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

March 24, 2016

Contents

1 Introduction

This documents describes procedure to install a stand-alone version of the **Regulatory Sequence Analysis Tools** (***RSAT***) on your computer [**?**, **?**, **?**, **?**].

2 Quick installation guide

The full installation of all the ***RSAT*** functionalities requires some external libraries and companion programs. Some installation steps are tricky, because they depend on the operating system. However, the majority of the tools does not require a full installation.

In order to give you a quick start, this section proposes a simplified installation procedure.

The detailed installation instructions willl be presented in the next chapters. In case of trouble, don't hesitate to contact Jacques.van-Helden@univ-amu.fr.

2.1 Requirements

Make sure that the following programs are installed on your machine.

- ***tar***
- ***git***
- ***cvs***
- ***wget***
- ***perl*** version 5.14 or later

2.2 Basic installation steps

1. **Download** the compressed tar archive from the ***RSAT*** distribution site, and uncompress it.

```
tar -xpzf rsat_yyyymmdd.tar.gz
```

2. Open a terminal, set your working directory to the *rsat* folder extracted from this archive, and run the ***RSAT*** **configuration script**.

```
cd rsat;
perl perl-scripts/configure_rsat.pl
```

Tip: at this stage, the way to specify some parameters might seem obscure to you (e.g. URL of your web site, cluster parameters, NeAT, pathway tools, paths for helper programs). Actually these parameters are only required for the full installation of a web- and cluster-enabled ***RSAT*** instance.

If you don't know how to specify a parameter, just leave it to its default value. You will have the possibility to re-run the configuration script later.

3. Define RSAT environment variable and adapt your path

```
source RSAT_config.bashrc
```

Check the RSAT environment variable

```
echo $RSAT
```

4. Include RSAT parameters in your default bash config.

For future sessions, include the following line in your bash configuration file (*$HOME/.bashrc* or, even better, */etc/bashrc* if you have admin rights on the machine).

```
source [RSAT_FULL_PATH]/RSAT_config.bashrc
```

Note: for Ubuntu distribution , the simplest solution is to create a soft link from the directory */etc/profile.d/*. This requires admin rights.

```
sudo ln -s $RSAT/RSAT_config.bashrc /etc/profile.d/rsat.sh
```

5. **Initialize *RSAT* folders**

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

6. Install the required Perl libraries. For this you need admin rights.

```

## Set working directory to RSAT
cd $RSAT

## Get the list of Perl modules to be installed
make -f makefiles/install_rsat.mk  perl_modules_list

## Check which Perl modules are already installed
make -f makefiles/install_rsat.mk perl_modules_check
## The locations of installed modules are stored in perl_modules_check.txt
more check_perl_modules_eval.txt

## Note: don't worry if tthe module Object::InsideOut is labelled as
## Failed, for some reason the check fails but the module does work.


############################################################## IN
## CASE SOME MODULES ARE MISSING, they can be installed with the
## following commands, but this requires admin rights, because only
## the system administrator has the right to install Perl modules with
```

```
## cpan.

cd $RSAT
sudo bash ## Become admin of your machine

## Load the RSAT configuration
source RSAT_config.bashrc

## Check that RSAT path has been defined
echo $RSAT

## Install missing modules (this requires admin rights)
make -f makefiles/install_rsat.mk perl_modules_install


## Important: exit from the admin status, because the next steps have to be done a
exit
```

- **Beware:** this step requires admin rights (you will be prompted for sudo password in order to install modules in cpan).
- ***cpan*** installation takes some time, and is interrupted by many prompts to let you choose some installation options. We recommend to use the default option for all questions (just press the "Enter" key). If you are bored to confirm each installation step, you can run the following target, which will automatically blindly accept all the default options.

```
    make -f makefiles/install_rsat.mk perl_modules_install_noprompt
```

- Even though some Perl modules may fail to install, don't worry too much. At this stage, you should be able to use most of ***RSAT*** functionalities.

7. compile ***RSAT*** programs written in C

```
make -f makefiles/init_rsat.mk compile_all
```

8. Install some third-party programs required by some ***RSAT*** scripts.

```
make -f makefiles/install_software.mk install_ext_apps
```

9. Install two model organisms, required for some of the Web tools.

```
download-organism -v 1 -org Saccharomyces_cerevisiae \
   -org Escherichia_coli_K_12_substr__MG1655_uid57779
```

Optionally, you can download additional organisms with the same command. The list of supported-organisms can be obtained with the command ***supported-organisms-server***.

2.3 Testing the basic installation

At this stage, you should now dispose of a local installation of ***RSAT*** with all the basic functionalities enabled. We can now test the proper functioning of the different types of programs.

2.3.1 Testing the path

RSAT envionment variable

```
echo $RSAT
```

RSAT exec dirs in the PATH

Check that the folders containing RSAT executables are included in your path.

```
echo $PATH | perl -pe 's|:|\n|g' | grep rsat
```

The result should contain the full path to the folders *bin*, *perl-scripts*, and *python-scripts*.

Testing a *RSAT* Perl script

Test a simple Perl script that does not require organisms to be installed.

```
random-seq -l 100 -n 2
```

Testing a *RSAT* python script

Test a simple python script that does not require organisms to be installed.

```
random-motif -l 10 -c 0.90
```

Testing a compiled C program

```
random-seq -l 1000 -n 100  | count-words -v 1 -1str
```

Testing external programs

vmatch

The program ***vmatch*** is an precious companion to ***RSAT*** discovery tools (***oligo-analysis***, ***dyad-analysis***, ***position-analysis***, ***local-word-analysis***).

It requires a freeware license (http://www.vmatch.de/).

After having obtained the license, you will receive a file named vmatch.lic, which should be copied in the directory $RSAT/bin.

```
## Quick test with a random sequence
random-seq -l 100 -n 1  | purge-sequence

## Realistic application: retrieve two overlapping promoters ->
## contain redundant sequences (on opposite strands). After purging,
## the second sequence should contain 'n' characters over the last
## ~600 base pairs.
retrieve-seq -org Saccharomyces_cerevisiae -q GAL1 -q GAL10  | purge-sequence
```

If you get an error message, see section 3.1.1 of this manual.

seqllogo

The program ***seqlogo*** is used to draw logos from position-specific scoring matrices. It is required for several ***RSAT***tools (***convert-matrix***, ***peak-motifs***, ***matrix-clustering***, ***footprint-discovery***, . . .).

```
## Locate the path of seqlogo
which seqlogo

## get the help for seqlogo
seqlogo

## ghostscript
which gs
gs --version
```

webllogo

Note: in 2015 we replaced the use of ***seqlogo*** by ***weblogo***. We keep both for backwards compatibility.

The program ***weblogo*** (version 3) is used to draw logos from position-specific scoring matrices. It is required for several ***RSAT***tools (***convert-matrix***, ***peak-motifs***, ***matrix-clustering***, ***footprint-discovery***, . . .).

```
## Locate the path of weblogo
which weblogo

## get the help for weblogo
weblogo --help
```

Testing supported organisms

Get the list of organisms supported on your computer.

```
supported-organisms
```

Get the list of organisms supported on the server http://rsat-tagc.univ-mrs.fr/rsat/. This script requires some Perl libraries for the SOAP/WSDL protocol. If it works, it means that you are ready to use ***RSAT***Web services.

```
supported-organisms-server -v 2 -server http://rsat-tagc.univ-mrs.fr/rsat/ \
   -o supported_on_rsat-tagc.tab ;
```

```
## Count the number of supported organisms on the remote server
wc -l  supported_on_rsat-tagc.tab

## Check the list of supported organisms on the remote server
more   supported_on_rsat-tagc.tab
```

Get the list of organisms supported on another server http://pedagogix-rsat.univ-mrs.fr/rs

```
supported-organisms-server -server http://pedagogix-tagc.univ-mrs.fr/rsat/ \
   -o supported_on_pedagogix-tagc.tab ;
wc -l  supported_on_pedagogix-tagc.tab
```

3 Full installation

The full installation of ***RSAT***sowftare suite includes some additional steps.

You should read this chapter only if you want to enable some of the following functionalities:

1. programs complementary to ***RSAT***, developed by other teams
2. local Web server
3. Web services
4. distributed computing on a cluster (or on multiple processors of a single computer)
5. metabolic pawthay analysis tools

3.1 Installing third-party programs

3.1.1 vmatch and mkvtree

The programs ***vmatch*** and ***mkvtree*** are required by the ***RSAT***progam ***purge-sequence***, which plays an important role to discard redundant sequences before running motif discovery algorithms (***oligo-analysis***, ***dyad-analysis***, ***position-analysis***, ***local-word-analysis***).

A free academic license can be obtained at Stefan Kurt's web page:
http://www.vmatch.de/

After having obtained the licence, install the 3 following files in the *$RSAT/bin* folder: *vmatch*, *mkvtree*, *vmatch.lic*.

Quick test for the correct functioning of ***purge-sequence***:

```
retrieve-seq -org Saccharomyces_cerevisiae -q GAL1 -q GAL10 -noorf \
  | purge-sequence
```

The second sequence (GAL10) should be masked (replaced by "n"), because GAL10 and GAL1 shar the same promoter (the genes are transcribed in opposite direction).

3.1.2 Other third-pary programs

Some additional freeware programs are required for some particular tasks in ***RSAT***. The list of these programs can be obtained as follows.

```
make -f makefiles/install_software.mk list_ext_apps
```

Calling the makefile with the target *install_ext_apps* will start the automatic installation of all these programs.

```
make -f makefiles/install_software.mk install_ext_apps
```

Note: for some programs, you may be prompted for the sudo password, depending on the configuration you entered in the previous step (with the script *configure_rsat.pl*).

In case of trouble, try to install the programs one by one by calling separately each target listed by *list_ext_apps*.

3.2 Configuring and activating a local *RSAT* Web server

In order to provide web access to the Regulatory Sequence Analysis Tools (***RSAT***), you need to adapt the configuration of your web server. This requires root privileges (can be done only by the system administrator of the computer).

1. A default configuration file is provided with the ***RSAT*** distribution (*rsat_apache_default.conf*).

   Copy this template to a file named *rsat.conf*, which you will edit to replace the string `[RSAT_PARENT_PATH]` by the full path of your *rsat* folder.

2. The configuration file should then be copied to some appropriate place in the Apache configuration folder of your computer. This place depends on the operating system (Mac OSX or Linux) and on the distribution (Linux Ubuntu, Centos, ...).

   Some Usual places:

   - On Centos: */etc/httpd/conf.d/rsat.conf*
   - On Ubuntu: */etc/apache2/sites-enabled/rsat.conf*
   - On Mac OSX: */etc/apache2/users/rsat.conf*

3. You need to restart the Web server (note: the command depends on your OS. Can be ***apachectl***, ***apache2ctl*** or ***httpd***.

   ```
   sudo apachectl restart
   ```

4. Check that all properties related to the Web site URL are properly defined in the ***RSAT*** property files *$RSAT/RSAT_config.props* and *$RSAT/RSAT_config.mk*.

   In principle you already configured these files in the beginning of the installation, with the command

   ```
   perl perl-scripts/configure_rsat.pl
   ```

   Note: it is important to properly define the URL fo the Web server (`RSAT_WWW` and related variables). The default URL (http://localhost/rsat/) only works if the server and client (your Web browser) are on the same machine. This internal access is very convenient to work in places where you don't have Internet connections, but does not allow other computers to use your Web server. If you want to enable Web queries from remote computers, you should specify an externally visible URL.

3.2.1 Activating web services on your *RSAT* instance

By default, Web services requests are redirected towards the main ***RSAT*** server. To configure your ***RSAT*** instance as a Web services provider, you first need to update the WSDL file, which provides all technical information about supported web services.

```
cd $RSAT ;
make -f makefiles/init_rsat.mk ws_init
```

This will also display the parameters of your local web services, which depend on the variable RSAT_WWW when you ran the ***RSAT*** configuration script (*perl-scripts/configure_rsat.pl*). Check that these parameters are correct.

After this, you should generate the web services stub, with the following command.

```
make -f makefiles/init_rsat.mk ws_stub
```

You can test if the web services are working.

```
make -f makefiles/init_rsat.mk ws_stub_test
```

4 Description and requirements

4.1 Description

The Regulatory Sequence Analysis Tools (***RSAT***) is a software suite specialized for the detection of cis-regulatory elements in genomic sequences. It also contains a series of complementary tools for genome management, statistics, and other related analyses.

The ***RSAT*** package comes along with the Network Analysis Tools (NeAT), a software suite combining a variety of tools for the analysis of biomolecular networks (interactome, regulatory networks, metabolic pathways).

4.2 Requirements

4.2.1 Operating system

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

1. Linux
2. Mac OSX (latest version tested: 10.8.3)
3. Sun Solaris
4. Dec Alpha

RSAT is not compatible with any version of Microsoft Windows and we have no intention to make it compatible in a foreseeable future.

4.2.2 Perl language

Most of the programs in ***RSAT*** are written in Perl. Version 5.1 or later is recommended. A set of Perl modules is required, the ***RSAT*** package includes a script to install them automatically (see Chapter 7).

4.2.3 Python language

Some of the programs in ***RSAT*** are written in Python.

Python release 2.7[1] is recommended, because it contains some required libraries for remote access to external resources (UCSC genome browser).

The following Python libraries are required for various programs.

- **setuptools**[2] is required to install other Python libraries (see installation instructions[3]).
- **suds**[4] is used for accessing the SOAP interface.

4.2.4 Helper applications

wget

The program ***wget***, is used to download

1. some helper programs developed by third-parties, which can be installed in ***RSAT***;
2. genomes from the ***RSAT*** server to your local ***RSAT*** installation.

wget is part of linux distribution. If it is not installed on your computer, you can download it from `http://www.gnu.org/software/wget/`. An installation package for Mac OSX can be found at `http://download.cnet.com/Wget/3000-18506_4-128268.html`.

gnuplot

The standard version of the ***RSAT*** 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/`.

git (only for developers)

For co-developers of the ***RSAT*** suite, the code is distributed program ***git***. For external users, there is no need to use ***git*** since the code is distributed as a compressed archive on a Web page.

[1] `http://www.python.org/getit/releases/2.7/`
[2] `http://pypi.python.org/pypi/setuptools`
[3] `http://pypi.python.org/pypi/setuptools#installation-instructions`
[4] `https://fedorahosted.org/suds/`

5 Obtaining *RSAT* distribution

For the time being, ***RSAT*** is distributed as a compressed archive.

The license can be obtained from the ***RSAT*** web site (`http://rsat.eu/`).

5.1 Installation from a compressed archive

Download the latest version of the ***RSAT*** distribution. Uncompress the archive containing the programs. The archive is distributed `tar` format. The `.tar.gz` file can be uncompressed with the command ***tar***, which is included in most Unix distributions.

```
tar -xpzf rsat_yyyymmdd.tar.gz
```

6 Initializing *RSAT*

6.0.1 Configuring your *RSAT* server

RSAT requires to specify a set of parameters, which will be stored in three property files:

1. *RSAT_config.props*: this file describes site-specific parameters. The same property file is loaded by the Perl, python and java programs, thereby ensuring the consistency of site-specific configuration.

2. *RSAT_config.mk*: a subset of parameters from the props file, which are required to run the makefiles (in particular during RSAT installation).

3. *RSAT_config.bashrc*: definiton of environment variables, paths and Perl library paths required to run the RSAT/NeAT tools.

4. *RSAT_config.conf*: configuration of the web site (for Apache servers).

In particular, *it is crucial to specify the full path of the variable RSAT*, which specifies the RSAT main directory.

The simplest way to update the configuration file is to run the following script.

```
## Enter in RSAT distribution folder
cd rsat

## Run the configuration script
perl perl-scripts/configure_rsat.pl
```

Alternatively, you can edit the files with a text editor of your choice.

6.0.2 Loading RSAT environment variables

The configuration script has created a bash file *RSAT_config.bashrc* in the ***RSAT***distribution directory (folder *rsat*).

This file should be loaded each time you enter a session. There are several alternative ways to do this.

1. Source the file manually at each new session

   ```
   ## Load the RSAT environment variables
   source RSAT_config.bashrc
   ```

2. Copy the content of this file in the bash configuration file in your home directory (*/.bashrc* or */.bash_profile*). ***RSAT*** environment variables will then be loaded for you at each connexion.

3. If you dispose of admin rights, you can copy the content of this file in the main bash configuration file:
   - in Lunix (Centos or Ubuntu), you can add a soft link (
     ```
     ln -s
     ```
) to *[RSAT_PARENT_PATH]/RSAT_config.bashrc* in the bas completion directory: */etc/bash_completion.d/*;
   - Mac OSX, add a line to the (*/etc/bashrc*) to source the file [RSAT_PARENT_PATH]/RSAT_config.

   RSAT environment variables will then be automatically loaded for each user of this computer.

6.1 Initializing the directories

In addition to the programs, the installation of rsat 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 rsat directory, and start the appropriate make file.

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

6.1.1 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.

Don't worry if you see a warning looking like this:

```
; WARNING The tabular file with the list of supported organism cannot be read
; WARNING Missing file [RSAT_PARENT_PATH]/rsat/public_html/data/supported_organisms.tab
```

This warning will disappear as soon as you download the first organism in ***RSAT***.

7 Installing 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***.

7.1 Before installing Perl modules: install the GD library

The Perl module ***GD.pm*** requires prior installation of the ***GD*** library.

- On *Linux* systems, this library can be installed with the package manager of the distribution. for example:
  - ***apt-get*** for Ubuntu
  - ***aptitude*** for Ubuntu (better treatment of dependencies than apt-get)
  - ***yum*** for Centos
  - ***yast*** for Suze
  - ...
- On *Mac OSX* systems, the installation of the GD library can be done with the program ***brew*** (http://brew.sh/).

  After having installed brew, you can install the GD library in your system by typing.

```
brew install gd
```

7.2 Automatic installation of Perl modules

The simplest way to install all the required Perl modules is to tye the command below. *Beware:* this command sudo requires administrator rights on the computer. If you don't have the root password, please consult your system administrator.

```
## Acquire the system administrator rights
sudo bash;

## Define the RSAT environment variable.
##
## You must replace [RSAT_PATH] by the full path to your rsat folder.
```

```
export RSAT=[RSAT_PATH]

## Check that the RSAT environment variable has been defined
echo $RSAT

## Check that the RSAT environment variable points towards the right directory
ls -1 $RSAT
## This should give you the list of the files and folders included in your rsat folder.

## Set your working directory to the rsat folder
cd $RSAT

## Display the list of Perl modules that will be installed
make -f makefiles/install_rsat.mk perl_modules_list;

## Print the command that will be used to install the Perl modules (just for checking)
make -f makefiles/install_rsat.mk perl_modules_cmd;

## Install the Perl modules
make -f makefiles/install_rsat.mk perl_modules_install;
```

Beware, ***cpan*** will frequently ask you to confirm the installation steps. you should thus check the CPAN process and answer "yes" at each prompt.

The following command enables to install all the Perl modules in a somewhat risky, but less cumbersome way. It relies on the command ***yes*** to automatically answer each question by a carriage return, which will lead ***cpan*** to chose the default option.

```
## Install the Perl modules
make -f makefiles/install_rsat.mk perl_modules_install;
```

In case some modules would not be properly installed with the above commands, you can try installing them manually (the list of required modules is listed in the next section).

7.3 Additional Perl modules required to support EnsEMBL genomes

This section is required only if you intend to use the ***RSAT*** programs interfaced to the Ensembl database. Since 2008, 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.

The first requirement is the ***BioPerl*** module, which has in principle been installed in Chapter 7). The MySQL client should also have been installed in Chapter 7).

To obtain EnsEMBL [1].

```
## Make sure you start from the right directory
cd $RSAT

## Display the parameters for installing Ensembl API (in particular,
## the version for Ensembl and EnsemblGenomes).
##
## Check the number of the latest release on the respective web sites.
##   Ensembl: http://www.ensembl.org/index.html
##   EnsemblGenomes: http://ensemblgenomes.org/
make -f makefiles/install_software.mk install_ensembl_api_param

## Install the ensembl library
make -f makefiles/install_software.mk install_ensembl_api

## Notes: you need to enter the following passwords for the CVS
## servers.
## - For Ensembl:  CVSUSER.
## - For bioperl:  cvs
```

Note that there may be incompatibilities between successive versions of the Ensembl API. The install script includes a parameter `ENSEMBL_VERSION` to specify the version ("branch") of the Ensembl API distribution. Moreover, there are different release numbers of the "historical" Ensembl database, and for the EnsemblGenomes databases (Bacteria, Fungi, Plants, Metazoa).

In addition, there are dependencies between releases. So, EnsemblGenomes 20 is compatible with Ensembl verison 73, whereas Ensembl has already released its version 74. In order to install an API compatible with Ensembl and EnsemblGenomes, we recommend always check the latest releases of both databases on the EnsemblGenomes web page (`http://www.ensemblgenomes`) and adapt the following command accordingly.

```
## Install the ensembl library with a specific branch number.
make -f makefiles/install_software.mk install_ensembl_api \
   ENSEMBL_VERSION=73
```

The installation script will print out a series of modifications of the PERL5LIB variable, that should be added to your bashrc file in order to provide support for Ensembl Perl API.

You should also check the specification of ensembl paths in the props file.

```
ensembl=[RSAT_PARENT_PATH]/rsat/lib/ensembl/modules
compara=[RSAT_PARENT_PATH]/rsat/lib/ensembl-compara/modules
variation=[RSAT_PARENT_PATH]/rsat/lib/ensembl-variation/modules
```

[1] Full instructions at `http://useast.ensembl.org/info/docs/api/api_cvs.html`

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=ensembldb.ensembl.org
```

Notes:

1. to access EnsEMBL versions above 47, you need port 5306 to be opened on your machine. This might require an intervention of your 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 ([2]).

[2] http://www.ensembl.org/info/using/api/api_installation.html

8 Compiling C programs in *RSAT*

Some of the tools available in ***RSAT***(***info-gibbs***, ***matrix-scan-quick***, ***count-words***) are written in the ***C*** language. 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*.

9 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.

9.1 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/`) was the primary source for installing genomes on ***RSAT***. Genomes are downloaded from the ftp site and installed locally on the ***RSAT*** server by parsing the .gbk files.
- The EBI genome directory (`ftp://ftp.ebi.ac.uk/pub/databases/genomes/Eukaryot`) contains supplementary genomes, which can be downloaded and installed on the ***RSAT*** server by parsing files in embl format.
- UCSC (`http://genome.ucsc.edu/`) for the multi-genome alingment files (multiz) used by ***peak-footprints***.
- Since 2008, ENSEMBL (`http://www.ensembl.org/`) genomes are supported by special tools (***retrieve-ensembl-seq***, ***supported-organisms-ensembl***), that remotely address queries to the Ensembl database.
- Since 2013, genomes can be downloaded and installed on ***RSAT*** servers, using the tool ***install-ensembl-genome***. Once installed, ensembl genomes can be queried with the same tools as the other genomes installed on ***RSAT*** servers (***retrieve-seq***, ***gene-info***, ...).

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

9.2 Requirement : wget

The download of genomes relies on the application ***wget***, which is part of linux distribution[1].

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.

9.3 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 installting 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*.

9.3.1 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
```

9.3.2 Importing a single organism

The command

```
download-organism
```

[1] For Linux: http://www.gnu.org/software/wget/; for Mac OSX http://download.cnet.com/Wget/3000-18506_4-128268.html

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 tenomes 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. *Beware*, the data volume to be downloaded is important, because the genome comes together with extra files (blast hits with other genoems, oligonucleotide and dyad frequencies). Depending on the network bandwidth, the download of a genome may take several minutes or tens of 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
```

9.3.3 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_K_12_substr__MG1655_uid57779 \
   -org Mycoplasma_genitalium_G37_uid57707
```

Note: genome names may change with time, since genome centers are occasionally adding new suffixes for genomes. The organism names indicated after the option *-org* should belong to the list of supported organisms collected with the command ***supported-organisms -server***.

9.3.4 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 Dec 2013, there are 203 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
```

10 Testing the command-line tools

10.1 Testing the access to the programs

10.1.1 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 -n 4
```

Should generate a random sequence of 200 nucleotides.
You can optionnally specify different frequencies for A,C,G and T residues.

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

10.1.2 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.

10.1.3 Python scripts

The ***RSAT*** distribution includes some Python scripts. To test if they are running correctly, you can try the proram ***random-motif***.

```
random-motif  -l 10 -c 0.85 -n 3
```

This command will generate 3 position-specific scoring matrix (PSSM) of 10-columns with 85% conservation of one residue in each column.

10.1.4 C programs

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

```
random-seq -l 10000 -n 10 | 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.

10.2 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 -feattype CDS \
        -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). We can combine ***retrieve-seq*** and ***oligo-analysis*** to check the frequencies of trinucleotides found at the start positions of all yeast genes.

```
retrieve-seq -org Saccharomyces_cerevisiae -feattype CDS \
  -type upstream -from 0 -to +2 -all \
  | oligo-analysis -l 3 -1str -return occ,freq -v 1 -sort
```

11 Installing third-party programs

11.1 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 optionally be integrated in the package. All third-party programs may be loacated in the directory *bin* directory of the ***RSAT*** distribution.

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

Some of those can be downloaded and installed automatically using the makefile *install_rsat.mk*. Before doing this, you must make sure that the program ***wget*** (this program is supported on Linux [1] and Mac OSX [2] systems).

You can then run the following commands to install some of the third-party programs that are complementary to ***RSAT***.

```
cd $RSAT;
make -f makefiles/install_software.mk install_ext_apps
```

Some other third-party programs will require a manual installation (in particular, ***vmatch*** and ***mkvtree***).

vmatch **and** ***mkvtree*** : developed by Stefan Kurtz, are used by the program ***purge-sequences***, to mask redundant sequences that bias motif discovery statistics.

seqlogo : developed by Thomas D. Schneider, is used used by the programs ***convert-matrix***, ***compare-matrices***, ***peak-motifs***, ***matrix-quality*** and a few others, to generate logos. ***seqlogo*** is the command-line version of ***WebLogo***[3].

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*** (ghostscript[4]) to create PNG and PDF output, and ***ImageMagic's convert***[5] to create GIFs.

[1] http://www.gnu.org/software/wget/
[2] http://download.cnet.com/Wget/3000-18506_4-128268.html
[3] http://weblogo.berkeley.edu/
[4] http://www.ghostscript.com/
[5] http://www.imagemagick.org/

Program	author	URL
vmatch+mkvtree	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 11.1: Programs from other developers which are complementary to the ***RSAT*** package.

matrix-based pattern discovery : several third-party pattern discovery programs can be optionally called from some ***RSAT*** task managers (e.g. ***multiple-family-analysis***, ***peak-motifs***).

- ***meme*** (Tim Bailey)
- ***consensus*** (Jerry Hertz)
- ***MotifSampler*** (Gert Thijs)
- ***gibbs*** (Andrew Neuwald)

Their installation is not properly required for ***RSAT*** functioning, but it may be convenient to install them in order to compare the results retured by alternative motif discovery approaches on the same data sets.

12 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. 9 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.

12.1 Adding support for Ensembl genomes

In addition to the genomes imported and maintained on your local ***RSAT*** server, the program ***retrieve-ensembl-seq*** allows you to retrieve sequences for any organism supported in the Ensembl database (`http://ensembl.org`).

For this, you first need to install the Bioperl and Ensembl Perl libraries (see section 7.3).

12.1.1 Handling genomes from Ensembl

The first step to work with Ensembl genomes is to check the list of organisms currently supported on their Web server.

```
supported-organisms-ensembl
```

You can then get more precise information about a given organism (build, chromosomes) with the command ***ensembl-org-info***.

```
ensembl-org-info -org Drosophila_melanogaster
```

Sequences can be retrieved from Ensembl with the command ***retrieve-ensembl-seq***.

You can for example retrieve the 2kb sequence upstream of the transcription start site of the gene *PAX6* of the mouse.

```
retrieve-ensembl-seq.pl -org Mus_musculus -q PAX6 \
  -type upstream -feattype mrna -from -2000 -to -1 -nogene -rm \
  -alltranscripts -uniqseqs
```

Options

- *-type upstream* specifies that we want to collect the sequences located upstream of the gene (more procisely, upstream of the mRNA).

- *-feattype mrna* indicates that the reference for computing coordinates is the mRNA. Since we collect upstream sequences, the 5'most position of the mRNA has coordinate 0, and upstream sequences have negative coordinates. Note that many genes are annotated with multiple RNAs for different reasons (alternative splicing, alternative transcription start sites). By default, the program will return the sequences upstream of each mRNA annotated for the query gene.

- *-nogene* clip the sequences to avoid overlapping the next upstream gene.

- *-rm* repeat masking (important for pattern discovery). Repetitive sequences are replaced by `N` characters.

12.2 Installing genomes and variations from *EnsEMBL*

??

RSAT includes a series of programs to download and install genomes from Ensembl.

1. ***install-ensembl-genome*** is a wrapper enabling to autmoatize the download (genome sequences, features, variations) and configuration tasks.

2. ***download-ensembl-genome*** downloads the genomics sequences and converts them in the raw format required for ***RSAT***.

3. ***download-ensembl-features*** downloads tab-delimited text files describing genomic features (transcripts, CDS, genes, ...).

4. ***download-ensembl-variations*** downloads tab-delimited text files describing genomic variations (polymorphism).

12.2.1 Installing genomes from Ensembl

The program ***install-ensembl-genome*** manages all the required steps to download and install a genome (sequence, features, and optionally variations) from Ensembl to ***RSAT***.

It performs the following tasks:

1. The option *-task genome* runs the program ***download-ensembl-genome*** to download the complete genomic sequence of a given organism from the ***EnsEMBL*** Web site, and formats it according to ***RSAT*** requirements (conversion from the original fasta sequence file to one file per chromosome, in raw format).

2. The option *-task features* runs ***download-ensembl-features*** to download the positions and descriptions of genomic features (genes, CDS, mRNAs, ...).

3. Optionally, when the option *-task variations* is activated, run ***download-ensembl-variations*** to download the description of genomic variations (polymorphism). Note that variations are supported only for a subset of genomes.

4. Update ***RSAT*** configuration (*-task config*) to make the genome available to other programs in the current ***RSAT*** site.

5. Run the additional tasks (*-task install*) required to have a fully functional genome on the local ***RSAT*** site: compute genomic statisics (intergenic sizes, ...) and background models (oligonucleotide and dyad frequencies).

6. With the option *-available_species*, the program returns the list species available on the Ensembl server, together with their status of availability for the 3 data types (genome sequence, features, variations). When this option is called, the program does not install any genome.

The detailed description of the program and the list of options can be obtained with the option *-help*.

```
## Get the description of the program + all options
install-ensembl-genome -help
```

Getting the list of available genomes

Before installing a genome, it is generally a good idea to know which genomes are available. For this, use the option *-available_species*.

```
## Retrieve the list of supported species on EnsEMBL
install-ensembl-genome -v 1  -available_species \
  -o available_species_ensembl.tab

## Read the result file
more available_species_ensembl.tab
```

Note: inter-individual variations are available for a subset only of the genomes available in ***EnsEMBL***. The option *-available_species* indicates, for each species, the availability (genome, features, variations). Obviously, the programs to analyse regulatory variations (***convert-variations***, ***retrieve-variation-seq***, ***variation-scan***) are working only for the genomes documented with variations.

Installing a genome from Ensembl

We can now download and install the complete genomic sequence for the species of our choice. For the sake of space and time economy, we will use a small genome for this manual: the budding yeast *Saccharomyces cerevisiae*.

Beware: some installation steps take a lot of time. For large genomes (e.g. Vertebrate organisms), the full installation can thus take several hours. This should in principle not be a big issue, since installing a genome is not a daily task, but it is worth knowing that the whole process requires a continuous connection during several hours.

```
## Install the genome sequences for a selected organism
install-ensembl-genome -v 2 -species Saccharomyces_cerevisiae
```

This command will automatically run all the installation tasks described above, except the installation of variations (see Section 12.2.3).

12.2.2 Installing genomes from EnsemblGenomes

The historical ***EnsEMBL*** project [1] was focused on vertebrate genomes + a few model organisms (*Saccharomyces cerevisiae*, *Drosophila melanogaster*, ...).

[1] http://www.ensembl.org/

A more recent project called ***EnsemblGenomes***[2] extends the ***EnsEMBL*** project to a wider taxonomic range (in Oct 2014, there are >15,000 genomes available at EnsemblGenomes, where as Ensembl only provides 69 genomes).

The program ***install-ensembl-genome*** supports the installation of genomes from ***EnsEMBL*** as well as ***EnsemblGenomes***. By default, it opens a connection to the historical ***EnsEMBL*** database, but the option *-db ensemblgenomes* enables to install genomes from the new project ***EnsemblGenomes***.

```
## Get the list of available species from the extended project
## EnsemblGenomes
install-ensembl-genome -v 2 -available_species -db ensemblgenomes \
   -o available_species_at_EnsemblGenome.txt
```

You can then identify your genome of interest in the file *available_species_at_EnsemblGenome.txt*, and start the installation (don't forget the option *-db ensemblgenomes*.

```
## Install Escherichia coli (strain K12 MG1665) from EnsemblGenomes
install-ensembl-genome -v 2 -db ensemblgenomes \
   -species Escherichia_coli_str_k_12_substr_mg1655
```

12.2.3 Downloading variations

The program ***download-ensembl-variations*** downloads variations from the ***EnsEMBL*** Web site, and installs it on the local ***RSAT*** site.

This program relies on ***wget***, which must be installed beforehand on your computer.

```
## Retrieve the list of supported species in the EnsEMBL variation database
download-ensembl-variations -v 1  -available_species
```

We can now download all the variations available for the yeast.

```
## Download all variations for a selected organism on your server
download-ensembl-variations -v 1 -species Saccharomyces_cerevisiae
```

Variation files are stored in a specific subfolder for the specified organism.

```
## Check the content of the variation directory for the yeast
make -f makefiles/variation-scan_demo.mk \
   SPECIES=Saccharomyces_cerevisiae ASSEMBLY=R64-1-1 \
   variation_stats
```

This command will indicate the location of the variation directory on your ***RSAT*** server, and count the number of lines for each variation file (there is one separate file per chromosome or contig).

[2] http://ensemblgenomes.org/

12.3 Importing genomes from NCBI BioProject

Tne BioProject database hosts the results of genome sequencing and transcriptome projects.

1. Open a connection to the Bioproject Web site
   http://www.ncbi.nlm.nih.gov/bioproject

2. Enter a query to select the organism of interest. E.g. `ostreococcus+tauri[orgn]`

3. If the organism genome has been sequenced, you should see a title "Genome Sequencing Projects" in the record. Find the relevant project and open the link.

   For example, for *Ostreococcus tauri*, the most relevant project is PRJNA51609
   http://www.ncbi.nlm.nih.gov/bioproject/51609

4. Take note of the *Accesssion* of this genome project: since a same organism might have been sequenced several times, it will be useful to include this Accession in the suffix of the name of the file fo be downloaded.

5. On the left side of the page, under *Related information*, click the link "*Nucleotide genomic data*". This will display a list of Genbank entries (one per contig).

6. **Important:** we recommend to create one separate directory per organism, and to name this directory according to the organism name followed by the genome project Accession number. For example, for *Ostreococcus tauri*, the folder name would be *Ostreococcus_tauri_PRJNA51609*.

   This convention will facilitate the further steps of installation, in particular the parsing of genbank-formatted files with the program ***parse-genbank.pl***.

7. In the top corner of the page, click on the *Send to* link and activate the following options.

   Send to > File > Genbank full > Create file

   Save the file in the organism-specific directory described in the previous step.

8. You can now parse the genome with the program ***parse-genbank.pl***. Note that ***parse-genbank.pl*** expects files with extension .gbk or .gbk.gz (as in the NCBI genome repository), whereas the BioProject genome appends the extension *.gb*. You should thus use the option *-ext gb*.

   ```
   parse-genbank.pl -v 2 -i Ostreococcus_tauri_PRJNA51609 -ext gb
   ```

   After parsing, run the program ***install-organism*** with the following parameters (adapt organism name).

   ```
   install-organism  -v 2 -org Ostreococcus_tauri_PRJNA51609 \
     -task config,phylogeny,start_stop,allup,seq_len_distrib \
     -task genome_segments,upstream_freq,oligos,dyads,protein_freq
   ```

12.4 Importing multi-genome alignment files from UCSC

12.4.1 Warning: disk space requirement

The UCSC multi-genome alignment files occupy a huge disk space. The alignments of 30 vertebrates onto the mouse genome (mm9 multiz30) requires 70Gb. If you intend to offer support for multi-genome alignments, it might be safe to acquire a separate hard drive for this data.

The complete data set available at UCSC in April 2012 occupies 1Tb in compressed form, and probably 7 times more once uncompressed. For efficiency reasons, it is necessary to uncompress these files for using them with the indexing system of ***peak-footprints***.

12.4.2 Checking supported genomes at UCSC

As a first step, we will check the list of supported genomes at the UCSC Genome Browser.

```
supported-organisms-ucsc
```

Each genome is assocaited with a short identifier, followed by a description. For example, several versions of the mouse genome are currently available.

```
mm10 Mouse Dec. 2011 (GRCm38/mm10) Genome at UCSC
mm9 Mouse July 2007 (NCBI37/mm9) Genome at UCSC
mm8 Mouse Feb. 2006 (NCBI36/mm8) Genome at UCSC
mm7 Mouse Aug. 2005 (NCBI35/mm7) Genome at UCSC
```

12.4.3 Downloading multiz files from UCSC

Multi-genome alignments at UCSC are generated with the program ***multiz***, which produces files in a custom text format called *maf* for Multi-Alignment file.

We show hereafter the command to download the mm9 version of the mouse genome, and install it in the proper directory for ***peak-footprints*** (*$RSAT/data/UCSC_multiz*).

```
download-ucsc-multiz -v 1 -org mm9
```

Beware: the download of all the multi-species alignments can take several hours for one genome.

The program will create the sub-directory for the mm9 genome, download the corespond-ing compressed multiz files (files with extension *.maf.gz*), uncompress them, and call ***peak-footprint*** with specific options in order to create a position index, which will be further used for fast retrieval of the conserved regions under peaks.

12.5 Installing genomes from NCBI/Genbank files

In the section 9, 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.

12.5.1 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 yo 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 contigous sequence, resulting from the assembly of short sequence fragments obtained during the sequencing. When a genome is completely sequenced and assembled, each chomosome 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 complemet)
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
```

12.5.2 Downloading genomes from NCBI/Genbank

The normal way to install an organism in ***RSAT***is to download the complete genome files from the NCBI [3], 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 [4] which presents two advantages?

- Genome files are compressed (gzipped), which strongly reduces the transfer and storage volume.

[3] ftp://ftp.ncbi.nih.gov/genomes/
[4] bio-mirror.net/biomirror/ncbigenomes/

- 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).

RSATincludes 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 genitamlium*.

```
## 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 dowload 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/
```

RSATparsers 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, of for the whole NCBI genome repository, but this requires sveral Gb of free disck space.

12.5.3 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
```

12.5.4 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 shoud 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
ILYLHEQIPVGQVLLGLPFQDAHKLSIGFTDDGDKRELRFRANGNIHKFPIRYDGDSNYH
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
TTATTCATCACAAGGACAAGAGCTTTGTTGGTTGACAGAGATGTTTTGCAGAAAGCCCTT
AAGGATGGTATTGCCTTGTTCAAGAAGAAACCAGTTGTTACTGAAGTAAATCTGACGACC
```

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.

12.5.5 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***.

12.5.6 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, i order to check that gene locations were corectly 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
```

12.5.7 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
```

12.5.8 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/
```

One 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.

12.6 Installing genomes from EMBL files

RSAT also includes a script ***parse-embl.pl*** to parse genomes from EMBL files. However, for practicaly 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 [5].

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 -1 $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_hanseni*.

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

[5] 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 -1 $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_hanseni/* \
  $RSAT/data/genomes/Debaryomyces_hansenii/genome

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