANGAVRQV-
ASFPRHNGAPAFSPDGSKLAFA
TOLB_ECOLI_165_218    164 TRIAYVVQTNGGQFPYELRVSDYDGYNQFVVHRSPQPLMSPAWSPDGSKLAYV
TOLB_ECOLI_256_306    255 SKLAFALSKTGS--LNLYVMDLASGQIRQV-TDGRSNNTEPTWFPDSQNLAFT
TOLB_ECOLI_307_350    306 -------DQAGR--PQVYKVNINGGAPQRI-TWEGSQNQDADVSSDGKFMVMV
TOLB_ECOLI_351_393    350 -------SNGGQ--QHIAKQDLATGGV-QV-LSSTFLDETPSLAPNGTMVIYS
                          1           *             *    :          .   .:.  :

        MENU: Type Letter Flag[number] and Return: ex |10
        |x      -->Set     the START to x
        >x      -->Set     the LEN   to x
        Cx      -->Set     the sCale to x
        Sname   -->Save    the  Alignment
        Bx      -->Save    Goes back x it
        return  -->Compute the  Alignment
        X       -->eXit

[ITERATION   1] [START=211] [LEN= 50] [SCALE=-100]      YOUR CHOICE:
For instance, to set the length of the domain to 40, type:

[ITERATION   1] [START=211] [LEN= 50] [SCALE=-100]      YOUR CHOICE:>40[return]
[return]

Which will generate:

TOLB_ECOLI_212_252    211 SKLAYVTFESGRSALVIQTLANGAVRQVASFPRHNGAPAF  251
TOLB_ECOLI_256_296    255 SKLAFALSKTGSLNLYVMDLASGQIRQVTDGRSNNTEPTW  295
TOLB_ECOLI_300_340    299 QNLAFTSDQAGRPQVYKVNINGGAPQRITWEGSQNQDADV  339
TOLB_ECOLI_344_383    343 KFMVMVSSNGGQQHIAKQDLATGGV-QVLSSTFLDETPSL  382
TOLB_ECOLI_387_427    386 TMVIYSSSQGMGSVLNLVSTDGRFKARLPATDGQVKFPAW  426
                          1   :     :     :           ::         .     40




        MENU: Type Letter Flag[number] and Return: ex |10
```

```
        |x      -->Set     the START to x
        >x      -->Set     the LEN   to x
        Cx      -->Set     the sCale to x
        Sname   -->Save    the  Alignment
        Bx      -->Save    Goes back x it
        return  -->Compute the  Alignment
        X       -->eXit

[ITERATION   3] [START=211] [LEN= 40] [SCALE=-100]      YOUR CHOICE:
```

If you want to indicate the coordinates, relative to a specific sequence, type:

|<seq_name>:start

Type S<your name> to save the current alignment, and extract a new motif.

Type X when you are done.

Output Control

Generic

Conventions Regarding Filenames

stdout, stderr, stdin, no, /dev/null are valid filenames. They cause the corresponding file to be output in stderr or stdout, for an input file, stdin causes the program to requests the corresponding file through pipe. No causes a suppression of the output, as does /dev/null.

Identifying the Output files automatically

In the t_coffee output, each output appears in a line:

```
##### FILENAME <name> TYPE <Type> FORMAT <Format>
```

-no_warning

Usage: -no_warning

Default: Switched off

Suppresseswarning output.

Alignments

-outfile

Usage: -outfile=<out_aln file,default,no>

Default:-outfile=default

Indicates the name of the alignment output by t_coffee. If the default is used, the alignment is named *<your sequences>.aln*

-output

Usage: -output=<format1,format2,...>

Default:-output=clustalw

Indicates the format used for outputting the -outfile.

Supported formats are:

clustalw_aln, clustalw	: ClustalW format.
gcg, msf_aln	: MSF alignment.
pir_aln	: pir alignment.
fasta_aln	: fasta alignment.
phylip	: Phylip format.
pir_seq	: pir sequences (no gap).
fasta_seq	: fasta sequences (no gap).

As well as:

score_ascii	: causes the output of a reliability flag
score_html	: causes the output to be a reliability plot in HTML
score_pdf	: idem in PDF (if ps2pdf is installed on your system).
score_ps	: idem in postscript.

More than one format can be indicated:

```
PROMPT: t_coffee sample_seq1.fasta -output=clustalw,gcg,
score_html
```

A publication describing the CORE index is available on:
http://www.tcoffee.org/Publications/Pdf/core.pp.pdf

-outseqweight

Usage: -outseqweight=<filename>

Default: not used

Indicates the name of the file in which the sequences weights should be saved..

-case

Usage: -case=<keep,upper,lower>

Default: -case=keep

Instructs the program on the case to be used in the output file (Clustalw uses upper case). The default keeps the case and makes it possible to maintain a mixture of upper and lower case residues.

If you need to change the case of your file, you can use seq_reformat:

```
PROMPT: t_coffee -other_pg seq_reformat -in sample_aln1.aln -
action +lower -output clustalw
```

-cpu

Usage: deprecated

-outseqweight

Usage: -outseqweight=<name of the file containing the weights applied>

Default: -outseqweight=no

Will cause the program to output the weights associated with every sequence in the dataset.

-outorder [cw]

Usage: -outorder=<input OR aligned OR filename>

Default:-outorder=input

Sets the order of the sequences in the output alignment: **-outorder=input** means the sequences are kept in the original order. **-outorder=aligned** means the sequences come in the order indicated by the tree. This order can be seen as a one-dimensional projection of the tree distances. **–outdorder=<filename>**Filename is a legal fasta file, whose order will be used in the final alignment.

-inorder [cw]

Usage: -inorder=<input OR aligned>

Default:-inorder=aligned

Multiple alignments based on dynamic programming depend slightly on the order in which the incoming sequences are provided. To prevent this effect sequences are arbitrarily sorted at the beginning of the program (-inorder=aligned). However, this affects the sequence order within the library. You can switch this off by ststing –inorder=input.

-seqnos

Usage: -seqnos=<on or off>

Default:-seqnos=off

Causes the output alignment to contain residue numbers at the end of each line:

```
T-COFFEE
seq1 aaa---aaaa--------aa 9
seq2 a-----aa-----------a 4

seq1 a-----------------a 11
seq2 aaaaaaaaaaaaaaaaaa 19
```

Libraries

Although, it does not necessarily do so explicitly, T-Coffee always end up combining libraries. Libraries are collections of pairs of residues. Given a set of libraries, T-Coffee makes an attempt to assemble the alignment with the highest level of consistence. You can think of the alignment as a timetable. Each library pair would be a request from students or teachers, and the job of T-Coffee would be to

assemble the time table that makes as many people as possible happy…

-out_lib

Usage: -out_lib=<name of the library,default,no>

Default:-out_lib=default

Sets the name of the library output. Default implies <run_name>.tc_lib

-lib_only

Usage: -lib_only

Default: unset

Causes the program to stop once the library has been computed. Must be used in conjunction with the flag **–out_lib**

Trees

-newtree

Usage: -newtree=<tree file>

Default: No file specified

Indicates the name of the file into which the guide tree will be written. The default will be <sequence_name>.dnd, or <run_name.dnd>. The tree is written in the parenthesis format known as newick or New Hampshire and used by Phylips (see the format section).

Do NOT confuse this guide tree with a phylogenetic tree.

Reliability Estimation

CORE Computation

The CORE is an index that indicates the consistency between the library of piarwise alignments and the final multiple alignment. Our experiment indicate that the higher this consistency, the more reliable the alignment. A publication describing the CORE index can be found on:

http://www.tcoffee.org/Publications/Pdf/core.pp.pdf

-evaluate_mode

Usage: -evaluate_mode=<t_coffee_fast,t_coffee_slow,t_coffee_non_extended >

Default: -evaluate_mode=t_coffee_fast

This flag indicates the mode used to normalize the t_coffee score when computing the reliability score.

t_coffee_fast: Normalization is made using the highest score in the MSA. This evaluation mode was validated and in our hands, pairs of residues with a score of 5 or higher have 90 % chances to be correctly aligned to one another.

t_coffee_slow: Normalization is made using the library. This usually results in lower score

and a scoring scheme more sensitive to the number of sequences in the dataset. Note that this scoring scheme is not any more slower, thanks to the implementation of a faster heuristic algorithm.

t_coffee_non_extended: the score of each residue is the ratio between the sum of its non extended scores with the column and the sum of all its possible non extended scores.

These modes will be useful when generating colored version of the output, with the **–output** flag:

```
PROMPT: t_coffee sample_seq1.fasta -evaluate_mode t_coffee_slow -
output score_ascii, score_html

PROMPT: t_coffee sample_seq1.fasta -evaluate_mode t_coffee_fast -
output score_ascii, score_html

PROMPT: t_coffee sample_seq1.fasta -evaluate_mode
t_coffee_non_extended -output score_ascii, score_html
```

Generic Output

-run_name

Usage: -run_name=<your run name>

Default: no default set

This flag causes the prefix <your sequences> to be replaced by <your run name> when renaming the default output files.

-quiet

Usage: -quiet=<stderr,stdout,file name OR nothing>.

Default:-quiet=stderr

Redirects the standard output to either a file. **-quiet** on its own redirect the output to /dev/null.

-align [CW]

This flag indicates that the program must produce the alignment. It is here for compatibility with ClustalW.

APDB, iRMSD and tRMSD Parameters

Warning: These flags will only work within the APDB package that can be invoked via the –other_pg parameter of T-Coffee:

t_coffee –other_pg apdb –aln <your aln>

-quiet [Same as T-Coffee]

-run_name [Same as T-Coffee]

-aln

Usage: -aln=<file_name>.

Default:none

Indicates the name of the file containing the sequences that need to be evaluated. The sequences whose structure is meant to be used must be named according to their PDB identifier.

The format can be FASTA, CLUSTAL or any of the formats supported by T-Coffee. APDB only evaluates residues in capital and ignores those in lower case. If your sequences are in lower case, you can upper case them using seq_reformat:

```
PROMPT: t_coffee -other_pg seq_reformat -in 3d_sample4.aln -action
+upper -output clustalw > 3d_sample4.cw_aln
```

The alignment can then be evaluated using the defaultr of APDB:

```
PROMPT: t_coffee -other_pg apdb -aln 3d_sample4.aln
```

The alignment can contain as many structures as you wish.

-n_excluded_nb

Usage: -n_excluded_nb=<integer>.

Default:1

When evaluating the local score of a pair of aligned residues, the residues immediately next to that column should not contribute to the measure. By default the first to the left and first to the right are excluded.

-maximum_distance

Usage: -maximum_distance=<float>.

Default:10

Size of the neighborhood considered around every residue. If .-local_mode is set to sphere, -maximum_distance is the radius of a sphere centered around each residue. If –local_mode is set to window, then –maximum_distance is the size of the half window (i.e. window_size=-maximum_distance*2+1).

-similarity_threshold

Usage: -similarity_threshold=<integer>.

Default:70

Fraction of the neighborhood that must be supportive for a pair of residue to be considered correct in APDB. The neighborhood is a sphere defined by –maximum_distance, and the support is defined by –md_threshold.

-local_mode

Usage: -local_mode=<sphere,window>.

Default:sphere

Defines the shape of a neighborhood, either as a sphere or as a window.

-filter

Usage: -filter=<0.00-1.00>.

Default:1.00

Defines the centiles that should be kept when making the local measure. Foir instance, -filter=0.90 means that the the 10 last centiles will be removed from the evaluation. The filtration is carried out on the iRMSD values.

-print_rapdb [Unsupported]

Usage: -print_rapdb (FLAG)

Default:off

This causes the prints out of the exact neighborhood of every considered pair of residues.

-outfile [Same as T-Coffee]

This flag is meant to control the output name of the colored APDB output. This file will either display the local APDB score or the local iRMD, depending on the value of –color_mode. The default format is defined by –ouptut and is score_html.

-color_mode

Usage: -color_mode=<apdb, irmsd>

Default:apdb

This flag is meant to control the colored APDB output (local score). This file will either display the local APDB score or the local iRMD.

Building a Server

We maintain a T-Coffee server (www.tcoffee.org). We will be pleased to provide anyone who wants to set up a similar service with the sources

Environment Variables

T-Coffee stores a lots of information in locations that may be unsuitable when running a server.

By default, T-Coffee will generate and rely on the follwing directory structure:

```
/home/youraccount/            #HOME_4_TCOFFEE
HOME_4_TCOFFEE/.t_coffee/     #DIR_4_TCOFFEE
DIR_4_TCOFFEE/cache           #CACHE_4_TCOFFEE
DIR_4_TCOFFEE/tmp             #TMP_4_TCOFFEE
DIR_4_TCOFFEE/methods         #METHOS_4_TCOFFEE
DIR_4_TCOFFEE/mcoffee         #MCOFFEE_4_TCOFFEE
```

By default, all these directories are automatically created, following the dependencies suggested here.

The first step is the determination of the HOME. By default the program tries to use HOME_4_TCOFFEE, then the HOME variable and TMP or TEMP if HOME is not set on your system or your account. It is your responsibility to make sure that one of these variables is set to some valid location where the T-Coffee process is allowed to read and write.

If no valid location can be found for HOME_4_TCOFFEE, the program exits. If you are running T-Coffee on a server, we recommend to hard set the following locations, where your scratch is a valid location.

```
HOME_4_TCOFFEE="your scratch"
TMP_4_TCOFFEE="your scratch"
DIR_4_TCOFFEE="your scratch"
CACHE_4_TCOFFEE="your scratch"
NO_ERROR_REPORT_4_TCOFFEE=1
```

Note that it is a good idea to have a cron job that cleans up this scratch area, once in a while.

Output of the .dnd file.

A common source of error when running a server: T-Coffee MUST output the .dnd file because it re-reads it to carry out the progressive alignment. By default T-Coffee outputs this file in the directory where the process is running. If the T-Coffee process does not have permission to write in that directory, the computation will abort...

To avoid this, simply specify the name of the output tree:

-newtree=<writable file (usually in /tmp)>

Chose the name so that two processes may not over-write each other dnd file.

Permissions

The t_coffee process MUST be allowed to write in some scratch area, even when it is ran by Mr nobody... Make sure the /tmp/ partition is not protected.

Other Programs

T-Coffee may call various programs while it runs (lalign2list by defaults). Make sure your process knows where to find these executables.

Formats

Parameter files

Parameter files used with -parameters, -t_coffee_defaults, -dali_defaults... Must contain a valid parameter string where line breaks are allowed. These files cannot contain any comment, the recommended format is one parameter per line:

```
<parameter name>=<value1>,<value2>....
<parameter name>=.....
```

Sequence Name Handling

Sequence name handling is meant to be fully consistent with ClustalW (Version 1.75). This implies that in some cases the names of your sequences may be edited when coming out of the program. Five rules apply:

Naming Your Sequences the Right Way

1-No Space
Names that do contain spaces, for instance:
>seq1 human_myc
will be turned into
>seq1
It is your responsibility to make sure that the names you provide are not ambiguous after such an editing. This editing is consistent with Clustalw (Version 1.75)

2-No Strange Character
Some non alphabetical characters are replaced with underscores. These are: ';:()'
Other characters are legal and will be kept unchanged. This editing is meant to keep in line with Clustalw (Version 1.75).

3-> is NEVER legal (except as a header token in a FASTA file)

4-Name length must be below 100 characters, although 15 is recommended for compatibility with other programs.
5-Duplicated sequences will be renamed (i.e. sequences with the same name in the same dataset) are allowed but will be renamed according to their original order. When sequences come from multiple sources via the –in flag, consistency of the

renaming is not guaranteed. You should avoid duplicated sequences as they will cause your input to differ from your output thus making it difficult to track data.

Automatic Format Recognition

Most common formats are automatically recognized by t_coffee. See -in and the next section for more details. If your format is not recognized, use readseq or clustalw to switch to another format. We recommend Fasta.

Structures

PDB format is recognized by T-Coffee. T-Coffee uses extract_from_pdb (cf –other_pg flag). extract_from_pdb is a small embeded module that can be used on its own to extract information from pdb files.

RNA Structures

RNA structures can either be coded as T-Coffee libraries, with each line indicating two paired residues, or as alifold output. The selex format is also partly supported (see the seq_reformat tutorial on RNA sequences handling).

Sequences

Sequences can come in the following formats: fasta, pir, swiss-prot, clustal aln, msf aln and t_coffee aln. These formats are the one automatically recognized. Please replace the '*' sign sometimes used for stop codons with an X.

Alignments

Alignments can come in the following formats: msf, ClustalW, Fasta, Pir and t_coffee. The t_coffee format is very similar to the ClustalW format, but slightly more flexible. Any interleaved format with sequence name on each line will be correctly parsed:

```
<empy line>       [Facultative]n
<line of text>    [Required]
<line of text>                [Facultative]n
<empty line>                  [Required]
<empty line>                  [Facultative]n
<seq1 name><space><seq1>
<seq2 name><space><seq2>
<seq3 name><space><seq3>
<empty line>                  [Required]
<empty line>                  [Facultative]n
<seq1 name><space><seq1>
<seq2 name><space><seq2>
<seq3 name><space><seq3>
<empty line>                  [Required]
<empty line>                  [Facultative]n
```

An empty line is a line that does NOT contain amino-acid. A line that contains the ClustalW annotation (.:*) is empty.

Spaces are forbidden in the name. When the alignment is being read, non character signs are ignored in the sequence field (such as numbers, annotation…).

Note: a different number of lines in the different blocks will cause the program to crash

or hang.

Libraries

T-COFFEE_LIB_FORMAT_01

This is currently the only supported format.

```
!<space> TC_LIB_FORMAT_01
<nseq>
<seq1 name> <seq1 length> <seq1>
<seq2 name> <seq2 length> <seq2>
<seq3 name> <seq3 length> <seq3>
!Comment
(!Comment)n
#Si1 Si2
Ri1 Ri2 V1 (V2, V3)
#1 2
12 13 99 (12/0 vs 13/1, weight 99)
12 14 70
15 16 56
#1 3
12 13 99
12 14 70
15 16 56
!<space>SEQ_1_TO_N
```

Si1: index of Sequence 1

Ri1: index of residue 1 in seq1

V1: Integer Value: Weight

V2, V3: optional values

Note 1: There is a space between the ! And SEQ_1_TO_N

Note 2: The last line (! SEQ_1_TO_N) indicates that:

Sequences and residues are numbered from 1 to N, unless the token SEQ_1_TO_N is omitted, in which case the sequences are numbered from 0 to N-1, and residues are from 1 to N.

Residues do not need to be sorted, and neither do the sequences. The same pair can appear several times in the library. For instance, the following file would be legal:

```
#1 2
12 13 99
#1 2
15 16 99
#1 1
12 14 70
```

It is also poosible to declare ranges of resdues rather than single pairs. For instance, the following:

```
#0 1
+BLOCK+  10 12 14 99
+BLOCK+  15 30 40 99
#0 2
15 16 99
#0 1
12 14 70
```

The first statement BLOCK declares a BLOCK of length 10, that starts on position 12 of sequence 1 and position 14 of sequence 2 and where each pair of residues within the block has a score of 99. The second BLOCK starts on residue 30 of 1, residue 40 of 2 and extends for 15 residues.

Blocks can overalp and be incompatible with one another, just like single constraints.

T-COFFEE_LIB_FORMAT_02

A simpler format is being developed, however it is not yet fully supported and is only mentioned here for development purpose.

```
! TC_LIB_FORMAT_02
#S1 SEQ1 [OPTIONAL]
#S2 SEQ2 [OPTIONAL]
...
!comment [OPTIONAL]
S1 R1 Ri1 S2 R2 Ri2 V1 (V2 V3)
=> N R1 Ri1 S2 R2 Ri2 V1 (V2 V3)
...
```

S1, S2: name of sequence 1 and 2

SEQ1: sequence of S1

Ri1, Ri2: index of the residues in their respective sequence

R1, R2: Residue type

V1, V2, V3: integer Values (V2 and V3 are optional)

Value1, Value 2 and Value3 are optional.

Library List

These are lists of pairs of sequences that must be used to compute a library. The format is:

```
<nseq> <S1> <S2>
2 hamg2 globav
3 hamgw hemog singa
...
```

Substitution matrices.

If the required substitution matrix is not available, write your own in a file using the following format:

ClustalW Style [Deprecated]

```
# CLUSTALW_MATRIX FORMAT
$
v1
v2 v3
v4 v5 v6
...
$
```

v1, v2... are integers, possibly negatives.

The order of the amino acids is: ABCDEFGHIKLMNQRSTVWXYZ, which means that v1 is the substitution value for A vs A, v2 for A vs B, v3 for B vs B, v4 for A vs C and so on.

BLAST Format [Recommended]

```
# BLAST_MATRIX FORMAT
# ALPHABET=AGCT
 A G C T
A 0 1 2 3
G 0 2 3 4
C 1 1 2 3
...
```

The alphabet can be freely defined

Sequences Weights

Create your own weight file, using the -seq_weight flag:

```
# SINGLE_SEQ_WEIGHT_FORMAT_01
seq_name1 v1
seq_name2 v2
...
```

No duplicate allowed. Sequences not included in the set of sequences provided to t_coffee will be ignored. Order is free. V1 is a float. Un-weighted sequences will see their weight set to 1.

Known Problems

1-Sensitivity to sequence order: It is difficult to implement a MSA algorithm totally insensitive to the order of input of the sequences. In t_coffee, robustness is increased by sorting the sequences alphabetically before aligning them. Beware that this can result in confusing output where sequences with similar name are unexpectedly close to one another in the final alignment.

2-Nucleotides sequences with long stretches of Ns will cause problems to lalign, especially when using Mocca. To avoid any problem, filter out these nucleotides before running mocca.

3-Stop codons are sometimes coded with '*' in protein sequences. This will cause the program to crash or hang. Please replace the '*' signs with an X.

4-Results can differ from one architecture to another, due rounding differences. This is caused by the tree estimation procedcure. If you want to make sure an alignment is reproducible, you should keep the associated dendrogram.

5-Deploying the program on a

Technical Notes

These notes are only meant for internal development.

Development

The following examples are only meant for internal development, and are used to insure stability from release to release

PROFILE2LIST

prf1: profile containing one structure

prf2: profile containing one structure

```
PROMPT: t_coffee  Rsample_profile1.aln,Rsample_profile2.aln -
mode=3dcoffee -outfile=aligned_prf.aln
```

Command Line List

These command lines have been checked before every release (along with the other CL in this documentation:

-external methods;

```
   PROMPT: t_coffee sample_seq1.fasta -
in=Mclustalw_pair,Mclustalw_msa,Mslow_pair -outfile=clustal_text
```

-fugue_client

```
PROMPT: t_coffee -in Ssample_seq5.fasta Pstruc4.pdb Mfugue_pair
```

-A list of command lines kindly provided by James Watson (used to crash the pg before version 3.40)

```
PROMPT: t_coffee -in Sseq.fas P2PTC Mfugue_pair

PROMPT: t_coffee -in S2seqs.fas Mfugue_pair -template_file SELF_P_

PROMPT: t_coffee -mode 3dcoffee -in Sseq.fas P2PTC

PROMPT: t_coffee -mode 3dcoffee -in S2seqs.fas -template_file
SELF_P_
```

-A list of command lines that crashed the program before 3.81

```
PROMPT: t_coffee sample_seq6.fasta –in Mfast_pair Msap_pair
Mfugue_pair –template_file template_file6.template
```

-A command line to read "relaxed" pdb files...

```
PROMPT: t_coffee –in Msap_pair Ssample_seq7.fasta –template_file
template_file7.template –weight 1001 –out_lib test_lib7.tc_lib –
lib_only
```

-Parsing of MARNA libraries

```
PROMPT: t_coffee –in Lmarna.tc_lib –outfile maran.test
```

-Parsing of long sequence lines:

```
PROMPT: t_coffee –in Asample_aln5.aln –outfile test.aln
```

To Do...

-implement UPGMA tree computation

-implement seq2dpa_tree

-debug dpa

-Reconciliate sequences and template when reading the template

-Add the server command lines to the checking procedure