Regulatory Sequence Analysis Tools
Installation guide

Jacques van Helden
jvhelden@ulb.ac.be
http://www.bigre.ulb.ac.be/Users/jvanheld/
Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe)
Laboratory of Genome and Network Biology
Université Libre de Bruxelles, Belgium
http://www.bigre.ulb.ac.be/

February 26, 2008

Contents

Chapter 1

Description and requirements

1.1 Description

This documents describes the installation procedure for the software package **Regulatory Sequence Analysis Tools** (***RSAT***).

1.2 Requirements

1.2.1 Operating system

RSATis 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 librairies, because I did not find a cygwin version of GD.pm)

RSATis 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

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***.

Math::CDF 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. Note that this librairy is currently restricted to a precision of 1e-16. 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.

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

Chapter 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 Installation from the CVS repository

2.1.1 CVS configuration

You need to indicate your ***cvs*** 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).

```
cvs -d [mylogin]@cvs.scmbb.ulb.ac.be:/cvs/rsat co rsa-tools
```

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 -xpzf rsa-tools_yyyymmdd.tar.gz
```

2.3 Adding *RSAT* 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 you shell, you can type

   ```
   echo $SHELL
   ```

   Now, if we assume that ***RSAT*** have been installed in the directory

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

   you should type the following command.

   If your shell is bash:

   ```
   export RSAT=/home/rsat/rsa-tools
   ```

   If your shell is csh or tcsh, you need to type a slightly different command:

   ```
   setenv RSAT /home/rsat/rsa-tools
   ```

2. Add the path of the ***RSAT***perl scripts and binaries to your path.

   If your shell is bash:

   ```
   export PATH=${PATH}:${RSAT}/bin
   export PATH=${PATH}:${RSAT}/perl-scripts
   ```

   If your shell is csh or tcsh:

   ```
   set path=($path $RSAT/bin)
   set path=($path $RSAT/perl-scripts)
   rehash
   ```

   (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.

2.4 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 rsa-tools
make -f makefiles/init_RSAT.mk init
```

2.5 Configuring *RSAT*for utilization on the command line

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
```

You need to edit this file and specify the parameters of your local configuration. In particular, it is essential to specify the variable RSAT, which specifies the RSAT main directory.

2.6 Downloading genomes

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

2.6.1 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/Eukaryota/*)

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

2.6.2 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 downlaod 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.

2.6.3 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*.

Importing a single organism

The makefile script *makefilesinit_RSAT.mk* includes a target to install and configure a single organism on your RSAT site.

```
cd \$RSAT

# Download a single genome from the RSAT web server.
# This requires the program wget.
make -f makefiles/init_RSAT.mk download_one_genome ORG=Saccharomyces_cerevisiae

# Declare the newly downlaoded genome as a supported organism
make -f makefiles/init_RSAT.mk configure_one_genome ORG=Saccharomyces_cerevisiae
```

You can now 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   Saccharomyces cerevisiae
```

Importing another organism

You can now proceed exactly in th same way to install any organism of your choice. For example, if you want to install Escherichia coli K12? you can run the ffollowing commands.

```
cd \$RSAT

# Download a single genome from the RSAT web server.
# This requires the program wget.
make -f makefiles/init_RSAT.mk download_one_genome ORG=Escherichia_coli_K12

# Declare the newly downlaoded genome as a supported organism
make -f makefiles/init_RSAT.mk configure_one_genome ORG=Escherichia_coli_K12

## Check that the new genome has bee added to the list of supported organisms
supported-organisms
```

Chapter 3

Testing the command-line tools

3.1 Testing the access to 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.

3.2 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)

3.3 Testing the graphical scripts

RSATincludes two graphical tools, ***feature-map*** and ***XYgraph***. These tools require the following 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.

3.4 Further steps

The installation is now finished, you can start the user's guide.

In case you would like to install additional genomes that are not supported on ***RSAT***main server, the next chapter indicates yiou how to proceed.

Chapter 4

Installing third-party programs

The ***RSAT***distribution only contains the programs developed by Jacques van Helden. A few additional programs, developed by third parties, can be integrated in the package. All third-party programs may be loacated 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 used by the program ***purge-sequences***, for the detection of sequence repeats.

seqlogo : developed by Thomas D. Schneider, is used 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*** (ghostscript) <http://www.cs.wisc.edu/~ghost> to create PNG and PDF output, and ***ImageMagic's convert*** <http://www.imagemagick.org> 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).

I 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.

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/meme/website/meme-download.html
MotifSampler	Gert Thijs	http://www.esat.kuleuven.ac.be/~thijs/download.html
gibbs	Andrew Neuwald	ftp://ftp.ncbi.nih.gov/pub/neuwald/gibbs9_95/

Table 4.1: Programs from other developers which are complementary to the ***RSAT***package.

Chapter 5

Installing genomes from NCBI/Genbank files

In the section 2.6, 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.

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.

5.1.1 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.

5.1.2 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
```

5.1.3 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
```

5.1.4 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
```

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 [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?

[1] *ftp://ftp.ncbi.nih.gov/genomes/*
[2] *bio-mirror.net/biomirror/ncbigenomes/*

- 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).

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.

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
```

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.

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***.

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
```

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
```

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.

5.8.1 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.

Chapter 6

Installing genomes from EMBL files

RSATalso 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 [1].

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
```

[1] *http://natchaug.labri.u-bordeaux.fr/Genolevures/download.php*

```
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.

6.0.2 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*. 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
```