



Université Libre de Bruxelles
Service de Bioinformatique des Génomes et Reséaux (BiGRe)
Laboratory of Genome and Network Biology
<http://www.bigre.ulb.ac.be/>

Regulatory Sequence Analysis Tools (*RSAT*)

Installation guide

Jacques VAN HELDEN & the *RSAT*team

December 21, 2010

Contents

1	Description and requirements	4
1.1	Description	4
1.2	Requirements	4
1.2.1	Operating system	4
1.2.2	Perl language	4
1.2.3	Perl modules	5
1.2.4	Python language	5
1.2.5	Java language	6
1.2.6	Helper applications	6
2	Obtaining <i>RSAT</i> distribution	7
2.1	Downloading <i>RSAT</i> from the CVS server	7
2.1.1	CVS configuration	7
2.1.2	Obtaining a first version of <i>RSAT</i> programs	7
2.1.3	Updating <i>RSAT</i> programs	8
2.2	Installation from a compressed archive	8
3	Initializing <i>RSAT</i>	9
3.1	Configuring <i>RSAT</i>	9
3.1.1	Adding <i>RSAT</i> to your path	9
3.2	Initializing the directories	10
3.3	Adapting <i>RSAT</i> local configuration	10
3.3.1	Editing the <i>RSAT</i> local configuration file	11
3.3.2	Checking the <i>RSAT</i> path	11
3.3.3	Checking the java path (required for pathway tools only)	11
3.4	Compiling C programs in <i>RSAT</i>	11
3.5	Libraries required to support Ensembl genomes	12
4	Downloading genomes	14
4.1	Original data sources	14
4.2	Requirement : wget	14
4.3	Importing organisms from the <i>RSAT</i> main server	15
4.3.1	Obtaining the list of organisms supported on the <i>RSAT</i> server	15
4.3.2	Importing a single organism	15
4.3.3	Importing a few selected organisms	16
4.3.4	Importing all the organisms from a given taxon	16

5	Testing the command-line tools	17
5.1	Testing the access to the programs	17
5.1.1	Perl scripts	17
5.1.2	Testing Perl graphical librairies	17
5.1.3	Python scripts	17
5.1.4	C programs	18
5.2	Testing genome installation	18
6	Installing third-party programs	19
6.1	complementary programs for the analysis of regulatory sequences	19
6.2	Recommended programs for the Network Analysis Tools (NeAT)	20
7	Installing additional genomes on your machine	22
7.1	Installing genomes from NCBI/Genbank files	22
7.1.1	Organization of the genome files	22
7.1.2	Downloading genomes from NCBI/Genbank	25
7.1.3	Parsing a genome from NCBI/Genbank	26
7.1.4	Parsing a genome from the Broad institute (MIT)	26
7.1.5	Updating the configuration file	27
7.1.6	Checking the start and stop codon composition	27
7.1.7	Calibrating oligonucleotide and dyad frequencies with <i>install-organisms</i>	28
7.1.8	Installing a genome in your own account	28
7.2	Installing genomes from EMBL files	29

1 Description and requirements

1.1 Description

This document describes the installation procedure for the software suite **Regulatory Sequence Analysis Tools (RSAT)**, which offers several dozens of tools to detect cis-regulatory elements in DNA sequences [?, ?, ?].

1.2 Requirements

1.2.1 Operating system

RSAT is a unix-based package. It has been installed successfully on the following operating systems.

1. Linux
2. Mac OSX
3. Sun Solaris
4. Dec Alpha
5. cygwin (under MS Windows 98) (except for the graphical libraries, because I did not find a cygwin version of GD.pm)

RSAT is not compatible with any version of Microsoft Windows and I have no intention to make it compatible in a foreseeable future. Since most programs are written in perl, part of them might run under windows, but some others will certainly not, because they include calls to unix system commands.

1.2.2 Perl language

Most of the programs in *RSAT* are written in perl. Version 5.1 or later is recommended.

1.2.3 Perl modules

Some perl modules are required for the graphical tools of *RSAT*, and for some other specific programs. The perl modules can be found in the Comprehensive Perl Archive Network (<http://www.cpan.org/>), or can be installed with the command **cpan**.

GD.pm Interface to Gd Graphics Library. Used by *XYgraph* and *feature-map*.

PostScript::Simple Produce PostScript files from Perl. Used by *feature-map*.

Statistics::Distributions is used to calculate some probability distribution functions. In particular, it is used by the program *position-analysis* to calculate the P-value of the chi2.

Notes

- In previous releases, the chi2 P-value was computed using *Math::CDF*, but the precision was limited to 1e-15. *Statistics::Distributions* can compute P-values down to 1e-65.
- For the discrete functions (binomial , Poisson, hypergeometric) *RSAT* relies on a custom library (\$RSAT/perl-scripts/lib/RSAT/stats.pm) which reaches a precision of 1e-300.

File::Spec, POSIX and Data::Dumper are required for some functions of *matrix-scan*.

Util::Properties is required to load property files, which are used to specify the site-specific configuration of your *RSAT* server. Property files are also useful to write your own perl clients for the Web service interface to *RSAT*(RSATWS).

Class::Std::Fast and Storable are required to bring persistence to data structures like organisms. The slow procedure of organisms loading can then be done only once. This library can be easily installed via CPAN.

XML::LibXML is required for parsing and writing XML and uses the XML::Parser::Expat library. It is necessary for some RSAT applications.

DBD::mysql and DBI those two libraries are required by the program *retrieve-ensembl-seq* in order to access the ENSEMBL database.

SOAP::WSDL, Module::Build::Compat and Util::Properties are required for the Web services. *RSAT* Web services is a convenient interface that permits to write Perl scripts to run *RSAT* queries on a remote server.

1.2.4 Python language

Some of the programs in *RSAT* are written in python. Version 2.4 or later is recommended.

1.2.5 Java language

Some of the NeAT tools are written in Java. Java Runtime Environment 5 or higher is recommended.

1.2.6 Helper applications

The standard version of the program ***XYgraph*** export figures in bitmapformat (png, jpeg). If you want to support vectorial drawings (pdf), which give a much better resolution for printing, you need to install the freeware software ***gnuplot*** (4.2 or later), which can be downloaded from <http://www.gnuplot.info/>.

2 Obtaining *RSAT* distribution

For the time being, *RSAT* is distributed as a compressed archive. In a near future, we will also distribute it via an anonymous CVS server, which will greatly facilitate the updates.

Note The CVS distribution will soon be available for external users, but we still need to configure the CVS server to accept a guest login. For the time being, the CVS distribution is still restricted to the people from the lab. Inbetween, the only distribution mode for external users is the compressed archive. If you are not member of the BiGRe laboratory, please skip the section *Installation from the CVS repository*.

2.1 Downloading *RSAT* from the CVS server

Warning: we currently cannot manage user profiles in our CVS server, so the *RSAT* code is still distributed as a compressed archive. Please skip this section if you downloaded a compressed archive from the Web server (a file named *rsa-tools_2009XXXX.tar.gz*, where XXXX indicates the date).

2.1.1 CVS configuration

You need to indicate your *cv*s client to use *ssh* as remote shell application. For this, you can specify an environment variable.

if your shell is *tcsh*, add the following line to the *.cshrc* file in your home directory.

```
setenv CVS_RSH ssh
```

If your shell is *bash*, add the following line to the *.bashrc* file in your home directory.

```
export CVS_RSH=ssh
```

2.1.2 Obtaining a first version of *RSAT* programs

The following command should be used the first time you retrieve the tools from the server (you need to replace [mylogin] by the login name you received when signing the *RSAT* license).

```
cv
```

This will create a directory *rsa-tools* on your computer, and store the programs in it. Note that at this stage the programs are not yet functional, because you still need to install genomes, which are not included in the CVS distribution.

2.1.3 Updating *RSAT* programs

Once the tools have been retrieved, you can obtain updates very easily. For this, you need to change your directory to the `rsa-tools` directory, and use the `cvs` command in the following way.

```
cd rsa-tools
cvs update -d
```

2.2 Installation from a compressed archive

Uncompress the archive containing the programs. The archive is distributed `tar` format.

The `.tar.gz` file can be uncompressed with the command ***tar***, which are part of the default unix installation.

```
tar -xpf rsa-tools_YYYYMMDD.tar.gz
```


3 Initializing *RSAT*

3.1 Configuring *RSAT*

In order to use the command-line version of *RSAT*, you first need an account on a Unix machine where *RSAT* has been installed, and you should know the directory where the tools have been installed (if you don't know, ask assistance to your system administrator).

In the following instruction, we will assume that *RSAT* is installed in the directory `/home/rsat/rsa-tools`. This path has to be replaced by the actual path where *RSAT* has been installed on your computer.

3.1.1 Adding *RSAT* to your path

Before starting to use the tools, you need to define an environment variable (*RSAT*), and to add some directories to your path.

1. Create an environment variable named *RSAT* and containing the path of *rsa-tools*.

The way to create an environment variable depends on your shell. To know your shell, you can type

```
echo $SHELL
```

The answer should be something like

```
/sbin/bash
```

or

```
/bin/tcsh
```

.

Now, if we assume that *RSAT* has been installed in the following directory.

```
/home/rsat/rsa-tools
```

We will declare the *RSAT* path to your shell by defining an environment variable named *RSAT*. We will then add the path of the *RSAT* perl scripts, python scripts and binaries to your path. In addition, add java jar files to your classpath.

2. If your default shell is **tcsh** or **csh**, type the following commands (you probably need to update the first command to specify the *RSAT* path of your machine).

```
setenv RSAT /home/rsat/rsa-tools
set path=($path $RSAT/bin)
set path=($path $RSAT/perl-scripts)
set path=($path $RSAT/python-scripts)
set CLASSPATH=($RSAT/java/lib/NeAT_javatools.jar)
rehash
```

If your shell is bash, you should type the following command:

```
export RSAT=/home/rsat/rsa-tools
export PATH=${PATH}:${RSAT}/bin
export PATH=${PATH}:${RSAT}/perl-scripts
export PATH=${PATH}:${RSAT}/python-scripts
export CLASSPATH=${CLASSPATH}:${RSAT}/java/lib/NeAT_javatools.jar
```

(the `rehash` command updates the list of executable programs)

If you are using a different shell than bash, csh or tcsh, the specification of environment variables might differ from the syntax above. In case of doubt, ask your system administrator how to configure your environment variables and your path.

The specification of the environment variables and paths are required each time you want to use **RSAT**. You can add these specification to your personal profile. This file is normally found at the root of your personal account, in the file `.bashrc` if your shell is bash, or `.cshrc` if your shell is csh or tcsh. If you don't know how to proceed, ask your system administrator.

3.2 Initializing the directories

In addition to the programs, the installation of `rsa-tools` requires the creation of a few directories for storing data, access logs (for the web server), and temporary files.

The distribution includes a series of make scripts which will facilitate this step. You just need go to the `rsa-tools` directory, and start the appropriate make file.

```
cd $RSAT
make -f makefiles/init_RSAT.mk init
```

3.3 Adapting **RSAT** local configuration

The **RSAT** distribution comes with a template configuration file named `RSAT_config_default.props` and located in the `rsa-tools` directory.

Copy this file to create your own config file `RSAT_config.props`.

```
cp RSAT_config_default.props RSAT_config.props
```

3.3.1 Editing the *RSAT* local configuration file

You need to edit the file *RSAT_config.props* and specify the parameters of your local configuration. In particular, *it is crucial to specify the full path of the variable RSAT*, which specifies the *RSAT* main directory.

3.3.2 Checking the *RSAT* path

The *RSAT* programs should now be included in your path. To check if this is done properly, just type:

```
random-seq -l 350
```

If your configuration is correct, this command should return a random sequence of 350 nucleotides.

You are now able to use any program from the *RSAT* package, until you quit your session. It is however not very convenient to set the path manually each time you open a new connection. You can modify your default configuration by including the above commands in the file *.cshrc* (in *tsh*) or *.bashrc* (in *bash*) which should be found at the root of your home directory. If you don't know how to modify this file, see the system administrator.

3.3.3 Checking the java path (required for pathway tools only)

To check the correct setting for the java-based pathway tools, you can type the following command.

```
java graphtools.util.ListTools
```

This should display the list of available pathway analysis tools, which are part of the *NeAT* package.

3.4 Compiling C programs in *RSAT*

Some of the tools available in *RSAT* (*info-gibbs*, *matrix-scan-quick*, *count-words*) are written in the C. The distribution only contains the sources of these tools, because the binaries are operating system-dependent. The programs can be compiled in a very easy way.

```
cd ${RSAT}
make -f makefiles/init_RSAT.mk compile_all
```

This will compile and install the following programs in the directory *\$RSAT/bin*.

The installation directory can be changed by redefining the *BIN* variable. For instance, if you have the system administrator privileges, you could install the compiled programs in the standard directory for compiled packages (*/usr/local/bin*).

```
cd ${RSAT}
make -f makefiles/init_RSAT.mk compile_all BIN=/usr/local/bin
```

- **info-gibbs**: a gibbs sampling algorithm based on optimization of the the information content of the motif [?].
- **count-words**: an efficient algorithm for counting word occurrences in DNA sequences. This program is much faster than **oligo-analysis**, but it only returns the occurrences and frequencies, whereas oligo-analysis returns over-representation statistics and supports many additional options. **count-words** is routinely used to compute word frequencies in large genome sequences, for calibrating the Markov models used by **oligo-analysis**.
- **matrix-scan-quick**: an efficient algorithm for scanning sequences with a position-specific scoring matrix. As its name indicates, **matrix-scan-quick** is *much* faster than the Perl script **matrix-scan**, but presents reduced functionalities (only computes the weight, returns either a list of sites or the weight score distribution).

3.5 Libraries required to support EnSEMBL genomes

Since 2009, a series of **RSAT** programs support a direct access to the EnSEMBL database in order to ensure a convenient access to genomes from higher organisms [?].

- **supported-organisms-ensembl**
- **ensembl-org-info**
- **retrieve-ensembl-seq.pl**
- **get-ensembl-genome.pl**

Those programs require to install a few Perl libraries as well as a MySQL client on your machine.

To obtain BioPerl:

```
## For this example, we install Bioperl and EnSEMBL libraries
## in $RSAT/lib, but you can install it in some other place
mkdir -p $RSAT/lib
cd $RSAT/lib

## Login into the BioPerl CVS server
cvs -d :pserver:cvs@code.open-bio.org:/home/repository/bioperl login
### (password is 'cvs')

## download the current version
cvs -d :pserver:cvs@code.open-bio.org:/home/repository/bioperl \
  checkout bioperl-live
```

To obtain EnSEMBL:

```
## Login on the EnSEMBL CVS server
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/ensembl login
## (password is 'CVSUSER')
```

```
## Download the current version of Ensembl
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/ensembl \
    checkout -r branch-ensembl-56 ensembl
## (you need to adapt the branch number, e.g. 56, as it updates).

## Download the current version of the Ensembl module for comparative genomics
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/ensembl \
    checkout -r branch-ensembl-56 ensembl-compara
## (you need to adapt the branch number, e.g. 56, as it updates).
```

Adapt the 3 following lines in the RSAT configuration file *RSAT_config.props* to specify the actual path of the bioperl and ensembl libraries on your computer.

```
ensembl=/home/rsat/rSAT-tools/lib/ensembl/modules
compara=/home/rsat/rSAT-tools/lib/ensembl-compara/modules
bioperl=/home/rsat/rSAT-tools/lib/bioperl-live
```

You also need to define the URL of the Ensembl database in that configuration file:

```
## Ensembl host
## Used by the Ensembl-accessing tools (retrieve-ensembl-seq,
## get-ensembl-genome).
## URL of the server for the Ensembl DB. By default, the
## main ensembl server is called, but a local server can be specified.
ensembl_host=ensembl.db.ensembl.org
```

Finally, you need to include the BioPerl and Ensembl libraries in the Perl module path (specified by the environment variable named `$PERL5LIB`).

If your shell is bash, add the following lines in the file *.bashrc* at the root of your account.

```
export PERL5LIB=${PERL5LIB}:${RSAT}/lib/ensembl/modules
export PERL5LIB=${PERL5LIB}:${RSAT}/lib/ensembl-compara/modules
export PERL5LIB=${PERL5LIB}:${RSAT}/lib/ensembl-variation/modules
```

If your shell is tcsh or csh, add the following lines in the file *.cshrc* at the root of your account.

```
setenv PERL5LIB ${PERL5LIB}:${RSAT}/lib/ensembl/modules
setenv PERL5LIB ${PERL5LIB}:${RSAT}/lib/ensembl-compara/modules
setenv PERL5LIB ${PERL5LIB}:${RSAT}/lib/ensembl-variation/modules
```

The Ensembl libraries also require the SQL client Perl module *DBD::mysql*, as well as *DBI*, which can be installed with **cpan** (for this you need root privileges). Note that the installation of the CPAN module *DBD:mysql* requires a prior installation of a MySQL client on your machine (<http://dev.mysql.com/downloads/>).

To access Ensembl versions above 47, you need port 5306 to be opened on your machine. This might require to consult the system administrator of your network in order to ensure that the Firewall accepts this port.

Detailed information about the Ensembl libraries can be obtained on the Ensembl web site (¹).

¹http://www.ensembl.org/info/using/api/api_installation.html

4 Downloading genomes

RSAT includes a series of tools to install and maintain the latest version of genomes.

The most convenient way to add support for one or several organisms on your machine is to use the programs **supported-organisms** and **download-organism**.

Beware, the complete data required for a single genome may occupy several hundreds of Mb, because **RSAT** not only stores the genome sequence, but also the oligonucleotide frequency tables used to estimate background models, and the tables of BLAST hits used to get orthologs for comparative genomics. If you want to install many genomes on your computer, you should thus reserve a sufficient amount of space.

4.1 Original data sources

Genomes supported on **RSAT** were obtained from various sources.

Genomes can be installed either from the **RSAT** web site, or from their original sources.

- NCBI/Genbank (<ftp://ftp.ncbi.nih.gov/genomes/>)
- ENSEMBL (<http://www.ensembl.org/>)
- The EBI genome directory (<ftp://ftp.ebi.ac.uk/pub/databases/genomes/Eukaryot>)

Other genomes can also be found on the web site of a diversity of genome-sequencing centers.

4.2 Requirement : wget

The download of genomes relies on the application **wget**, which is part of linux distribution. **wget** is a “web aspirator”, which allows to download whole directories from ftp and http sites. You can check if the program is installed on your machine.

```
wget -help
```

This command should return the help pages for **wget**. If you obtain an error message (“command not found”), you need to ask your system administrator to install it.

4.3 Importing organisms from the *RSAT*main server

The simplest way to install organisms on our *RSAT*site is to download the *RSAT*-formatted files from the web server. For this, you can use a web aspirator (for example the program *wget*).

Beware, the full installation (including Mammals) requires a large disk space (several dozens of Gb). You should better start installing a small genome and test it before processing to the full installation. We illustrate the approach with the genome of our preferred model organism: the yeast *Saccharomyces cerevisiae*.

4.3.1 Obtaining the list of organisms supported on the *RSAT*server

By default, the program *supported-organisms* returns the list of organisms supported on your local *RSAT*installation. You can however use the option *-server* to obtain the list of organisms supported on a remote server.

```
supported-organisms -server
```

The command can be refined by restricting the list to a given taxon of interest.

```
supported-organisms -server -taxon Fungi
```

You can also ask additional information, for example the date of the last update and the source of each genome.

```
supported-organisms -server -taxon Fungi -return last_update,source,ID
```

4.3.2 Importing a single organism

The command

```
download-organism
```

allows you to download one or several organisms. Beware, the complete data for a single genome may occupy several tens of Megabytes (Bacterial genomes) or a few Gigabases (Mammalian). Downloading genomes thus requires a fast Internet connection, and may take time. If possible, please download genomes during the night (European time).

As a first step, we recommend to download the genome of the yeast *Saccharomyces cerevisiae*, since this is the model organism used in our tutorials.

```
download-organism -v 1 -org Saccharomyces_cerevisiae
```

In principle, the download should start immediately, and last for a few minutes.

After the task is completed, you can check if the configuration file has been correctly updated by typing the command.

```
supported-organisms
```

In principle, the following information should be displayed on your terminal.

Saccharomyces_cerevisiae

You can also add parameters to get specific information on the supported organisms.

```
supported-organisms -return ID,last_update
```

4.3.3 Importing a few selected organisms

The program **download-organism** can be launched with a list of organisms by using iteratively the option `-org`.

```
download-organism -v 1 -org Escherichia_coli_K12 -org Salmonella_typhi
```

4.3.4 Importing all the organisms from a given taxon

For comparative genomics, it is convenient to install on your server all the organisms of a given taxon. For this, you can simply use the option `-taxon` of **download-organism**.

Before doing this, it is wise to check the number of genomes that belong to this taxon on the server.

```
## Get the list of organisms belonging to the order "Enterobacteriales" on the server
supported-organisms -taxon Enterobacteriales -server

## Count the number of organisms
supported-organisms -taxon Enterobacteriales -server | wc -l
```

In Oct 2009, there are 94 Enterobacteriales supported on the **RSAT**server. Before starting the download, you should check two things:

1. Has your network a sufficient bandwidth to ensure the transfer in a reasonable time ?
2. Do you have enough free space on your hard drive to store all those genomes ?

If the answer to both questions is “yes”, you can start the download.

```
download-organism -v 1 -taxon Enterobacteriales
```


5 Testing the command-line tools

5.1 Testing the access to the programs

5.1.1 Perl scripts

From now on, you should be able to use the perl scripts from the command line. To test this, run:

```
random-seq -help
```

This should display the on-line help for the random sequence generator.

```
random-seq -l 200 -r 4 -a a:t 0.3 c:g 0.2
```

Should generate a random sequence of 200 nucleotides, with approximate proportions of 60% A/T and 40% G/C.

5.1.2 Testing Perl graphical librairies

RSAT includes some graphical tools (*feature-map* and *XYgraph*), which require a proper installation of Perl modules.

GD.pm Interface to Gd Graphics Library.

PostScript::Simple Produce PostScript files from Perl.

To test if these modules are available on your machine, type.

```
feature-map -help
```

If the modules are available, you should see the help message of the program feature-map. If not, you will see an error message complaining about the missing librairies. In such a case, ask your system administrator to install the missing modules.

5.1.3 Python scripts

From 2009 releases, the **RSAT** distribution includes some Python scripts. To test that they are running correctly, you can try.

```
random-motif -l 10 -c 0.85 -n 20
```

This command will generate a 20-columns position-specific scoring matrix (PSSM) with 85% conservation of one residue in each column.

5.1.4 C programs

You can test the correct installation of the C programs with the following command.

```
random-seq -l 1000 -a a:t 0.4 c:g 0.1 \  
| count-words -l 2 -v 1 -2str -i /dev/stdin
```

The first program (***random-seq***) is a Perl script, which generates a random sequence. The output is directly piped to the C program ***count-words***, which computes the frequencies and occurrences of each dinucleotide.

5.2 Testing genome installation

We will now test if the genomes are correctly installed. You can obtain the list of supported organisms with the command:

```
supported-organisms
```

If this command returns no result, it means that genomes were either not installed, or not correctly configured. In such a case, check the directories in the *data/genomes* directory, and check that the file *data/supported_organisms.pl*.

Once you can obtain the list of installed organisms, try to retrieve some upstream sequences. You can first read the list of options for the ***retrieve-seq*** program.

```
retrieve-seq -help
```

Select an organism (say *Saccharomyces cerevisiae*), and retrieve all the start codons with the following options :

```
retrieve-seq -org Saccharomyces_cerevisiae \  
-type upstream -from 0 -to +2 -all \  
-format wc -nocomment
```

This should return a set of 3 bp sequences, mostly ATG (in the case of *Saccharomyces cerevisiae* at least)

6 Installing third-party programs

6.1 complementary programs for the analysis of regulatory sequences

The *RSAT* distribution only contains the programs developed by the *RSAT* team.

A few additional programs, developed by third parties, can be integrated in the package. All third-party programs may be located in the directory *bin* directory of the *RSAT* distribution. In order to obtain these programs, please download them from their original site.

In particular, we recommend to install the following programs.

vmatch : developed by Stefan Kurtz, is used by the program ***purge-sequences***, for the detection of sequence repeats.

seqlogo : developed by Thomas D. Schneider, is used by the program ***convert-matrix*** to generate logos. It can be downloaded from <<http://weblogo.berkeley.edu/>>. ***seqlogo*** is the command-line version of ***WebLogo***.

Download the source code archive and uncompress it. Copy the following files to the directory *bin* of your *RSAT* distribution: ***seqlogo***, ***logo.pm***, ***template.pm*** and ***template.eps***.

seqlogo requires a recent version of ***gs*** (ghostscript¹) to create PNG and PDF output, and ***ImageMagic's convert***² to create GIFs.

patser : developed by Jerry Hertz, is used for matrix-based pattern matching.

matrix-based pattern discovery : several other pattern discovery programs have been embedded in the *RSAT* program ***multiple-family-analysis***: ***consensus*** (Jerry Hertz), ***meme*** (Tim Bailey), ***MotifSampler*** (Gert Thijs), ***gibbs*** (Andrew Neuwald).

We particularly recommend the installation of ***mkvtree*** and ***vmatch*** (Stefan Kurtz), because these programs are used by the program *purge-seq* to discard redundant sequence fragments.

In order to add functionalities to *RSAT*, install some or all of these programs and include their binaries path *rsa-tools/bin*. If you are not familiar with the installation of unix programs, ask assistance to your system administrator.

¹<http://www.ghostscript.com/>

²<http://www.imagemagick.org/>

Program	author	URL
vmatch	Stefan Kurtz	http://www.vmatch.de/
seqlogo	Thomas Sneider	http://weblogo.berkeley.edu/
patser	Jerry Hertz	ftp://ftp.genetics.wustl.edu/pub/stormo/Consensus/
consensus	Jerry Hertz	ftp://ftp.genetics.wustl.edu/pub/stormo/Consensus/
meme	Tim Bailey	http://meme.sdsc.edu/
MotifSampler	Gert Thijs	http://www.esat.kuleuven.ac.be/~thijs/download.html

Table 6.1: Programs from other developers which are complementary to the *RSAT* package.

6.2 Recommended programs for the Network Analysis Tools (NeAT)

For the Network Analysis Tools (NeAT), we recommend to install the following programs, which offer complementary functionalities for the analysis of networks/graphs.

Some programs come in the contrib directory of the *RSAT* distribution. Some others have to be downloaded from their original distribution site.

To compile the program **floydwarshall** located in the *contrib/floydwarshall* directory of *RSAT*, use this command :

```
gcc $RSAT/contrib/floydwarshall/floydwarshall.c -o $RSAT/bin/floydwarshall
```

In addition, the contributed programs REA and kWalks need to be installed.

To install kWalks, type:

```
cd $RSAT/contrib/kwalks;
tar xzvf kwalks.tgz;
cd src;
make clean; make
```

You can test taht the compilation worked by running the following command.

```
$RSAT/contrib/kwalks/bin/lkwalk
```

This should display the help message of **lkwalk**.

Check that the KWALKS_ROOT variable in the RSAT config file (*\$RSAT/RSAT_config.props*) points to the correct path (it should be the absolute path of *\$RSAT/contrib/kwalks/bin/*).

To install REA, type:

```
cd $RSAT/contrib/REA;
tar xzvf REA.tgz;
rm -f *.o;
make
```

You can test taht the compilation worked by running the following command.

```
$RSAT/contrib/REA/REA
```

This should display the help message of **REA**.

Check that the REA_ROOT variable in the RSAT config file (*\$RSAT/RSAT_config.props*) points to the correct path (it should be the absolute path of *\$RSAT/contrib/REA*).

7 Installing additional genomes on your machine

The easiest way to install genomes on your machine is to download them from the main **RSAT**server, as indicated in the Chapter “Downloading genomes” (Chap. 4 of the installation guide).

In some cases, you may however wish to install a genome by yourself, because this genome is not supported on the main **RSAT**server. For this, you can use the programs that we use to install new genomes on the main **RSAT**server.

7.1 Installing genomes from NCBI/Genbank files

In the section 4, we saw that the genomes installed on the main **RSAT**server can easily be installed on your local site. In some cases, you would like to install additional genomes, which are not published yet, or which are not supported on the main **RSAT**server.

If your genomes are available in Genbank (files .gbk) or EMBL (files .embl) format, this can be done without too much effort, using the installation tools of **RSAT**.

The parsing of genomes from their original data sources is however more tricky than the synchronization from the **RSAT**server, so this procedure should be used only if you need to install a genome that is not yet supported.

If this is not your case, you can skip the rest of this section.

7.1.1 Organization of the genome files

In order for a genome to be supported, **RSAT**needs to find at least the following files.

1. organism description
2. genome sequences
3. feature tables (CDS, mRNA, ...)
4. lists of names/synonyms

From these files, a set of additional installation steps will be done by **RSAT**programs in order to compute the frequencies of oligonucleotides and dyads in upstream sequences.

If you installed **RSAT**as specified above, you can have a look at the organization of a supported genome, for example the yeast *Saccharomyces cerevisiae*.

```
cd ${RSAT}/public_html/data/genomes/Saccharomyces_cerevisiae/genome
ls -l
```

As you see, the folder *genome* contains the sequence files and the tables describing the organism and its features (CDSs, mRNAs, ...). The **RSAT** parser exports tables for all the feature types found in the original genbank file. There are thus a lot of distinct files, but you should not worry too much, for the two following reasons:

1. **RSAT** only requires a subset of these files (basically, those describing organisms, CDSs, mRNAs, rRNAs and tRNAs).
2. All these files can be generated automatically by **RSAT** parsers.

Organism description

The description of the organism is given in two separate files.

```
cd ${RSAT}/public_html/data/genomes/Saccharomyces_cerevisiae/genome
ls -l organism*.tab
```

```
more organism.tab
```

```
more organism_names.tab
```

1. *organism.tab* specifies the ID of the organism and its taxonomy. The ID of an organism is the TAXID defined by the NCBI taxonomical database, and its taxonomy is usually parsed from the .gbk files (but you may need to specify it yourself in case it would be missing in your own data files).
2. *organism_name.tab* indicates the name of the organism.

Genome sequence

A genome sequence is composed of one or more contigs. A contig is a contiguous sequence, resulting from the assembly of short sequence fragments obtained during the sequencing. When a genome is completely sequenced and assembled, each chromosome comes as a single contig.

In **RSAT**, the genome sequence is specified as one separate file per contig (chromosome) sequence. Each sequence file must be in raw format (i.e. a text file containing the sequence without any space or carriage return).

In addition, the genome directory contains one file indicating the list of the contig (chromosome) files.

```
cd $RSAT/data/genomes/Saccharomyces_cerevisiae/genome/
```

```
## The list of sequence files
cat contigs.txt
```

```
## The sequence files
ls -l *.raw
```

Feature table

The *genome* directory also contains a set of feature tables giving the basic information about gene locations. Several feature types (CDS, mRNA, tRNA, rRNA) can be specified in separate files (*cds.tab*, *mrna.tab*, *trna.tab*, *rrna.tab*).

Each feature table is a tab-delimited text file, with one row per feature (cds, mrna, ...) and one column per parameter. The following information is expected to be found.

1. Identifier
2. Feature type (e.g. ORF, tRNA, ...)
3. Name
4. Chromosome. This must correspond to one of the sequence identifiers from the fasta file.
5. Left limit
6. Right limit
7. Strand (D for direct, R for reverse complement)
8. Description. A one-sentence description of the gene function.

```
## The feature table
head -30 cds.tab
```

Feature names/synonyms

Some genes can have several names (synonyms), which are specified in separate tables.

1. ID. This must be one identifier found in the feature table
2. Synonym
3. Name priority (primary or alternate)

```
## View the first row of the file specifying gene names/synonyms
head -30 cds_names.tab
```

Multiple synonyms can be given for a gene, by adding several lines with the same ID in the first column.

```
## An example of yeast genes with multiple names
grep YFL021W cds_names.tab
```


7.1.2 Downloading genomes from NCBI/Genbank

The normal way to install an organism in *RSAT* is to download the complete genome files from the NCBI ¹, and to parse it with the program *parse-genbank.pl*.

However, rather than downloading genomes directly from the NCBI site, we will obtain them from a mirror ² which presents two advantages?

- Genome files are compressed (gzipped), which strongly reduces the transfer and storage volume.
- This mirror can be queried by *rsync*, which facilitates the updates (with the appropriate options, *rsync* will only download the files which are newer on the server than on your computer).

RSAT includes a makefile to download genomes from different sources. We provide hereafter a protocol to create a download directory in your account, and download genomes in this directory. Beware, genomes require a lot of disk space, especially for those of higher organisms. To avoid filling up your hard drive, we illustrate the protocol with the smallest procaryote genome to date: *Mycoplasma genitalium*.

```
## Creating a directory for downloading genomes in your home account
cd $RSAT
mkdir -p downloads
cd downloads

## Creating a link to the makefile which allows you to download genomes
ln -s $RSAT/makefiles/downloads.mk ./makefile
```

We will now download a small genome from NCBI/Genbank.

```
## Downloading one directory from NCBI Genbank
cd $RSAT/downloads/
make one_genbank_dir NCBI_DIR=Bacteria/Mycoplasma_genitalium
```

We can now check the list of files that have been downloaded.

```
## Downloading one directory from NCBI Genbank
cd $RSAT/downloads/
ls -l ftp.ncbi.nih.gov/genomes/Bacteria/Mycoplasma_genitalium/
```

RSAT parsers only use the files with extension *.gbk.gz*.

You can also adapt the commande to download (for example) all the Fungal genomes in a single run.

```
## Downloading one directory from NCBI Genbank
cd $RSAT/downloads/
make one_ncbi_dir NCBI_DIR=Fungi
```

You can do the same for Bacteria, or for the whole NCBI genome repository, but this requires several Gb of free disk space.

¹<ftp://ftp.ncbi.nih.gov/genomes/>

²bio-mirror.net/biomirror/ncbigenomes/

7.1.3 Parsing a genome from NCBI/Genbank

The program ***parse-genbank.pl*** extract genome information (sequence, gene location, ...) from Genbank flat files, and exports the result in a set of tab-delimited files.

```
parse-genbank.pl -v 1 \  
-i $RSAT/downloads/ftp.ncbi.nih.gov/genomes/Bacteria/Mycoplasma_genitalium
```

7.1.4 Parsing a genome from the Broad institute (MIT)

The website <http://www.broad.mit.edu/> offers a large collection of genomes that are not available on the NCBI website. We wrote a specific parser for the Broad files.

To this, download the following files for the organism of interest : the supercontig file, the protein sequences and the annotation file in the GTF format.

These files contain sometimes too much information that should be removed.

This is an example of the beginning of the fasta file containing the protein traduction. In this file, we should remove everything that follows the protein name.

```
>LELG_00001 | Lodderomyces elongisporus hypothetical protein (translation) (1085 aa)  
MKYDTAAQLSLINPQTLKGLPIKPFPLSQPVFVQGVNNDTKAITQGVFLDVTVHFISLPA  
ILYLHEQIPVGQVLLGLPFQDAHKLSIGFTDDGDKRELRFANGNIHKFPIRYDGDSNYH  
IDSFPTVQVSQTVVIPPLSEMLRPAFTGSRASEDDIRYFVDQCAEVSDVFYIKGGDPGRL
```

This is an example of the beginning of the fasta file containing the contigs. In this file, we should remove everything that follows the name of the contig.

```
>supercontig_1.1 of Lodderomyces elongisporus  
AAGAGCATCGGGCAAATGATGTTTTTCAGTCCATCAATGTGATGGATCTGATAGTTGAAG  
GTCCTGATGAAGTTCAACCATTTGTAAACCCGATTTACAAAGTGTGAATTATCGAGTGGT  
TTATTCATCACAAAGGACAAGAGCTTTGTTGGTTGACAGAGATGTTTTGCAGAAAGCCCTT  
AAGGATGGTATTGCCTTGTTCAGAAGAAACCAGTTGTTACTGAAGTAAATCTGACGACC
```

This is an example of the beginning of the GTF file containing the contigs annotation. We should rename the contig name so that it corresponds to the fasta file of contig. To this, we will remove the text in the name of the contig (only keep the supercontig number) and add a prefix.

```
supercont1.1%20of%20Lodderomyces%20elongisporus LE1_FINAL_GENECALL start_codon  
322 324 . + 0 gene_id "LELG_00001"; transcript_id "LELT_00001";  
supercont1.1%20of%20Lodderomyces%20elongisporus LE1_FINAL_GENECALL stop_codon  
3574 3576 . + 0 gene_id "LELG_00001"; transcript_id "LELT_00001";  
supercont1.1%20of%20Lodderomyces%20elongisporus LE1_FINAL_GENECALL exon 322  
3576 . + . gene_id "LELG_00001"; transcript_id "LELT_00001";  
supercont1.1%20of%20Lodderomyces%20elongisporus LE1_FINAL_GENECALL CDS 322 3573  
. + 0 gene_id "LELG_00001"; transcript_id "LELT_00001";
```

We use the parse ***parse-broad-mit***.

```
parse-broad-mit.pl -taxid 36914 -org Lodderomyces_elongisporus \  
-nuc_seq lodderomyces_elongisporus_1_supercontigs.fasta \  
-gtf lodderomyces_elongisporus_1_transcripts.gtf \  
-gtf_remove 'supercont' \  
-gtf_remove '%20of%20Lodderomyces%20elongisporus' \  

```

```
-contig_prefix LELG_ -nuc_remove supercontig_ \
-nuc_remove ' of Lodderomyces elongisporus' \
-aa lodderomyces_elongisporus_1_proteins.fasta -aa_remove ' .*'
```

This will create the raw files, the feature files and the protein sequence file.

7.1.5 Updating the configuration file

After having parsed the genome, you need to perform one additional operation in order for **RSAT** to be aware of the new organism: update the configuration file.

```
install-organism -v 1 -org Mycoplasma_genitalium -task config
```

```
## Check the last lines of the configuration file
tail -15 $RSAT/data/supported_organisms.pl
```

From now on, the genome is considered as supported on your local **RSAT** site. You can check this with the command **supported-organisms**.

7.1.6 Checking the start and stop codon composition

Once the organism is found in your configuration, you need to check whether sequences are retrieved properly. A good test for this is to retrieve all the start codons, and check whether they are made of the expected codons (mainly ATG, plus some alternative start codons like GTG or TTG for bacteria).

The script **install-organism** allows you to perform some additional steps for checking the conformity of the newly installed genome. For example, we will compute the frequencies of all the start and stop codons, in order to check that gene locations were correctly parsed.

```
install-organism -v 1 -org Mycoplasma_genitalium -task start_stop
```

```
ls -l $RSAT/data/genomes/Mycoplasma_genitalium/genome/*start*
```

```
ls -l $RSAT/data/genomes/Mycoplasma_genitalium/genome/*stop*
```

The stop codons should be TAA, TAG or TGA, for any organism. For eucaryotes, all start codons should be ATG. For some procaryotes, alternative start codons (GTG, TGG) are found with some genome-specific frequency.

```
cd $RSAT/data/genomes/Mycoplasma_genitalium/genome/
```

```
## A file containing all the start codons
more Mycoplasma_genitalium_start_codons.wc
```

```
## A file with trinucleotide frequencies in all start codons
more Mycoplasma_genitalium_start_codon_frequencies
```

```
## A file containing all the stop codons
more Mycoplasma_genitalium_stop_codons.wc
```

```
## A file with trinucleotide frequencies in all stop codons
more Mycoplasma_genitalium_stop_codon_frequencies
```

7.1.7 Calibrating oligonucleotide and dyad frequencies with *install-organisms*

The programs *oligo-analysis* and *dyad-analysis* require calibrated frequencies for the background models. These frequencies are calculated automatically with *install-organism*.

```
install-organism -v 1 -org Debaryomyces_hansenii \  
-task allup,oligos,dyads,upstream_freq,protein_freq
```

Warning: this task may require several hours of computation, depending on the genome size. For the *RSAT* server, we use a PC cluster to regularly install and update genomes. As the task *allup*, is a prerequisite for the computation of all oligonucleotide and dyad frequencies, it should be run directly on the main server before computing oligo and dyad frequencies on the nodes of the cluster. We will thus proceed in two steps. Note that this requires a PC cluster and a proper configuration of the batch management program.

```
## Retrieve all upstream sequences  
## Executed directly on the server  
install-organism -v 1 -org Debaryomyces_hansenii \  
-task allup  
  
## Launch a batch queue for computing all oligo and dyad frequencies  
## Executed on the nodes of a cluster  
install-organism -v 1 -org Debaryomyces_hansenii \  
-task oligos,dyads,upstream_freq,protein_freq -batch
```

7.1.8 Installing a genome in your own account

In some cases, you might want to install a genome in your own account rather than in the *RSAT* folder, in order to be able to analyze this genome before putting it in public access.

In this chapter, we explain how to add support for an organism on your local configuration of *RSAT*. This assumes that you have the complete sequence of a genome, and a table describing the predicted location of genes.

First, prepare a directory where you will store the data for your organism. For example:

```
mkdir -p $HOME/rsat-add/data/Mygenus_myspecies/
```

Once you have this information, start the program *install-organism*. You will be asked to enter the information needed for genome installation.

Updating your local configuration

- Modify the local config file
- You need to define an environment variable called `RSA_LOCAL_CONFIG`, containing the full path of the local config file.

7.2 Installing genomes from EMBL files

RSAT also includes a script ***parse-embl.pl*** to parse genomes from EMBL files. However, for practical reasons we generally parse genomes from the NCBI genome repository. Thus, unless you have a specific reason to parse EMBL files, you can skip this section.

The program ***parse-embl.pl*** reads flat files in EMBL format, and exports genome sequences and features (CDS, tRNA, ...) in different files.

As an example, we can parse a yeast genome sequenced by the “Genolevures” project ³.

Let us assume that you want to parse the genome of the species *Debaryomyces hansenii*.

Before parsing, you need to download the files in your account,

- Create a directory for storing the EMBL files. The last level of the directory should be the name of the organism, where spaces are replaced by underscores. Let us assume that you store them in the directory *\$RSAT/downloads/Debaryomyces_hansenii*.
- Download all the EMBL file for the selected organism. Save each name under its original name (the contig ID), followed by the extension *.embl*)

We will check the content of this directory.

```
ls -l $RSAT/downloads/Debaryomyces_hansenii
```

On my computer, it gives the following result

```
CR382133.embl
CR382134.embl
CR382135.embl
CR382136.embl
CR382137.embl
CR382138.embl
CR382139.embl
```

The following instruction will parse this genome.

```
parse-embl.pl -v 1 -i $RSAT/downloads/Debaryomyces_hansenii
```

If you do not specify the output directory, a directory is automatically created by combining the current date and the organism name. The verbose messages will indicate you the path of this directory, something like *\$HOME/parsed_data/embl/20050309/Debaryomyces_hansenii*.

You can now perform all the steps above (updating the config file, installing oligo- and dyad frequencies, ...) as for genomes parsed from NCBI.

³<http://natchaug.labri.u-bordeaux.fr/Genolevures/download.php>

Installing a genome in the main *RSAT* directory

Once the genome has been parsed, the simplest way to make it available for all the users is to install it in the *RSAT* genome directory. You can already check the genomes installed in this directory.

```
ls -l $RSAT/data/genomes/
```

There is one subdirectory per organism. For example, the yeast data is in *\$RSAT/data/genomes/Saccharomyces_cerevisiae/*. This directory is further subdivided in folders: *genome* and *oligo-frequencies*.

We will now create a directory to store data about *Debaryomyces_hansenii*, and transfer the newly parsed genome in this directory.

```
## Create the directory
mkdir -p $RSAT/data/genomes/Debaryomyces_hansenii/genome

## Transfer the data in this directory
mv $HOME/parsed_data/embl/20050309/Debaryomyces_hansenii/* \
  $RSAT/data/genomes/Debaryomyces_hansenii/genome

## Check the transfer
ls -ltr $RSAT/data/genomes/Debaryomyces_hansenii/genome
```