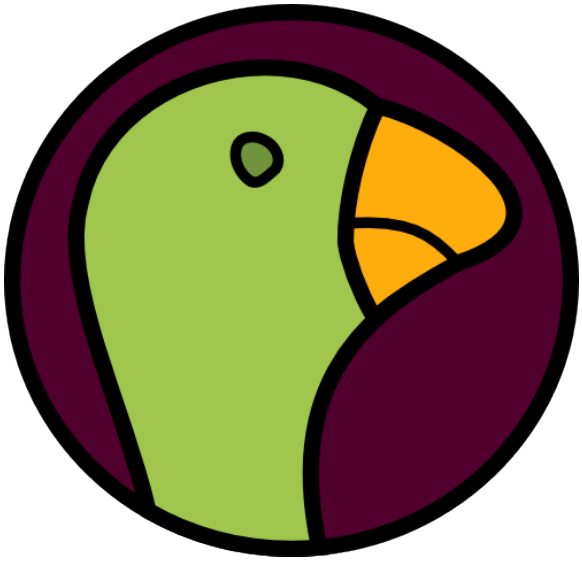


Plant & Food RESEARCH
RANGAHAU AHUMĀRA KAI

Moa

author: Mark Fiers

date: Mon Apr 5 22:02:02 NZST 2010

git: master 61c5ef6f7d530e1ee2b55c83025eb0cca2f21f4f

Contents

Chapter 1

Introduction

NOTE: both the software and the manual are under development. Expect things to change.

A bioinformatics project is often build up of many separate, interdependent, steps. Once projects get bigger it becomes necessary to organize and automate such a project. There is number of different ways to automate a bioinformatics project[3,5,6,8]. Many bioinformaticians, however, continue writing tailor-made scripts to organize and automate their work. Scripting has some obvious advantages, of which flexibility is probably most important. A downside to scripting is that projects usually do not scale very well and easily become cluttered.

Moa is aimed at organizing and automating a bioinformatics project without losing the flexibility of tailor made scripts. The best way to understand what Moa aims to achieve is by example:

```
moa new blast -t "run a demo"
moa set blast_db=/data/blast/db/nt
moa set blast_input_dir=../sequences/
moa set moa_postprocess='grep polymerase gff/*gff > pol.gff'
moa run
```

In the first line, a new BLAST[1] job (titled "run demo") is created in the current directory. What really happens is that Moa creates a small script (a Makefile to be precise) that imports a BLAST specific template. The next two lines set a few parameters for BLAST. In the last line Moa is executed and 'blasts' the input sequences (in `../sequences`) against the database in `/data/blast/db/nt`. BLAST output files (XML) are generated and converted to GFF (GFF conversion is an extra, not part of the BLAST suite). The last line is probably the most special; it is a single (shell) commmand that will be executed after BLAST is executed (there is also a corresponding `moa_preprocess`). In this case all BLAST hits that have the word "polymerase" in their description are gathered

into a single GFF file.

Moa employs GNU make to create and use building blocks in bioinformatics data analysis. GNU Make is originally developed to aid in compiling software. Compilation usually involves the execution of many interdependent compilation and linker steps. GNU make is able to compile large software projects with tens of thousands of source files based on a Makefile that describes how a target file is to be created. GNU Make is flexible enough to be used with practically any programming language. Moreover, GNU Make can be used to automate any series of commands as long as they can be executed from the command line. It is therefore not only possible, but an excellent idea (not mine), to use Gnu Make in bioinformatics projects (see biowiki, nodalpoint or biomake)

Moa aims to do the following things:

- *Organize a project*: Each Moa job must be located in its own directory. It is possible to automatically execute a directory tree of Moa jobs. Proper use of these features will result in a logical project structure.
- *Reuse building blocks*: Moa templates are GNU Makefiles that follow a set of conventions. It is easy to implement new building blocks. (see chapter X).
- *Document*: It is possible to add meta-data such as a title and description to each Moa job, making it easy to
- *Provide a uniform interface*: Moa allows you to operate your project almost exclusively using a single command (conveniently called `moa`).

1.0.1 Example session

The best way to understand how to use Moa is a sample session.

We'll start by creating directories to hold the data and analysis structure:

```
mkdir introduction
cd introduction
```

We've created a directory `introduction` for the tutorial. Within this directory we'll organize the components of our analysis. We want to initialize this directory so that it becomes a part of this Moa pipeline. This is useful later, if we want to run all analysis at once. To do this, run:

```
moa new project -t 'Introduction'
```

The `moa new` command is used to create new moa jobs. In this case we create a job with the template "project". In itself this template does not do anything

but serves to group new projects. The `-t` parameters assigns a title to this Moa job. We will now create a new directory to hold the first step of the pipeline:

```
mkdir 10.download
cd 10.download
moa new
```

Moa does not dictate a directory structure for your analysis pipeline, but to make full usage of Moa it is advisable to create a logical organization. Two important features of Moa that relate to this are:

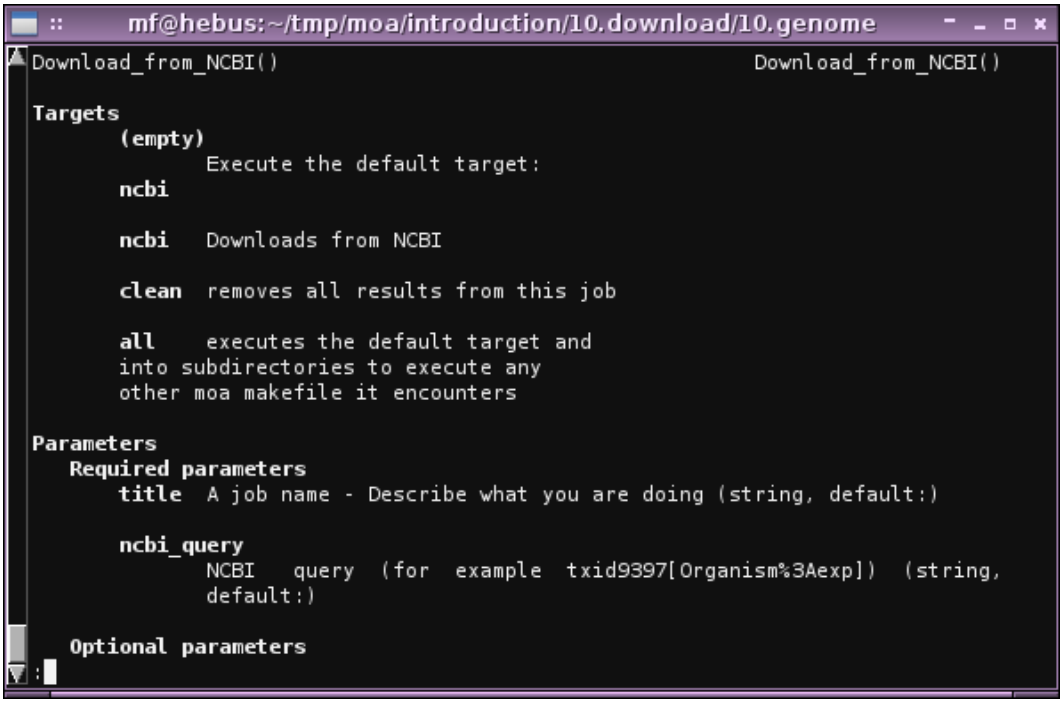
- Each moa job is contained in one directory. Output files of a job are typically stored in that directory. It is not possible to have more than one Moa jobs in a directory.
- Moa is able to automatically execute all jobs in a directory and the underlying sub-directories (using `moa all`).

If properly used, these two features force a logical, modular, project structure. To assign an order to the steps inside a directory it is possible to prepend a number to the directory name (i.e. "10."). Note that Moa sorts directories alphabetically and not numerically

We will now created a new folder to hold a genome sequence we are about to download and set up the Moa job to actually do the download.

```
mkdir 10.genome
cd 10.genome
moa new -t 'download a potato BAC' ncbi
```

Here we create a Moa job to download a sequence from NCBI by using the "ncbi" template. Once a Moa job is instantiated you can run `moa help` to get some information on how to use this template



Note that if you want help on how to use the moa itself, you should use `moa --help`

Before you can get the data from NCBI, you will have to tell Moa what you want to download. This is easy if you know the Genbank accession numbers. In this case we'll download the nucleotide sequence (from the database "nuccore") with the accession id AC237669.1

```
moa set ncbi_db=nuccore
moa set ncbi_query=AC237669.1
```

You can check if the parameters are set correctly by running `moa show`. This should come back with the following text: title download a potato BAC ncbi_query AC237669.1 ncbi_db nuccore ncbi_sequence_name

If everything seems fine, you can run this job:

```
moa
```

Or, you could also have used `moa run`. It is possible that you get an error message notifying that "wget" or "xml_grep" cannot be found. Most, properly written, Moa templates do prerequisite checking if necessary. If either of these tools is missing, you will need to install them first (possibly by running `sudo apt get install wget` or `sudo apt-get install xml-twig-tools`, or ask your system administrator)

If Moa runs, quite a lot of output is generated. If things go wrong, there is probably a clue to why it did not work in this output. If the Moa job is successful,

the last line should be "Moa finished - Succes!". If you do an `ls` you now see a `fasta` directory with one fasta file. This fasta file contains the downloaded genome.

Now we can start doing things with the downloaded sequence. To see what other templates are available, try `moa list`.

Chapter 2

Installation

2.1 Prerequisites

Moa is developed and tested on Ubuntu, RHEL and Archlinux and is expected to operate without much problems on all modern Linux distributions. Moa has the following prerequisites (and a large number more for all templates). The version numbers are an indication, not strict prerequisites. Other, even older, versions might work.

- Gnu Make (3.81)
- Git (1.6). Used to download the latest version, and possibly by the git plugin.
- Python (2.6). Python version 2.5 and lower or 3.0 will not work.
- Bash (4.1.2). 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 distributed together with Moa.
- Pandoc (Preferably 1.5). Pandoc is used in generating the on the fly help for which the ancient 0.46, bundled with Ubuntu, will also work. Pandoc is also used to generating LaTeX & HTML documentation. For these a recent version is strongly recommended.

2.1.1 Bioinformatics tools

Each of the wrapped tools, obviously, requires that these tools are present. Usually, unless mentioned otherwise, Moa expects all tools to be present on the

system PATH. Most templates check for prerequisite tools to be present and will generate a clear error if this is not the case.

2.2 Installation from source

2.2.1 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 assume installation in the latter location.

2.2.2 Downloading Moa

Moa is hosted at github:

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

Currently there are no stable releases so the only option is to download the latest version of the software using Git. Git makes it easy to stay up to date with the latest version and, even better, allows anybody to submit code to the Moa repository. To download Moa using Git (assuming installation in `~/moa`)

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

2.2.3 Downloading a tarball

As an alternative, it is possible to download an (automatically generated) archive of the latest Moa version from: http://github.com/mfiers/Moa/tarball/master, for example with the following command:

`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-077a672.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-077a672.tar.gz
mv mfiers-Moa-077a672 ~/moa
```

After following either procdure you should have a directory with the Moa source. It should, amongst others contain a `~/moa/bin` folder.

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. Moa has a script that does this for you:

```
. ~/project/moa/bin/moainit.sh
```

(Note the dot!)

You can also add this line to your `~/.bashrc` and run `. ~/.bashrc`.

If you are installing Moa to be used by all users of your system system, please remember to set the attributes:

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

Moa should now work.

Chapter 3

Extending Moa

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

3.1 Creating a simple template - example

A template is, as stated, basically a Makefile that adheres to a set of 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 two parts:

- Definitions
- Implementation

This order is very important! Parts of the Moa core are included inbetween the definitions and the implementation. Getting the order wrong might cripple your template.

In the remainder of this chapter we will describe how to implement a new tool base on a simple example that creates the reverse complement of a FASTA sequence using the EMBOSS[7] revseq utility.

All templates are stored in the `$MOABASE/template` directory. This template will be stored under the name `revseq.mk`.

3.1.1 Definitions

Each template start with including the first part of the Moa core:

```
include $(MOABASE)/template/moa/prepare.mk
```

`moaBasePre` defines a set of default Moa variables and has some macro's that make variable definition easier. The template defintion continues by defining a set of variables used in the latter part of the template.

3.1.2 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	A short title for this template
moa_description	A short description of this template

For our Revseq example:

```
moa_title = Revseq
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 next line ends with a backslash. No spaces are allowd after the backslash. Indenting the next line is not necessary, but it does enhance readability.

namespace definition: `moa_ids`

Each MOA template defines a set of variables and targets (things to do). These variables and targets (usually) share a single id in their name. In the case of template variables this is a convention, making templates easier to read. Targets, however, are often executed automatically, based on the expected name. A `moa_id` is defined in the following way (for our example):

```
moa_ids += revseq
```

A `moa_id` should be unique and is ideally short and consise. It is also advisable to save the template under the same name.

job specific variables

The next part of the definitions can be used freely to define variables to be used in the implementation. Each of these variables, ideally, start with the `moa_id` (but this is not enforced). The advantage of defining variables here is that they will be accessible via the command line and the API. For example, you can set a define variable using the following command line:

```
moa set title='Descriptive title'
```

There are two types of variables that can be defined: mandatory and optional.

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

Most templates will have variables specific to the job. These can be defined by

optional job specific variables

3.1.3 Include the moa core library

To include the core moa library, the following line needs to be added to the Makefile

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

3.1.4 Implementation

3.1.5 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

3.2 Reference

3.2.1 Moa Makefile load order

Makefiles are sensitive to the order in which definitions are made, and thus the order in which the include files are loaded. Moa broadly recognizes two stages: "definition" and "implementation". The implementation phase starts once the moa core library is loaded.

Moa makefile load starts with loading the template makefile in the current work directory. This Makefile loads a number of other makefiles that load more Makefiles. The following list shows a detailed load order

- **Makefile**: The Makefile in the working directory
  - **prepare.mk**: initial definitions. At the start of the prepare.mk file the following files are loaded:
    - **gmsl**: The GNU Make Standard library, a number of utilities for use in Makefiles.

    * **global configuration** (`$(MOABASE)/etc/moa.conf`): This file loads the global default configuration file (`$(MOABASE)/etc/moa.conf.mk.default`)
    * **Project configuration**: (if present). Moa attempts to find this in the first parent directory of the current working directory that contains a moa project with template "project".
    * **Local configuration** (`moa.mk`)
    * **Plugin definitions**: For each plugin name defined in the variable `moa_plugins`, moa attempts to load a file called `$(MOABASE)/template/moa/plugins/P`

  Once these files are loaded, more Moa specific definitions follow in `prepare.mk`

- **template Makefile**: (`$(MOABASE)/template/TEMPLATENAME.mk`) A makefile specific for the job at hand. This template Makefile might attempt to load prepare.mk, unless it was already loaded earlier. The first part of the template Makefile is used for defining template specific variables.

  The definition phase of a Moa Makefile is concluded by loading:

  * **Moa core** (`$(MOABASE)/template/moa/core.mk`). The first thing the Moa core libraries do is loading a set of plugins:
    - **Plugin cores**: (`$(MOABASE)/template/moa/plugins/PLUGINNAME.mk`)

  After the plugins are loaded moa defines a number of core targets, most importantly, the default target that defines the execution order (see the next paragraph). As much of the functionality as possible is definined as a plubin.

Once the core library has loaded, the template specific targets are parsed.

3.2.2 Execution order

Run

- `moa_hooks_prewelcome`
- `moa_welcome`
- `moa_hooks_precheck`
- `moa_check`
- `moa_prepare`
- `$(moa_id)_prepare`

- `$(moa_id)`
- `$(moa_id)_post`
- moa_post

3.2.3 Environment variables

These environment variables are used by Moa:

`MOAANSI` The default is to use (ANSI) colored characters in the output. To prevent this from happening, set this (environment) variable to `no`.

`MOAPROJECTROOT` The root of a moa project - project root is a parent directory of the current directoy that has a moa job with template `project`. If there is no project root, this variable is undefined.

3.2.4 Global functions

These function are meant to be used at the top level of a Makefile (meaning, not inside a target command block). Function can be called using:

`$(call FUNCTIONNAME,ARGUMENT1,ARGUMENT2,...)`

`$(call moa_fileset_define,ID,EXTENSION,HELP)` Define a set of files to be recoginized by Moa.

`$(call moa_fileset_remap,INPUT_ID,OUTPUT_ID,OUTPUT_EXTENSION)` Remap a set of input files to ....

`$(call moa_fileset_remap_nodir,INPUT_ID,OUTPUT_ID,OUTPUT_EXTENSION)` as `moa_fileset_remap`, but without prefixing the set with a subdirectory

3.2.5 Command functions

The following commands render a command that can be executed inside a target command block

'$(call echo,TEXT) Returns an echo statement for the text with a green block prepended. The color allows for easy recognition of the echo'd statements. Note that these only work within the code block of a target.

`$(call errr,TEXT)` as `$(call echo,TEXT)`, but with a red marker (error)

`$(call exer,TEXT)` as `$(call errr,TEXT)`, but exits the Makefile with an error

`$(call exerUnlock,TEXT)` as `$(call exer,TEXT)`, but remove the Moa lockfile

`$(call warn,TEXT)` as `$(call echo,TEXT)`, but with a yellow marker

3.2.6 Variables

$(comma) a comma

$e Can be used in place of Makefile "@". A @ prepended to a command inside a target in a Makefile supresses echoing of that line during execution. If $e is used, then supression is depending on executing moa with the -v (verbose) parameter.

$(empty) empty

$(parC) parentheses close

$(parO) parentheses open

$(sep) contains the pipe symbol "|"

$(space) a single space

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 act_crunch

A.1.1 Targets

(empty) Execute the default target:

crunch

crunch generate a list of bidirectional best blast hits.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.1.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

crunch_input_fila_a First multifasta input file (file, default:)

crunch_input_fila_b First multifasta input file (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

crunch_protein Are we looking at proteins? (T|*F*)

crunch_eval e value cutoff (float, default:1e–10)

crunch_nothreads threads to run crunch with (note the overlap with the Make -j parameter) (integer, default:4)

A.2 adhoc

Run a specified oneliner or script on a set of inputfiles

A.2.1 Targets

(empty) Execute the default target:

adhoc

adhoc adhoc files

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.2.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

Optional parameters

adhoc_name_sed SED expression to be executed on each file name - allows you to change file names (string, default:s/a/a/)

adhoc_output_dir Output subdirectory, defaults to '.' (directory, default:.)

adhoc_parallel allow parallel execution. If, for example, concatenating to one single file, you should not have multiple threads. (T|*F*)

adhoc_process Command to process the files. If undefined, hardlink the files. (string, default:ln -f $< $t)

adhoc_limit limit the number of files adhoced (with the most recent files first, defaults to 1mln) (integer, default:1000000)

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

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.3 bartab

A.3.1 Targets

(empty) Execute the default target:

bartab

bartab .. to be written ..

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.3.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

bartab_in input file for bartab (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

bartab_qin Quality scores for the input fasta file (file, default:)

bartab_map A file mapping barcodes to metadata (file, default:)

bartab_out base output name (integer, default:bartab)

bartab_forward_primer remove forward primer (string, default:)

bartab_reverse_primer remove reverse primer (string, default:)

bartab_min_length minimun acceptable sequence length (integer, default:50)

bartab_trim Trim barcode (*T*|F)

bartab_extra_parameters extra parameters to feed bartab (string, default:)

A.4 bidibebla

A.4.1 Targets

(empty) Execute the default target:

bdbb

bdbb generate a list of bidirectional best blast hits.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.4.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

bdbb_input_file_a First multifasta input file (file, default:)

bdbb_input_file_b First multifasta input file (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

bdbb_protein Are we looking at proteins? (T|F)

bdbb_eval e value cutoff (float, default:1e–10)

bdbb_nothreads threads to run bdbb with (note the overlap with the Make -j parameter) (integer, default:4)

A.5 blast

A.5.1 Targets

(empty) Execute the default target:

blast

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[2]). Additionally, a simple text report is created. moa_additional_targets += blast_report

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.5.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

blast_input_dir Directory with the BLAST input files (directory, default:)

blast_db Location of the blast database. You can either define the blast db parameter as used by blast, or any of the blast database files, in which case the extension will be removed before use (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

blast_input_extension file extension for the files in blast_input_dir (string, default:fasta)

blast_input_glob glob to select a subset of files from blast_input_dir (string, default:*)

blast_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

blast_input_limit Number of files to use, if not defined: all files (integer, default:)

blast_gff_source source field to use in the gff (string, default:BLAST)

blast_program blast program to use (default: blastn) (blastx|*blastn*|blastp|t*blastn*|tblastx)

blast_eval e value cutoff (float, default:1e–10)

blast_nohits number of hits to report (integer, default:50)

blast_nothreads threads to run blast with (note the overlap with the Make -j parameter) (integer, default:2)

blast_gff_blasthit (T,**F**) - export an extra blasthit feature to the created gff, grouping all hsp (match) features. (T|*F*)

A.6 blastSingle

A.6.1 Targets

(empty) Execute the default target:

blast

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[2]). Additionally, a simple text report is created. moa_additional_targets += blast_report

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.6.2 Parameters

Required parameters

blast_input_file Input fasta file to BLAST (file, default:)

title A job name - Describe what you are doing (string, default:)

blast_input_dir Directory with the BLAST input files (directory, default:)

blast_db Location of the blast database. You can either define the blast db parameter as used by blast, or any of the blast database files, in which case the extension will be removed before use (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

blast_input_extension file extension for the files in blast_input_dir (string, default:fasta)

blast_input_glob glob to select a subset of files from blast_input_dir (string, default:*)

blast_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

blast_input_limit Number of files to use, if not defined: all files (integer, default:)

blast_gff_source source field to use in the gff (string, default:BLAST)

blast_program blast program to use (default: blastn) (blastx|*blastn*|blastp|t*blastn*|tblastx)

blast_eval e value cutoff (float, default:1e–10)

blast_nohits number of hits to report (integer, default:50)

blast_nothreads threads to run blast with (note the overlap with the Make -j parameter) (integer, default:2)

blast_gff_blasthit (T,**F**) - export an extra blasthit feature to the created gff, grouping all hsp (match) features. (T|*F*)

A.7 blastdb

A.7.1 Targets

(empty) Execute the default target:

blastdb

blastdb Takes either a set of fasta files or a single multi-fasta input file and creates a BLAST database.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.7.2 Parameters

Required parameters

bdb_name Database name to create. (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

bdb_input_dir Dir with the input fasta files, defaults to ./fasta (directory, default:)

bdb_input_extension extension of the input sequence files, defaults to fasta (string, default:fasta)

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

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

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.8 blat

Run BLAT on an set of input files (query) vs a database.

A.8.1 Targets

(empty) Execute the default target:

blat

blat : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.8.2 Parameters

Required parameters

blat_db type of the database (dna, prot or dnax) (['dna',|'prot',|'dnax']**)

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 (file, default:)

blat_input_dir source field in the generated gff (directory, default:)

blat_gff_source Source field for the generated GFF files (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

blat_input_extension extension of the input files (string, default:fasta)

blat_eval evalue cutoff to select the reported hits on (defaults to 1e–15) (float, default:1e–10)

blat_db_id_list a sorted list of db ids and descriptions, enhances the report generated (file, default:)

blat_db_type type of the database (dna, prot or dnax) (['*dna*',|'prot',|'*dnax*'])

blat_query_type type of the query (dna, rna, prot, dnax or rnax) (['*dna*',|'rna',|'prot',|'*dnax*',|'rnax'])

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.9 bowtie

Run BOWTIE on an set of input files (query) vs a database index.

A.9.1 Targets

(empty) Execute the default target:

bowtie

bowtie : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.9.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

bowtie_db The bowtie database to use. It is allowed to define one of the bowtie database files (.[0–9].ebwt). (file, default:)

bowtie_input_dir Input files for bowtie (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

bowtie_input_extension file extension for the files in bowtie_input_dir (string, default:fastq)

bowtie_input_glob glob to select a subset of files from bowtie_input_dir (string, default:*)

bowtie_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

bowtie_input_limit Number of files to use, if not defined: all files (integer, default:)

bowtie_input_format Format of the input files (*fastq*|fasta)

bowtie_extra_params extra parameters to feed bowtie (string, default:)

bowtie_paired_ends perform a paired end analysis. If so, the input files are expected to be of the form '*_1.fastq' and ' *_2.fastq' (T|*F*)

bowtie_forward_suffix Last part of the sequence name identifying a file with forward reads (string, default:_1)

bowtie_reverse_suffix Last part of the sequence name identifying a file with reverse reads (string, default:_2)

bowtie_output_format Format of the output file (bowtie|*bam*|sam)

bowtie_insertsize Expected insertsize (float, default:5000)

bowtie_insertsize_sed SED expression to filter the expected insertsize from the input file name (string, default:)

bowtie_insertsize_min multiplier determining the minimal acceptable value for two paired reads to be apart. If the bowtie_insertsize is 10000 and this parameter is set at 0.8, than reads that are closer together than 8000 nt are rejecte (float, default:0.1)

bowtie_insertsize_max Max insertsize for a paired alignment (float, default:10)

A.10 bowtiedb

Builds a bowtie index from a reference sequence

A.10.1 Targets

(empty) Execute the default target:

bowtiedb

bowtiedb : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.10.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

bowtiedb_input_dir Sequence files used to build a bowtie database (directory, default:)

bowtiedb_name Name of the bowtie index to create (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

bowtiedb_input_extension file extension for the files in bowtiedb_input_dir (string, default:fasta)

bowtiedb_input_glob glob to select a subset of files from bowtiedb_input_dir (string, default:*)

bowtiedb_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

bowtiedb_input_limit Number of files to use, if not defined: all files (integer, default:)

A.11 cleanFasta

A.11.1 Targets

(empty) Execute the default target:

clean_fasta

clean_fasta Cleanup of a FASTA file (in place!)

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.11.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

cf_input_dir Directory with the sequences to run cleanfasta on (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

cf_input_extension input file extension (string, default:fasta)

sed_command The sed command cleaning the code, defaults to '/>/!s/[ACGTNacgtn]/N/g' (string, default:/>/!s/[ACGTNacgtn]/N/g)

A.12 clustalgroup

A.12.1 Targets

(empty) Execute the default target:

clustalgroup

clustalgroup run clustalw

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.12.2 Parameters

Required parameters

cwg_input_dir This set of sequences to run clustalw on (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

cwg_input_extension Input file extension (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.13 clustalpair

A.13.1 Targets

(empty) Execute the default target:

clustalpair

clustalpair run clustalw

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.13.2 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) (directory, default:)

input_dir_b The set to compare against (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

input_extension Extension of the input files (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.14 clustalw

A.14.1 Targets

(empty) Execute the default target:

clustalw

clustalw run clustalw

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.14.2 Parameters

Required parameters

input_dir_a This set is compared to the sequences in input_dir_b. (directory, default:)

input_dir_b The set to compare against. Only a forward comparison is made (a against b, not the other way round) (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

input_extension Extension of the input files (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.15 concatenate

Concatenate a set of fasta files into one.

A.15.1 Targets

(empty) Execute the default target:

concatenate

concatenate Concatenate a set of FASTA files

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.15.2 Parameters

Required parameters

input_dir Directory with the input data (directory, default:)

name name of the file, the outputfile will become ./name.fasta (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

input_extension Extension of the input files (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.16 create.gbrowse.db

A.16.1 Targets

(empty) Execute the default target:

upload2gbrowse

upload2gbrowse : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters initGbrowse : Clean & initalize a gbrowse database. **Warning: all data will be lost!** gupgo : Actually do the upload. upload2gbrowse NEVER does this automatically!

A.16.2 Parameters

Required parameters

gup_user gbrowse db user. If not defined, this defaults to 'moa'. (string, default:)

gup_db gbrowse database. If not defined, this defaults to 'moa'. (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

gup_gff_extension extension of the GFF files to upload (.gff) (string, default:gff)

gup_fasta_extension extension of the FASTA files to upload (.fasta) (string, default:fasta)

gup_upload_fasta upload fasta to gbrowse (T/F) (T|*F*)

gup_upload_gff upload gff to gbrowse (T/F) (T|*F*)

gup_force_upload upload to gbrowse, ignore gup_lock and upload all, not only files newer that upload_gff or upload_fasta (T|*F*)

marks_extensions Add some extensions to the Gbrowse database to be initalized, for use by Mark. (T|*F*)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.17 crunch

A.17.1 Targets

(empty) Execute the default target:

crunch

crunch create crunch files

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.17.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

crunch_input_dir Directory with input fasta files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

crunch_input_extension file extension for the files in crunch_input_dir (string, default:fasta)

crunch_input_glob glob to select a subset of files from crunch_input_dir (string, default:*)

crunch_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

crunch_input_limit Number of files to use, if not defined: all files (integer, default:)

crunch_eval e value cutoff (float, default:1e–10)

crunch_nothreads threads to run crunch with (note the overlap with the Make -j parameter) (integer, default:4)

A.18 dottup

Use dottup (from EMBOSS) to compare two sets of sequences

A.18.1 Targets

(empty) Execute the default target:

dottup

dottup Run dottup

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.18.2 Parameters

Required parameters

dottup_input_dir_a This set is compared to the sequences in input_dir_b. (directory, default:)

dottup_input_dir_b The set to compare against (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

dottup_input_extension Extension of the dottup input files (string, default:fasta)

dottup_wordsize Wordsize used to discover similarities between sequences (integer, default:8)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.19 dottupSelf

A.19.1 Targets

(empty) Execute the default target:

dotself

dotself run clustalw

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.19.2 Parameters

Required parameters

dotself_input_dir Set of sequences to use (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

dotself_input_extension Extension of input files (string, default:fasta)

dotself_wordsize Wordsize used for recognizing similarity (integer, default:6)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.20 empty

A.20.1 Targets

(empty) Execute the default target:

empty

empty : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.20.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.21 fasta2gff

Create gff from a fasta file to accompany upload to a gbrowse db

A.21.1 Targets

(empty) Execute the default target:

fasta2gff

fasta2gff : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.21.2 Parameters

Required parameters

f2g_gffsource Source to be used in the gff (string, default:)

f2g_input_dir Directory with the input fasta files (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

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

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

f2g_options options to be passed to the fasta2gff script (string, default:)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.22 gap4export

Export data from an assembly using gap4

A.22.1 Targets

(empty) Execute the default target:

gap4export

gap4export Export data from an assembly using gap4

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.22.2 Parameters

Required parameters

ge_input_dir Directory with the input data (directory, default:)

ge_input_pattern file name pattern (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.23 gather

gather a set of files and create hardlinks to. Hardlinks have as advantage that updates are noticed via the timestamp. Hence, make recognizes them.

A.23.1 Targets

(empty) Execute the default target:

gather

gather gather files

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.23.2 Parameters

Required parameters

g_input_dir list of directories with the input files (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

g_input_pattern glob pattern to download (string, default:*)

g_name_sed SED expression to be executed on each file name - allows you to change file names (string, default:s/a/a/)

g_output_dir Output subdirectory, defaults to '.' (directory, default:.)

g_parallel allow parallel execution (T) or not (**F**). If for example concatenating to one single file, you should not have multiple threads. (T|*F*)

g_process Command to process the files. If undefined, hardlink the files. (string, default:ln -f 9084< 9084(g_target))

g_limit limit the number of files gathered (with the most recent files first, defaults to 1mln) (integer, default:1000000)

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

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.24 genemarks

A.24.1 Targets

(empty) Execute the default target:

genemarks

genemarks : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.24.2 Parameters

Required parameters

genemarks_input_dir directory containing the input sequences (directory, default:)

genemarks_matrix the matrix to use (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

genemarks_gff_source source field to use in the gff. Defaults to geneMarkS (string, default:genemarks)

genemarks_input_extension input file extension. Defaults to 'fasta' (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.25 getorf

A.25.1 Targets

(empty) Execute the default target:

getorf

getorf : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.25.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

getorf_input_dir Input files for getorf (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

getorf_input_extension file extension for the files in getorf_input_dir (string, default:fasta)

getorf_input_glob glob to select a subset of files from getorf_input_dir (string, default:*)

getorf_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

getorf_input_limit Number of files to use, if not defined: all files (integer, default:)

getorf_gff_source source field to use in the gff. (string, default:getorf)

getorf_minsize minimal nucleotide size of the predicted ORF. (integer, default:30)

getorf_maxsize maximal nucleotide size of the predicted ORF. (integer, default: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. (0|1|2|3|4|5|6|7|8|9|10|*11*|12|13|14|15|16|21|22|23)

getorf_find What to output? 0: Translation between stop codons, 1: Translation between start & stop codon, 2: Nucleotide sequence between stop codons; 3: Nucleotide sequence between start and stop codons. Default: 3 (0|1|2|*3*)

A.26 glimmer3

A.26.1 Targets

(empty) Execute the default target:

glimmer3

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.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.26.2 Parameters

Required parameters

glimmer3_input_dir Input directory with the sequences to run glimmer3 on (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

glimmer3_gff_source source field to use in the gff. Defaults to glimmer3 (string, default:glimmer3)

glimmer3_input_extension input file extension. Defaults to 'fasta' (string, default:fasta)

glimmer3_max_overlap Maximum overlap, see the glimmer documentation for the -o or —max_olap parameter (integer, default:50)

glimmer3_gene_len Minimum gene length (glimmer3 -g/—gene_len) (integer, default:110)

glimmer3_treshold treshold for calling a gene a gene (glimmer3 -t) (integer, default:30)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.27 gmap

Run GMAP on an set of input files (query) vs a database index.

A.27.1 Targets

(empty) Execute the default target:

gmap

gmap : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.27.2 Parameters

Required parameters

gmap_db Gmap db (file, default:)

gmap_input_file input file with the sequences to map (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

gmap_extra_parameters extra parameters to feed to gmap (string, default:)

gmap_invert_gff Invert the GFF (T/F) (T|F)

gmap_gff_source Source field to use in the output GFF (string, default:gmap)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.28 gmapdb

Builds gmapdb index from a reference sequence

A.28.1 Targets

(empty) Execute the default target:

gmapdb

gmapdb : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.28.2 Parameters

Required parameters

gmapdb_input_dir The reference sequence to build a gmap database with. (directory, default:)

gmapdb_name Name of the gmap index to create (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

gmapdb_input_extension Extension of the input files, defaults to 'fasta' (string, default:fasta)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.29 gsMapper

Run the Roche GS Reference mapper

A.29.1 Targets

(empty) Execute the default target:

gsmap

gsmap : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.29.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

gsmap_sfffile SFF files with reads to map against the reference sequences (file, default:)

gsmap_name Name identifying this mapping in the output gff (string, default:)

gsmap_reference_fasta A multifasta file with the reference sequence(s)with the library id. (file, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

gsmap_min_overlap_len Minimum overlap length in the assembly step (integer, default:40)

gsmap_min_overlap_ident Minimum identity length in the assembly step (integer, default:90)

gsmap_annotation Gene annotation file in the UCSC GenePred format (file, default:)

A.30 h_blast

A.30.1 Targets

(empty) Execute the default target:

h_blast

h_blast Similar to a normal blast, but now running on an hadoop cluster

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.30.2 Parameters

Required parameters

hadoop_base location of the hadoop installation (directory, default:)

h_blast_input_dir location of the hadoop installation (directory, default:)

h_blast_db Location of the blast database (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

hdfs_base htfs://SERVER:PORT for the hdfs filesystem, defaults to hdfs://localhost:9000 (string, default:hdfs://localhost:9000)

h_blast_input_extension input file extension (string, default:fasta)

h_blast_program blast program to use (default: blastn) (*blastn*|blastp|blastx|t*blastn*|tblastx)

h_blast_eval e value cutoff (float, default:1e–10)

h_blast_nohits number of hits to report (integer, default:50)

h_blast_nothreads threads to run blast with (note the overlap with the Make -j parameter) (integer, default:1)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.31 lftp

A.31.1 Targets

(empty) Execute the default target:

lftp

lftp execute the download

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.31.2 Parameters

Required parameters

lftp_url The base url to download from (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

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

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

lftp_noclean set of files not to be deleted by the powerclean (string, default:moa.mk Makefile)

lftp_pattern glob pattern to download (string, default:'*')

lftp_lock Lock this job after running. This means that you will have to manually unlock the job before lftp actually reruns. This is a good choice if your downloading large datasets or have a slow connection (*T*|F)

lftp_user username for the remote site (string, default:)

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

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

lftp_dos2unix Run dos2unix to prevent problems with possible dos text files (T|*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. (mirror|*get*)

lftp_get_name target name of the file to download (string, default:)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.32 maq_fasta2bfa

A.32.1 Targets

(empty) Execute the default target:

f2b

f2b : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.32.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

f2b_input_dir input FASTA files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

f2b_input_extension file extension for the files in f2b_input_dir (string, default:fasta)

f2b_input_glob glob to select a subset of files from f2b_input_dir (string, default:*)

f2b_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

f2b_input_limit Number of files to use, if not defined: all files (integer, default:)

A.33 maq_fastq2bfq

A.33.1 Targets

(empty) Execute the default target:

fq2bq

fq2bq : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.33.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

fq2bq_input_dir input FASTA files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

fq2bq_input_extension file extension for the files in fq2bq_input_dir (string, default:fastq)

fq2bq_input_glob glob to select a subset of files from fq2bq_input_dir (string, default:*)

fq2bq_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

fq2bq_input_limit Number of files to use, if not defined: all files (integer, default:)

A.34 maq_match_pair

A.34.1 Targets

(empty) Execute the default target:

maqpair

maqpair : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.34.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

maqpair_read_dir directory containing the forward reads (string, default:)

maqpair_forward_suffix Suffix of each forward filename - recognize forward files this way. Note this is not a regular extension, no '.' is assumed between the filename & suffix (string, default:_f.bfq)

maqpair_reference Reference bfa file to map the reads to (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

maqpair_reverse_suffix suffix of reverse files (string, default:_r.bfq)

maqpair_RF_maxdist max outer distance for an RF readpair (corresponds to the -A parameter). This applies to long insert illumina pairs (integer, default:15000)

maqpair_maxdist max outer distance for a (non RF) readpair. This applies to illumina matepairs - i.e. short inserts (integer, default:250)

A.35 moatest

A.35.1 Targets

(empty) Execute the default target:

moatest

moatest Do nothing - no need to call this.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.35.2 Parameters

Required parameters

txt test variable (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

test_opt test variable (string, default:konijntje)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.36 mummer

Run mummer between two sequences

A.36.1 Targets

(empty) Execute the default target:

mummer

mummer : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.36.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

mum_input_a_dir Set 1 input fasta files (directory, default:)

mum_input_b_dir Set 1 input fasta files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

mum_input_a_extension file extension for the files in mum_input_a_dir (string, default:fasta)

mum_input_a_glob glob to select a subset of files from mum_input_a_dir (string, default:*)

mum_input_a_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

mum_input_a_limit Number of files to use, if not defined: all files (integer, default:)

mum_input_b_extension file extension for the files in mum_input_b_dir (string, default:fasta)

mum_input_b_glob glob to select a subset of files from mum_input_b_dir (string, default:*)

mum_input_b_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

mum_input_b_limit Number of files to use, if not defined: all files (integer, default:)

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

mum_plot_raw plot an alternative visualization where mummer does not attempt to put the sequences in the correct order (T|F)

A.37 ncbi

A.37.1 Targets

(empty) Execute the default target:

ncbi

ncbi Downloads from NCBI

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.37.2 Parameters

Required parameters

ncbi_query NCBI query (for example txid9397[Organism%3Aexp]) (string, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

ncbi_db NCBI database (string, default:nuccore)

ncbi_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. (string, default:)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.38 newbler

Run a simple, out of the box, newbler assembly. As an extra feature, this template automatically creates uniquely named links to the two main output fasta files (454AllContigs.fna, 454LargeContigs.fna). This is convenient for subsequence 'gather' steps. The links are named after the directory.

A.38.1 Targets

(empty) Execute the default target:

newbler

newbler : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.38.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

newbler_input_dir input SFF files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

newbler_input_extension file extension for the files in newbler_input_dir (string, default:sff)

newbler_input_glob glob to select a subset of files from newbler_input_dir (string, default:*)

newbler_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

newbler_input_limit Number of files to use, if not defined: all files (integer, default:)

newbler_library_name A library identifier for this assembly. This is used to create an extra fasta file, named using this variable, that contain the generated contigs with their ids prepended with the library id. (string, default:)

newbler_mids mids to use for this assembly (string, default:)

newbler_mid_configuration Mid configuration file to use (file, default:)

newbler_min_identity Minimal overalp identity used during assembly (integer, default:)

newbler_largecontig_cutoff min length of a contig in 454LargeContigs.fna (integer, default:)

A.39 nstretch

Run NSTRETCH on an set of input files

A.39.1 Targets

(empty) Execute the default target:

nstretch

nstretch : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.39.2 Parameters

Required parameters

nstretch_input_dir input dir with the fasta files (directory, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

nstretch_input_extension extension of the input files (string, default:fasta)

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

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.40 pregap

Run Pregap. Note that running phrap could be a part of this.

A.40.1 Targets

(empty) Execute the default target:

pregap

pregap Run pregap

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.40.2 Parameters

Required parameters

input_dir Directory with the input data (string, default:)

input_pattern file name pattern (string, default:)

cloning_vector File containing the cloning vector (file, default:)

sequencing_vector File containing the sequencing vector (file, default:)

ecoli_screenseq File containing ecoli screen sequences (file, default:)

repeat_masker_lib File with a repeatmasker library (file, default:)

vector_primerfile File with the vector primers (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

quality_value_clip quality cutoff (integer, default:10)

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

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.41 project

A.41.1 Targets

(empty) Execute the default target:

project

project This template does not do anything - it is a project placeholder.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.41.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

project_description A description of what this project is supposed to achieve, how to use it, and what parameters are most important to set (string, default:)

A.42 repeatmasker

A.42.1 Targets

(empty) Execute the default target:

repm

repm : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.42.2 Parameters

Required parameters

repm_input_file blast database of the reference set (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

repm_species species (string, default:repmfolds)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.43 revseq

A.43.1 Targets

(empty) Execute the default target:

getorf

getorf : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.43.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

getorf_input_dir Input files for getorf (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

getorf_input_extension file extension for the files in getorf_input_dir (string, default:fasta)

getorf_input_glob glob to select a subset of files from getorf_input_dir (string, default:*)

getorf_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

getorf_input_limit Number of files to use, if not defined: all files (integer, default:)

getorf_gff_source source field to use in the gff. (string, default:getorf)

getorf_minsize minimal nucleotide size of the predicted ORF. (integer, default:30)

getorf_maxsize maximal nucleotide size of the predicted ORF. (integer, default: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. (0|1|2|3|4|5|6|7|8|9|10|*11*|12|13|14|15|16|21|22|23)

getorf_find What to output? 0: Translation between stop codons, 1: Translation between start & stop codon, 2: Nucleotide sequence between stop codons; 3: Nucleotide sequence between start and stop codons. Default: 3 (0|1|2|*3*)

A.44 sam2bam

A.44.1 Targets

(empty) Execute the default target:

sam2bam

sam2bam : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.44.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

sam2bam_input_dir input SAM files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

sam2bam_input_extension file extension for the files in sam2bam_input_dir (string, default:sam)

sam2bam_input_glob glob to select a subset of files from sam2bam_input_dir (string, default:*)

sam2bam_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

sam2bam_input_limit Number of files to use, if not defined: all files (integer, default:)

A.45 scaffolder

A.45.1 Targets

(empty) Execute the default target:

scaf

scaf : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.45.2 Parameters

Required parameters

scaf_reference_file blast database of the reference set (file, default:)

scaf_input_file input file with the sequences to scaffold (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

scaf_prefix prefix for scaffolding output files (string, default:scaffolds)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.46 sffinfo

A.46.1 Targets

(empty) Execute the default target:

sffinfo

sffinfo Use the Roche sffinfo tool to extract reads, quality scores, flowgrams and accession ids from one or more sff files

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.46.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

sffinfo_input_dir Sff input files (directory, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

sffinfo_input_extension file extension for the files in sffinfo_input_dir (string, default:sff)

sffinfo_input_glob glob to select a subset of files from sffinfo_input_dir (string, default:*)

sffinfo_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

sffinfo_input_limit Number of files to use, if not defined: all files (integer, default:)

sffinfo_accessions Output the accessions (T|F)

sffinfo_sequences Output the sequences (T|F)

sffinfo_quality Output quality scores (T|F)

sffinfo_flowgrams output the flowgrams (T|F)

sffinfo_untrimmed output untrimmed sequences & qualities (T|F)

A.47 traverse

A.47.1 Targets

(empty) Execute the default target:

traverse

traverse Do nothing - no need to call this.

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.47.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.48 varscan

Run VARSCAN to detect snps

A.48.1 Targets

(empty) Execute the default target:

varscan

varscan : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.48.2 Parameters

Required parameters

varscan_input_file Varscan input alignments file (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

varscan_extra_params location of varscan.pl, defaults to '/usr/lib/perl5/site_perl/5.8.8/varscan.pl' (string, default:)

varscan_output_name Base name of the output files (string, default:out)

varscan_perl_file the varscan (perl) executable (file, default:)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.49 vmatch

Run VMATCH on an set of input files (query) vs a database index.

A.49.1 Targets

(empty) Execute the default target:

vmatch

vmatch : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.49.2 Parameters

Required parameters

vmatch_db vmatch db to compare against (file, default:)

vmatch_input_file input file with the sequences to map (file, default:)

title A job name - Describe what you are doing (string, default:)

Optional parameters

vmatch_extra_parameters extra parameters to feed to vmatch (string, default:)

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

A.50 vmatchdb

A.50.1 Targets

(empty) Execute the default target:

vmatchdb

vmatchdb : clean : removes all results from this job all : executes the default target and : into subdirectories to execute any : other moa makefile it encounters

A.50.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

vmatchdb_input_dir Input files for vmatch (directory, default:)

vmatchdb_name Name of the vmatch index to create (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

vmatchdb_input_extension file extension for the files in vmatchdb_input_dir (string, default:fasta)

vmatchdb_input_glob glob to select a subset of files from vmatchdb_input_dir (string, default:*)

vmatchdb_input_sort Sort order. Choose from: u - unsorted, s - size, sr - size reverse, t - time, tr - time reverse ()

vmatchdb_input_limit Number of files to use, if not defined: all files (integer, default:)

vmatchdb_pl Prefix length (integer, default:)

A.51 vpcr

A.51.1 Targets

(empty) Execute the default target:

vpcr

vpcr Predict the fragments that woulc be generated by a PCR

clean removes all results from this job

all executes the default target and

into subdirectories to execute any

other moa makefile it encounters

A.51.2 Parameters

Required parameters

title A job name - Describe what you are doing (string, default:)

vpcr_bowtie_db Location of the bowtie database used for the vpcr (file, default:)

vpcr_primer_1 First primer to use (string, default:)

vpcr_primer_2 Second primer to use (string, default:)

Optional parameters

moa_precommand A single command to be executed before the main operation starts. For more complicated processing, please override the moa_preprocess target in the local Makefile. (string, default:)

moa_postcommand A single shell command to be executed after the Moa is finished. For more complex processing please override the moa_postprocess target in the local Makefile. (string, default:)

vpcr_insert_min minimal insert size for a fragment (integer, default:10)

vpcr_insert_max maximum insert size for a vpcr fragment (integer, default:10000)

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

[3] Mark W E J Fiers, Ate van der Burgt, Erwin Datema, Joost C W de Groot, and Roeland C H J van Ham. High-throughput bioinformatics with the cyrille2 pipeline system. *BMC Bioinformatics*, 9:96, 2008.

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

[5] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, Nov 2004.

[6] Simon C Potter, Laura Clarke, Val Curwen, Stephen Keenan, Emmanuel Mongin, Stephen M J Searle, Arne Stabenau, Roy Storey, and Michele Clamp. The ensembl analysis pipeline. *Genome Res*, 14(5):934–941, May 2004.

[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] James Taylor, Ian Schenck, Dan Blankenberg, and Anton Nekrutenko. Using galaxy to perform large-scale interactive data analyses. *Curr Protoc Bioinformatics*, Chapter 10:Unit 10.5, Sep 2007.