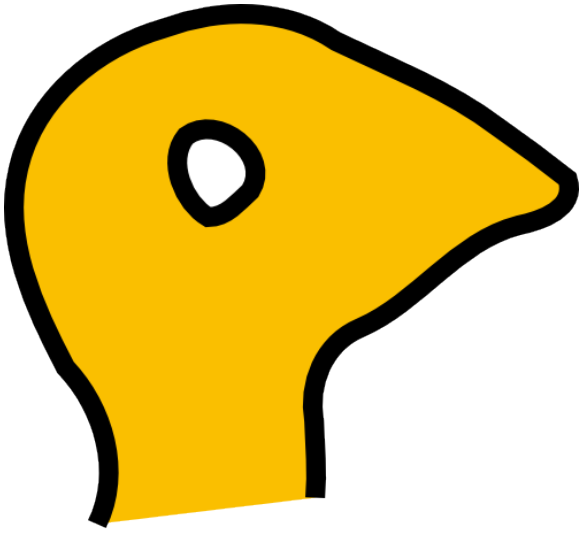

Plant & Food RESEARCH
RANGAHAU AHUMĀRA KAI

Moa

author: Mark Fiers

date: Tue Sep 29 12:32:39 NZDT 2009

git: master 3cae30ba5b53741e79f108097f44a8b799e6796a

Contents

Chapter 1

Introduction

‘ ’ NOTE” Both the software and the manual are still under heavy development.

Moa is a piece of software build around GNU Make[5] that allows you to use Gnu Make run bioinformatics pipelines.

GNU Make is an excellent tool to automate the compilation of software. Gnu make determines how a file is created, what it's dependencies are, and what needs to be executed. Gnu Make uses so called Makefiles to describe a project. A bioinformatics project is often of the same form as compiling software.

Moa wraps a set of common bioinformatics tools as Makefiles. Features of Moa are:

- A uniform interface; all Moa makefiles use a central library that provides a uniform, command line, interface to configuring and executing jobs.
- Interaction; templates are designed to interact with each other, hence make it easy to build pipelines from these buliding blocks.
- Parallel execution; Gnu make facillitates parallel execution of jobs.

Apart from a set of template Makefiles, the Moa contains several other

- moaBase; a central library describing a number of central routines used by all Makefiles
- The “moa” helper script; a frontend to using Moa.
- Additional helper scripts; several of the template files require helper scripts that are part of the moa package.
- Couchdb interface; Moa is able to store information on each job in a couchdb. See chapter XX.

1.0.1 Example session

To really understand how easy it is to use Moa, a sample session:

```
mkdir test
cd test
moa new lftp
```

Chapter 2

Installation

2.1 Prerequisites

Moa is developed on Ubuntu[9] and RHEL[6] Linux and is expected to operate without much problems on most modern Linux distributions. Moa is depends on the following list of software. The version numbers are an indication, not strict prerequisites. Other, even older, versions might work.

- Gnu Make 3.81
- Git 1.6. To download the Moa software. Alternatively it is possible to download a tarball.
- Python 2.6. Python version 2.5 and lower will not work, several supporting scripts use 2.6 specific functionality
- Bash. Many of the embedded scripts expect the Bash shell. Luckily, Bash is the default shell of almost all Linux distributions.
- Gnu Make Standard Library (GSML). A set of standard routines for Gnu Make. GSML is embedded in this distribution.

2.1.1 Couchdb

Moa can use Apache's Couchdb as a central storage of information on Moa jobs. allowing other Moa jobs to refer hereto. If you want to use this, the following prerequisites are added to the list:

- Apache Couchdb 0.9.0. Only when using couchdb functionality, see the chapter on Couchdb

- Couchdb-python. Only when using couchdb functionality, see the chapter on Couchdb

For more information, read the chapter on couchdb.

2.1.2 Bioinformatics tools

Each of the wrapped tools, obviously, requires that these tools are present. Usually, unless mentioned otherwise, Moa expects all tools to be installed in the system PATH. All requirements are described in the reference chapter.

2.1.3 Deciding where to install Moa

You will need to choose a location to install Moa to, this usually depends on who is going to use the software. Moa can be installed system wide for all users of this machine, for example in `/opt/moa`. However, if you will be the only person using Moa, install it in your home directory, for example under `~/moa`. The remainder of this chapter assumes an installation in your home directory.

2.2 Downloading Moa

Moa is hosted at github:

`http://github.com/mfiers/Moa`

Currently there are no stable releases so the best option is to download the latest version of the software, this can be done using Git or by downloading a source archive.

2.2.1 Using Git

Using git is a good choice as long as there are no releases. Git makes it very easy to stay up to date with the latest version and, even better, allows anybody to submit bugfixes to the Moa repository (more on that later). To download Moa using Git, enter the following commands (assuming you're installing Moa in your home directory):

```
cd ~
git clone git://github.com/mfiers/Moa.git moa
```

2.2.2 Downloading an archive

As an alternative, it is possible to download an (automatically generated) archive of the latest Moa version here, for example, using the following commands:

```
wget http://github.com/mfiers/Moa/tarball/master
```

The archive that is downloaded will have a rather long name that looks something like `mfiers-Moa-b13ddf78c6a1ae9a714c7d9979a1b1de0ed08462.tar.gz`. This archive needs to be unpacked in a temporary directory and then moved to its final location:

```
mkdir /tmp/moa_install
cd /tmp/moa_install
tar xvzf mfiers-Moa-b13ddf78c6a1ae9a714c7d9979a1b1de0ed08462.tar.gz
mv mfiers-Moa-b13ddf78c6a1ae9a714c7d9979a1b1de0ed08462.tar.gz ~/moa
```

After following either procdure; downloading the archive or using Git, the source code tree should be should be in its final location. The tree should contain the following directories:

```
./moa
    ./bin
    ./doc
    ./etc
    ./lib
    ./template
    ./test
    ./util
    ./www
    ./COPYING
    ./INSTALL
    ./README
    ./VERSION
```

2.3 Configuration

Configuration of Moa is simple: The Moa `/bin/` directory must be included in the PATH and an environment variable must be set pointing to the Moa directory. The easiest way to do this is by adding the following lines to your `.bashrc`:

```
export PATH=/opt/moa/bin:$PATH
export MOABASE=/opt/moa
```

and run:

```
source .bashrc
```

Also, if you are running Moa to be used by all users of your system system, please remember the file attributes correctly:

```
chmod a+rX -R $MOABASE
chmod a+rx $MOABASE/bin/*
```

Chapter 3

Using Moa

3.1 Creating a pipeline

3.1.1 Guiding principles

Most (bioinformatics?) projects start small, and grow over time. From that perspective it is advisable to give the organization of your project some thought on forehand.

When using Moa the separate analysis steps of a pipeline each reside in a directory. The output data of each analysis usually resides in the same directory or a subdirectory thereof. Moa has templates that assist in downloading and organizing data. This has as result that all project data in a Moa project will be organized in a directory tree on your filesystem. Such a tree must represent both the data in logical way as well as the analysis pipeline organization.

Although there are likely multiple ways of achieving a healthy organization of a Moa project, this manual proposes the following organization:

- On the highest levels organize your project according to fundamental divisions in the project or data source. For example, if you work with data from multiple organisms, that might be a good top level division.
- On lower levels start organizing your annotation pipeline. Since most

3.1.2 Setting up analysis steps

3.2 Running a pipeline

3.2.1 Running one job

3.2.2 Running a series of jobs

Chapter 4

Using GBrowse

The Generic Genome Browser (Gbrowse)[8] is a popular tool for ...

to be written

Chapter 5

Couchdb

Couchdb[2] is a novel type of database that is almost copmletely unlike a SQL database. In it's simplest form it is a high performance key-value datastore. Moa uses Couchdb to store information on all analyses performed by Moa. This means that for each job that Moa performs, a record is created in the Couchdb database. This record has a unique identifier, called `jid`. Moa creates jids on the fly by combining the template name, the directory name and a unique identifier (to prevent collisions). These names are not always very descriptive, so it is advisable to set a jid manually. This is possible using the following command (do not use spaces!):

```
make set jid=SensibleName
```

Each time Moa executes, the analysis record in Couchdb is updated. The record contains all parameters used, the type of analysis done and the location (current directory) of the analysis. It is possible to update the couchdb record without running the analysis using:

```
make register
```

Moa/Couchdb records are a set of key/value pairs, that look like this:

The most important application of Couchdb in Moa is to refer to other jobs using Couchdb identifiers. In a Moa project without couchdb references to the output of other jobs is done by defining the path to that analysis. If, at a certain moment, the project structure needs to be rearranged, it can be hard to discover which path references need to be updated. Use of couchdb solves this, instead of refering to a path, it is now possible to refer to a jid / value combination.

allows a user to refer to another Moa job by the identifier, as opposed to using (relative) directories. The biggest advantage is that is now possible to shuffle your directories around without breaking the pipeline structure.

5.0.3 Configuration

Please follow the couchdb documentation to set up a local server. Moa has been developed with the latest version of Couchdb (currently 0.9.1). It might be possible to use an older version, but that has not been tested.

All Moa configuration for couchdb is done in `$MOABASE/etc/moa.conf.mk`.

The default setting of Moa is to not use couchdb. This can be overrided by setting: usecouchdb=T

Moa expects a Couchdb server on `localhost:5984`. This can be overridden using: couchserver=other.server:portnumber

All information

5.0.4 Using couchdb with Moa

Instead of using `make set key=value`, couchdb variables are set using `make cset jid\^{}key`

LocalWords: jid SensibleName

Chapter 6

Extending Moa

This chapter describes how to create new templates for use with Moa. Creating a template is not very difficult, once you have a basic understanding of how Makefiles work. Pprobably the hardest part is ensuring that templates are able to interact with other templates.

A template is, as stated, not much more than Makefile that adheres to certain standards. To understand how Makefiles work, please read the Gnu Make Manual. Note that creating Makefiles can be somewhat complex at first, given that the logic differs from scripting languages. The easiest way to do this is to work from an existing Makefile.

Each template exists of the following parts:

- Definition
- Include moaBase
- Implementation

The order in which everything is defined in a template is very important! It is advisable to not define variables depending on other variables in the definition phase.

In the remainder of this chapter we will describe a simple template that creates the reverse complement of a FASTA file using the EMBOSS[7] revseq utility

6.1 Definition

The definition is a list of variables defining what your template does and giving Moa information on how to use this template.

6.1.1 Describing the new template

The following variables define what your template does. These variables are used in generating the help files, the manual and the website.

Identifier	Description
moa_title	The title for this template
moa_description	A short description of this template
moa_ids	A unique, short, identifier for this template

Example:

```
moa_title = Reverse Complement
moa_description = This Moa template takes a set of      \
    input FASTA sequences and determines the reverse    \
    complement using the EMBOSS revseq utility.
moa_ids += revcom
```

Note that lines are allowed to break over multiple lines, given that each line that continues to the enxt line ends with a backslash. No spaces are allowd after the backslash and the new line must be indented (with at least one space).

6.1.2 Moa organizational units - moa_ids

In the previous chapter, both title and description are fairly self evident. The `moa_ids` variable is, however, more complicated. Each template must have, at least one, unique, preferaby short, identifier linked to it. This `moa_id` helps in defining variable space for each template. The moa_id returns when defining template specific variables and targets. All template specific variables have the moa_id as a part of their name, so do the major targets of a template.

Use of uniqued ids allow Moa to stack several templates into a larger, more complicated, templates. This might be usefull describing a set of resembling tasks that have a lot of overlapping code. Another powerful use is to create complex jobs that execute a mini-pipeline in one run. For example, gathering a filter a specific set of sequences (using the gather template) and creating a BLAST database from that.

Using ids allows functional sepeartion of tasks withing a template, or within a stacked template. It is advisable to start creating templates with only one task. For each task, a set of specific variables need to be defined.

Given that a template can define multiple tasks, a moa_id are added to the moa_ids array using the following syntax:

```
moa_ids += revcomp
```

6.1.3 taks specfic variables

Identifier	Description
moa_title	The title for this template
moa_description	A short description of this template
moa_ids	A unique, short, identifier for this template

6.2 Include moaBase

To includie moaBase add the following line to your Makefile:

```
include $(shell echo $$MOABASE)/template/moaBase.mk
```

6.3 Implementation

6.3.1 define dependant variables

6.3.2 define targets

Each task, identified by a unique moa_id, needs to define a set of four targets. For example, if your template defines: `moa_id += revomp` then the following four targets are expected to be defined and are automatically executed:

- MOA_ID - revcomp
- MOA_ID_prepare - revcomp_prepare
- MOA_ID_post - revcomp_post
- MOA_ID_clean - revcomp_clean

Each of these targets must be defined in a new template, although they could can be empty. In the following paragraphs, each of these targets are discussed, in the order that they are executed.

Prepare execution: revcomp_prep

The MOA_ID_prep target contains commands that are executed prior to the main run. In the case of reverse complementing sequences this target can be used to create a directory to store the output sequences. Using a separate subdirectory to

Round up execution: revcomp_post

Clean up: revcomp_clean

- **revcomp**: the main target, executes the main task of this template. In this case it takes a set of input sequences and write the reverse complement back to disk.
- **revcomp_prepare**:
- **revcomp_post**: Optional commands to be exeucted after everything is finished. In the case of reverse complementing a set of sequences there is not much to do. The BLAST template, however, uses this target to create an overall BLAST report
- **revcomp_clean**: Cleans up all reverse complemented sequences

Appendix A

Template reference

This chapter contains, as a reference, all help documentation of all templates currently in the Moa repository. It is possible to get the (latest) version of the help for each template by running:

```
make help
```

in a directory with a Moa analysis. Moreover, if an Makefile links to multiple templates, on the fly generated help will detail all targets that can be used and all parameters that can be defined.

A.1 blast

Wraps BLAST[1], the most popular similarity search tool in bioinformatics

A.1.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (blast)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

blast Running BLAST takes an input directory (*blast_input_dir*), determines what sequence files are present (with the parameter *blast_input_extension*) and executes BLAST on each of these. Moa BLAST is configured to create XML output (as opposed ot the standard text based output) in the *./out* directory. The output XML is subsequently converted to GFF3[4] by

the custom *blast2gff* script (build around biopython[3]). Additionally, a simple text report is created.

blast_report Generate a text BLAST report.

A.1.2 Parameters

Required parameters

blast_db Location of the blast database

blast_gff_source source field to use in the gff

Optional parameters

blast_input_dir directory containing the input sequences

input_extension Extension of the input files

blast_program blast program to use (default: blastn)

blast_eval e value cutoff

blast_nohits number of hits to report

blast_nothreads threads to run blast with (note the overlap with the Make -j parameter)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.2 blastSingle

Wraps BLAST[1], the most popular similarity search tool in bioinformatics

A.2.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (blast)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

blast Running BLAST takes an input directory (*blast_input_dir*), determines what sequence files are present (with the parameter *blast_input_extension*) and executes BLAST on each of these. Moa BLAST is configured to create XML output (as opposed ot the standard text based output) in the *./out* directory. The output XML is subsequently converted to GFF3[4] by the custom *blast2gff* script (build around biopython[3]). Additionally, a simple text report is created.

blast_report Generate a text BLAST report.

A.2.2 Parameters

Required parameters

blast_input_file Input fasta file to BLAST

blast_db Location of the blast database

blast_gff_source source field to use in the gff

Optional parameters

blast_input_dir directory containing the input sequences

input_extension Extension of the input files

blast_program blast program to use (default: blastn)

blast_eval e value cutoff

blast_nohits number of hits to report

blast_nothreads threads to run blast with (note the overlap with the Make -j parameter)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.3 blastdb

A.3.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (blastdb)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

blastdb :

A.3.2 Parameters

Required parameters

bdb_name Database name to create.

Optional parameters

bdb_input_dir Dir with the input fasta files, defaults to ./fasta

bdb_input_extension extension of the input sequence files, defaults to fasta

bdb_fasta_file The file with all FASTA sequences for the blastdb concatenated. This can be used as an alternative to defining and . Morover. If all your sequences are already in a single file, then using this parameter prevents duplication of that file.

bdb_protein Protein database? (T)rue) or not (F)alse (default: F)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.4 blat

A.4.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (blat)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

blat :

A.4.2 Parameters

Required parameters

blat_db Blat db file (multifasta)

blat_gff_source undefined

Optional parameters

blat_input_file input query file. If this variable is not defined, the combination of blat_input_dir and blat_input_extension is used to find a list of input files

blat_input_dir source field in the generated gff

blat_input_extension extension of the input files

blat_eval evalue cutoff to select the reported hits on (defaults to 1e–15)

blat_db_id_list a sorted list of db ids and descriptions, enhances the report generated

blat_db_type type of the database (dna, prot or dnax)

blat_query_type type of the query (dna, rna, prot, dnax or rnax)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.5 bowtie

A.5.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (bowtie)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

bowtie :

A.5.2 Parameters

Required parameters

bowtie_db Bowtie db

bowtie_input_dir input dir with the query files

Optional parameters

bowtie_input_extension Extension of the input files, defaults to fastq

bowtie_input_format Format of the input files, defaults to fastq

bowtie_extra_params extra parameters to feed bowtie

bowtie_output_name undefined

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.6 bowtiedb

A.6.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (bowtiedb)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

bowtiedb :

A.6.2 Parameters

Required parameters

bowtiedb_input_dir The reference sequence to build a bowtie database with.

bowtiedb_name Name of the bowtie index to create

Optional parameters

bowtiedb_input_extension Extension of the input files, defaults to 'fasta'

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.7 cleanFasta

A.7.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (clean_fasta)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

clean_fasta Cleanup of a FASTA file (in place!)

A.7.2 Parameters

Required parameters

Optional parameters

cf_input_dir undefined

cf_input_extension undefined

sed_command The sed command cleaning the code, defaults to '/^>/!s/[ACGTNacgtn]/N/g'

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.8 clustalgroup

A.8.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (clustalgroup)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

clustalgroup run clustalw

A.8.2 Parameters

Required parameters

cwg_input_dir This set of sequences to run clustalw on

Optional parameters

cwg_input_extension undefined

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.9 clustalpair

A.9.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (clustalpair)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

clustalpair run clustalw

A.9.2 Parameters

Required parameters

input_dir_a This set is compared to the sequences in input_dir_b. only a forward comparison is made (a against b, not the other way round)

input_dir_b The set to compare against

Optional parameters

input_extension Extension of the input files

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.10 clustalw

A.10.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (clustalw)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

clustalw run clustalw

A.10.2 Parameters

Required parameters

input_dir_a This set is compared to the sequences in input_dir_b. only a forward comparison is made (a against b, not the other way round)

input_dir_b The set to compare against

Optional parameters

input_extension Extension of the input files

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.11 concatenate

A.11.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (concatenate)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

concatenate Concatenate a set of FASTA files

A.11.2 Parameters

Required parameters

input_dir Directory with the input data

name A unique project name defining this job. Cannot have spaces.

Optional parameters

input_extension Extension of the input files

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.12 create.gbrowse.db

A library that aids in uploading FASTA and GFF to a Generic Genome Browser database. This template is only to be used embedded in another template. This library expects that the following variables are preset; gup_fasta_dir, gup_gff_dir, gffsource gup_upload_fasta, gup_upload_gff

A.12.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (upload2gbrowse)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

upload2gbrowse :

initGbrowse Clean & initalize a gbrowse database. **Warning: all data will be lost!**

gupLock Prevent this job from uploading anything to the Generic Genome Browser database

gupUnlock Allow this job to upload to the Generic Genome Browser database

A.12.2 Parameters

Required parameters

gup_user gbrowse db user. If not defined, this defaults to 'moa'.

gup_db gbrowse database. If not defined, this defaults to 'moa'.

gup_gffsource the gff source field, used in batch operations

Optional parameters

gup_gff_extension extension of the GFF files to upload (.gff)

gup_fasta_extension extension of the FASTA files to upload (.fasta)

gup_upload_limit Do not upload more that this number of files (infinite)

gup_upload_fasta upload fasta to gbrowse (T/F)

gup_upload_gff upload gff to gbrowse (T/F)

gup_force_upload upload to gbrowse, ignore gup_lock and upload all, not only files newer that upload_gff or upload_fasta

marks_extensions Add some extensions to the Gbrowse database to be initalized, for use by Mark.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.13 dottup

A.13.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (dottup)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

dottup run clustalw

A.13.2 Parameters

Required parameters

dottup_input_dir_a This set is compared to the sequences in input_dir_b. only a forward comparison is made (a against b, not the other way round)

dottup_input_dir_b The set to compare against

Optional parameters

dottup_input_extension undefined

dottup_wordsize undefined

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.14 empty

A.14.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (empty)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

empty :

A.14.2 Parameters

Required parameters

Optional parameters

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.15 fasta2gff

Derive GFF from a FASTA file, usually to accompany the Sequence for upload to a generic genome browser database.

A.15.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (fasta2gff)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

fasta2gff :

A.15.2 Parameters

Required parameters

f2g_gffsource Source to be used in the gff

Optional parameters

f2g_input_dir Directory with the input fasta (default: ./fasta)

f2g_output_dir Directory with the output gff (default: ./gff)

f2g_input_extension glob pattern of the fasta files (default: *.fasta)

f2g_options options to be passed to the fasta2gff script

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.16 gap4export

A.16.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (gap4export)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

gap4export Export data from an assembly using gap4

A.16.2 Parameters

Required parameters

ge_input_dir Directory with the input data

ge_input_pattern file name pattern

Optional parameters

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.17 gather

A.17.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (gather)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

gather gather files

A.17.2 Parameters

Required parameters

g_input_dir list of directories with the input files

g_input_pattern glob pattern to download

Optional parameters

g_name_sed undefined

g_output_dir Output subdirectory, defaults to '.'

g_process Command to process the files. If undefined, hardlink the files.

g_limit limit the number of files gathered (with the most recent files first, defaults to 1mln)

g_powerclean Do brute force cleaning (T/F). Remove all files, except moa.mk & Makefile when calling make clean. Defaults to F.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.18 genemarks

predict genes using geneMarkS

A.18.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (genemarks)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

genemarks :

A.18.2 Parameters

Required parameters

genemarks_input_dir directory containing the input sequences

genemarks_matrix the matrix to use

Optional parameters

genemarks_gff_source source field to use in the gff. Defaults to geneMarkS

genemarks_input_extension input file extension. Defaults to 'fasta'

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.19 getFromNcbi

Download a set of sequences from NCBI based on a query string (ncbi_query) and database (ncbi_db). This tempate will run only once (!), after a succesful run it creates a 'lock' file that you need to remove to rerun

A.19.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (getFromNcbi)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

getFromNcbi Downloads from NCBI

A.19.2 Parameters

Required parameters

ncbi_db NCBI database (for example nucest)

ncbi_query NCBI query (for example txid9397[Organism%3Aexp])

Optional parameters

gfn_sequence_name Sequence name to download. When this parameter is set, the template assumes that only one sequence is to be downloaded, the rest will be discarded.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.20 getorf

Predicts open reading frames using the EMBOSS[?] getorf tool.

A.20.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (getorf)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

getorf :

A.20.2 Parameters

Required parameters

getorf_input_dir undefined

Optional parameters

getorf_gff_source undefined

getorf_input_extension undefined

getorf_minsize minimal nucleotide size of the predicted ORF, (**30**)

getorf_maxsize maximal nucleotide size of the predicted ORF, (**1000000**)

getorf_circular Is the sequence linear (Y/**N**)

getorf_table Genetic code to use: **0** Standard; 1 Standard with alternative initiation codons; 2 Vertebrate Mitochondrial; 3 Yeast Mitochondrial; 4 Mold, Protozoan, Coelenterate Mitochondrial and Mycoplasma/Spiroplasma; 5 Invertebrate Mitochondrial; 6 Ciliate Macronuclear and Dasycladacean; 9 Echinoderm Mitochondrial; 10 Euplotid Nuclear; 11 Bacterial; 12 Alternative Yeast Nuclear; 13 Ascidian Mitochondrial; 14 Flatworm Mitochondrial; 15 Blepharisma Macronuclear; 16 Chlorophycean Mitochondrial; 21 Trematode Mitochondrial; 22 Scenedesmus obliquus; 23 Thraustochytrium Mitochondrial

getorf_find What to output? See the getorf manual (**0**)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.21 glimmer3

Predicts (prokaryotic) using glimmer3.

A.21.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (glimmer3)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

glimmer3 Glimmer3 is a open reading frame discovery program from the EMBOSS[?] package. It takes a set of input sequences and predicts all open reading frames. Additionally, this template converts the default output (predicted protein sequences) to GFF3.

A.21.2 Parameters

Required parameters

glimmer3_input_dir undefined

Optional parameters

glimmer3_gff_source source field to use in the gff. Defaults to glimmer3

glimmer3_input_extension input file extension. Defaults to 'fasta'

glimmer3_max_overlap Maximum overlap, see the glimmer documentation for the -o or —max_olap parameter

glimmer3_gene_len Minimum gene length (glimmer3 -g/—gene_len)

glimmer3_treshold treshold for calling a gene a gene (glimmer3 -t)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.22 gmap

A.22.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (gmap)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

gmap :

A.22.2 Parameters

Required parameters

gmap_db Gmap db

gmap_input_file input file with the sequences to map

Optional parameters

gmap_extra_parameters extra parameters to feed to gmap

gmap_invert_gff Invert the GFF (T/*F*)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.23 gmapdb

A.23.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (gmapdb)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

gmapdb :

A.23.2 Parameters

Required parameters

gmapdb_input_dir The reference sequence to build a gmap database with.

gmapdb_name Name of the gmap index to create

Optional parameters

gmapdb_input_extension Extension of the input files, defaults to 'fasta'

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.24 lftp

Use LFTP to download files. This template has two modi, one is set 'lftp_mode' to 'mirror' data, in which case both 'lftp_url' and 'lftp_pattern' (default *) are used. The other modus is 'lftp_mode=get', when one file defined by 'lftp_url' is downloaded. In the mirror mode it is possible to download only those files that are newer as the files already downloaded by using the 'lftp_timestamp' parameter

A.24.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (lftp)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

lftp execute the download

A.24.2 Parameters

Required parameters

lftp_url The base url to download from

Optional parameters

lftp_timestamp Depend on lftp to decide if a file needs updating, else a touchfile is created that you need to delete or touch before updating (T/F)

lftp_powerclean Do brute force cleaning (T/F). Remove all files, except moa.mk & Makefile when calling make clean. Defaults to F.

lftp_noclean set of files not to be deleted by the powerclean

lftp_pattern glob pattern to download

lftp_user username for the remote site

lftp_pass password for the remote site, note that this can be defined on the commandline using: 'make lftp_pass=PASSWORD'

lftp_output_dir subdir to create & write all output to. If not defined, data will be downloaded to directory containing the Makefile

lftp_dos2unix (T/F) Run dos2unix to prevent problems with possible dos text files (default=F).

lftp_mode Mode of operation - 'mirror' or 'get'. Mirror enables timestamping. Get just gets a single file. If using get, consider setting depend_lftp_timestamp to F. When using 'get', the full url should be in lftp_url. lftp_pattern is ignored. Defaults to mirror.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.25 mummer

A.25.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (mummer)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

mummer :

A.25.2 Parameters

Required parameters

mum_input_dir_a This set is compared to the sequences in input_dir_b. only a forward comparison is made (a against b, not the other way round)

mum_input_dir_b The set to compare against

Optional parameters

mum_input_extension undefined

mum_breaklen Set the distance an alignment extension will attempt to extend poor scoring regions before giving up (default 200)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.26 nstretch

A.26.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (nstretch)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

nstretch :

A.26.2 Parameters

Required parameters

Optional parameters

nstretch_input_dir input dir with the fasta files

nstretch_input_extension extension of the input files

nstretch_len minimal number of Ns before its reported (default 10)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.27 pregap

A.27.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (pregap)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

pregap Run pregap

A.27.2 Parameters

Required parameters

input_dir Directory with the input data

input_pattern file name pattern

cloning_vector File containing the cloning vector

sequencing_vector File containing the sequencing vector

ecoli_screenseq File containing ecoli screen sequences

repeat_masker_lib File with a repeatmasker library

vector_primerfile File with the vector primers

Optional parameters

quality_value_clip quality cutoff (default=10)

pregap_template the template pregap config file to use. if not defined, Moa tries ./files/pregap.config.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.28 traverse

Do nothing, except be a part in executing full directory structures

A.28.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (traverse)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

traverse Do nothing - no need to call this.

A.28.2 Parameters

Required parameters

Optional parameters

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.29 upload2gbrowse

A library that aids in uploading FASTA and GFF to a Generic Genome Browser database. This template is only to be used embedded in another template. This library expects that the following variables are preset; gup_fasta_dir, gup_gff_dir, gffsource gup_upload_fasta, gup_upload_gff

A.29.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (upload2gbrowse)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

upload2gbrowse :

initGbrowse Clean & initalize a gbrowse database. **Warning: all data will be lost!**

gupLock Prevent this job from uploading anything to the Generic Genome Browser database

gupUnlock Allow this job to upload to the Generic Genome Browser database

A.29.2 Parameters

Required parameters

gup_gffsource the gff source field, used in batch operations

gup_upload_fasta upload fasta to gbrowse (T/F)

gup_upload_gff upload gff to gbrowse (T/F)

gup_user gbrowse db user. If not defined, this defaults to 'moa'.

gup_db gbrowse database. If not defined, this defaults to 'moa'.

gup_gffsource the gff source field, used in batch operations

Optional parameters

gup_fasta_dir input directory with fasta files to upload to gbrowse

gup_gff_dir input directory with gff files to upload to gbrowse

gup_gff_extension extension of the GFF files to upload (.gff)

gup_fasta_extension extension of the FASTA files to upload (.fasta)

gup_upload_limit Do not upload more that this number of files (infinite)

gup_upload_fasta upload fasta to gbrowse (T/F)

gup_upload_gff upload gff to gbrowse (T/F)

gup_force_upload upload to gbrowse, ignore gup_lock and upload all, not only files newer that upload_gff or upload_fasta

marks_extensions Add some extensions to the Gbrowse database to be initalized, for use by Mark.

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.30 varscan

A.30.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (varscan)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

varscan :

A.30.2 Parameters

Required parameters

varscan_input_file Varscan input alignments file

Optional parameters

varscan_extra_params location of varscan.pl, defaults to '/usr/lib/perl5/site_perl/5.8.8/varscan.pl'

varscan_output_name Base name of the output files

varscan_perl_file undefined

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.31 vmatch

A.31.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (vmatch)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

vmatch :

A.31.2 Parameters

Required parameters

vmatch_db vmatch db to compare against

vmatch_input_file input file with the sequences to map

Optional parameters

vmatch_extra_parameters extra parameters to feed to vmatch

vmatch_invert_gff Invert the GFF (T/*F*)

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

A.32 vmatchdb

Builds a vmatchdb index from a sequence

A.32.1 Targets

(empty) Leaving the target unspecified executes the default target(s): (vmatchdb)

clean removes all results from this job

all executes the default target and into subdirectories to execute any other moa makefile it encounters

vmatchdb :

A.32.2 Parameters

Required parameters

vmatchdb_input_dir The sequence to build a vmatch database from.

vmatchdb_name Name of the vmatch index to create

Optional parameters

vmatchdb_input_extension Extension of the input files, defaults to 'fasta'

vmatch_pl prefix length

jid Unique identifier for this job. Jids are autogenerated if undefined. A descriptive jid is important, particularly if you are using couchdb.

project A project name; group your analyses.

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, Oct 1990.

[2] Apache couchdb. http://couchdb.apache.org/.

[3] Biopython. http://biopython.org/.

[4] Generic Feature Format (v3). http://song.sourceforge.net/gff3.shtml.

[5] Gnu Make. http://www.gnu.org/software/make/.

[6] Red Hat Enterprise Linux. http://www.redhat.com/rhel/.

[7] P. Rice, I. Longden, and A. Bleasby. Emboss: the european molecular biology open software suite. *Trends Genet*, 16(6):276–277, Jun 2000.

[8] Lincoln D Stein, Christopher Mungall, ShengQiang Shu, Michael Caudy, Marco Mangone, Allen Day, Elizabeth Nickerson, Jason E Stajich, Todd W Harris, Adrian Arva, and Suzanna Lewis. The generic genome browser: a building block for a model organism system database. *Genome Res*, 12(10): 1599–1610, Oct 2002.

[9] Ubuntu. http://www.ubuntu.com/.