Logol usage

Olivier Sallou olivier.sallou@irisa.fr

June 20, 2012

Abstract

Logol is a pattern search and matching tool on genomic sequences.

Contents

1 Introduction

Logol is a logol grammar interpreter and a pattern search tool. It takes as input a biological sequence (DNA, RNA or protein), and a grammar file. The grammar is a logol grammar that describes a pattern to be found in the input sequence. Logol analyses the grammar and executes a program to match the pattern on the sequence. It returns a result file containing the matches with all required details.
Logol grammar is a highly descriptive language dedicated to biological sequence analysis. It defines sequence information with allowed mutations and morphism. Models and variables can be used to express a specific pattern and can be used to find repetition of a pattern along the sequence.
Logol is open source, and free of use.

1.1 History

- 2011/09/29 Creation

1.2 License

License CeCILL

2 Definitions

The term should will be used when an operation is not mandatory but highly recommended to perform the operation in the best conditions.
The term must will be used when a condition is mandatory to meet the program requirements.
The term can will be used to describe a program option (command-line or configuration or installation related) and is not limited to this option.
The term may will be used to define a condition that is not dependent upon the current program (cpu speed for example, cluster queue management...).

3 Installation

3.1 Prerequisites

- JRE 1.6+
- Ruby
- Vmatch or Cassiopee software is required to run Logol. It is advised to install one first.
- SWI-Prolog if using package or SWI-prolog compilation.
- Sicstus 4+ if using Sicstus prolog compilation (not needed at runtime)

Cassiopee is a search with dna ambiguity and error support. Vmatch has better performances over large sequences (genomes) but does not support ambiguity.
Default is to use Cassiopee, which is open source. This gem must be installed manually:

```
gem install cassiopee
```

Vmatch can be obtained from S. Kurtz at http://www.vmatch.de/.

3.2 From a package

To install the Logol package, copy the install package file to the final system (Linux) and execute it (dpkg,rpm). Configuration is located in /etc/logol and results are available in /var/lib/logol/results. Default is to use Cassiopee Ruby gem.

3.3 From source

Compilation additionally requires: ant 1.7+, jdk 1.6+, SWI-prolog 5.10+ or Sicstus 4+
Extract the code from SVN:
`svn checkout https://scm.gforge.inria.fr/svn/logolexec/vX.Y/LogolExec Logol`
Go in the directory and execute compilation:
`ant test_sicstus`
or
`ant create-jar`
`ant test_swi`
This will create executables and run some unit tests. All tests should pass before configuring the application.

4 Configuration

The file logol.properties contains the default configuration.
Two directories are required to run Logol. The first one is the place where result files will be written. If running on a cluster, this directory must be accessible from all the nodes of the cluster. The second one is a temporary directory to hold temporary files (deleted after work).
Many parameters are defined in this file, a description is available for each.

```
##
# Configuration file for LogolMatch. Some properties may be override by
# command-line parameters.
##
# Minimum size to use to split a file to parallelize treatments
minSplitSize=2000000
# Maximum size to display a solution, 0 is no limit. Above the limit,
# the variable is replaced by "-" character
maxResultSize=0
# Maximum size of a solution (used to optimize the search)
#maxMatchSize=30000
maxMatchSize=0
# Temporary directory used for the analysis. Should be local to the node
workingDir=/tmp/Logol
# Directory where to place the results. In case of cluster usage,
# result must be a shared directory between nodes
dir.result=/tmp/Logol
# Maximum length of a spacer when looking forward for a match
#maxSpacerLength=10000
```

```
maxSpacerLength=0
# Maximum length of a variable in a match
#maxLength=1000
maxLength=0
# Minimum length of a variable in a match
minLength=2
# Default strategy to use, 1 must be keep by default
parentStrategy=1
# Number of processor on computer running the analysis,
# or number of available processors on DRM  nodes.
# Can speed up the search process when sequence file can be splitted.
nbProcessor=1
# Max Number of jobs to run for a single sequence when used in DRM config.
nbJobs=1
# Default number to limit number of results (must be above 0)
maxSolutions=100
# Minimum size of tree index. In case of use of small sequences,
# should be set to 2, else use 4. (see vmatch manual). This applies for all se
minTreeIndex=2
# Host where is smtp server (if email required)
smtp.host=localhost
# Mail user for smtp host
mail.user=
# DRM queue command if a specific queue is to be used
# Example for SGE: drm.queue= -q long
drm.queue=
# Suffix tool   0: Cassiopee (default), 1: Vmatch
suffix.tool = 0
suffix.path=
```

Default configuration (except directories) should apply to most of usages, but parameters should be carefully studied to improve the performances of the software.
The configuration file described here is the default configuration file. However, a per-request configuration file can be specified in command-line, this allows to adapt the parameters to specific queries/sequences.

5 Usage

5.1 Sequences

Input sequences must be in a single file in NCBI Fasta format. All headers must be like:

```
>gi|51511735|ref|nc_000018.8|nc_000018 test sequence for logol validation
acgcgcggta
```

References are not checked and can be any value.

5.2 Web interface - Genouest web site only

For the web interface, connect to the web container (http://webapps.genouest.org/LogolDesigner) at the application URL.
For LogolDesigner, an index page provides links to the online help and software as well as some screencasts.
For the Logolanalyser, an online help is available under URL at org.irisa.genouest.logol.LogolAnalyser/help/LogolAnalyser.html

5.3 Command-line

Options specified in command-line superseeds default configuration options.
Use programname.sh -h to get a list of available options.

5.3.1 LogolMultiExec.sh

LogolMultiExec.sh is an intermediate program only. It takes as input one or more sequences, and dispatch them to LogolExec.
If configuration allows it, it can also split a sequence in smaller part, in such a case, it is also in charge of merging the results fr the sequence. On DRM systems, it creates a new job for each (sub-)sequence. On non-DRM system, all (sub-)sequences are executed sequentially.
A man page is available for options.

5.3.2 LogolExec.sh

Called by LogolMultiExec.sh, it can be run directly when using a single sequence. A man page is available for options.

5.3.3 Grammar checks

To check a Logol grammar file, one simply execute LogolExec.sh -check -g mygrammarfile.

6 Results

Results are zipped in a single file. There is one result file per input sequence, in XML format.
The model is the id of the model defined in the grammar. Variables are the detailled value

of the match according to the grammar. The Id of the match is unique for the sequence result file. Reverse complement search, when selected, will have a begin position higher than end position.
Here is the DTD of the XML document:

```
<!ELEMENT sequences ( fastaHeader, grammar, model, match+ ) >
<!ELEMENT fastaHeader ( #PCDATA ) >
<!ELEMENT grammar ( #PCDATA ) >
<!ELEMENT match ( model, id, begin, end, errors, distance, variable+ ) >
<!ELEMENT model ( #PCDATA ) >
<!ELEMENT id ( #PCDATA ) >
<!ELEMENT begin ( #PCDATA ) >
<!ELEMENT content ( #PCDATA ) >
<!ELEMENT end ( #PCDATA ) >
<!ELEMENT errors ( #PCDATA ) >
<!ELEMENT distance ( #PCDATA ) >
<!ELEMENT variable ( begin, end, size, errors, content, text ) >
<!ELEMENT size ( #PCDATA ) >
<!ELEMENT text ( #PCDATA ) >
<!ATTLIST variable name CDATA  #REQUIRED >
```

7 FAQ

- Out of memory issue when running LogolMatch programs:

  Depending on sequence size, it may be required to increase the JVM max memory. To do so, in case of problem, edit the .sh files and increase the -Xmx parameter value (should be at least 2 times the sequence size).

- How to add a cost specific function:

  Create a program in LogolMatch/tools directory. Script should return the number of errors found to stdout and have execution rights for the LogolMatch? user.

- I have some grammar errors when running a search

  Look at the error message, it usually specify the kind of error.

- I have no results or result file is empty

  Look the stdout information messages, there could be grammar issues, or suffix tree file creation issue. In case of use via a DRM system, edit the generated xxx.ojobid and xxx.ejobid to get job stream information. If this is not enough, it is possible to increase the level of information in the file LogolMatch/log4j.properties. Modify the level of log4j.logger.org.irisa.genouest.logol and lo4j.logger.org.irisa.genouest.logol.StreamGobbler from ERROR to INFO or DEBUG.