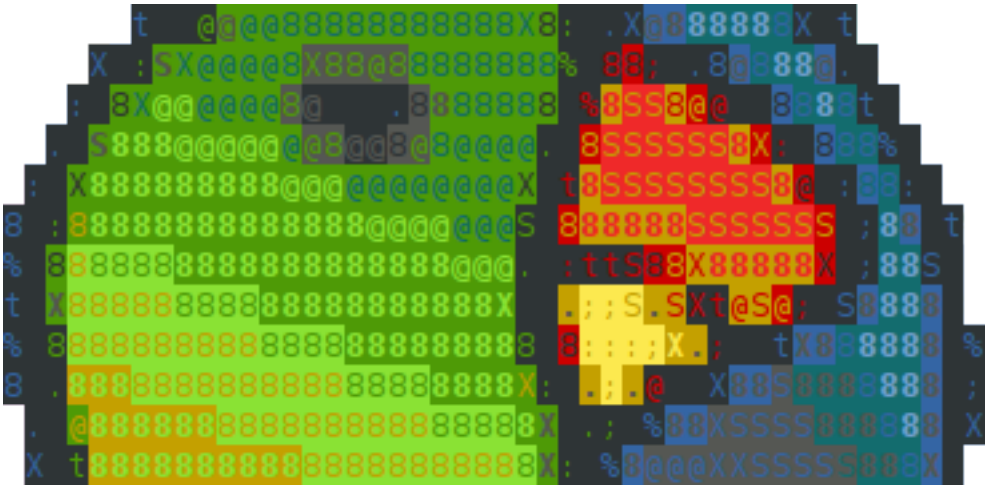


Moa Documentation

Release 0.10.11

Mark Fiers

May 16, 2011

CONTENTS

1 Introduction **3**
- 1.1 Example session . . . 4

2 Installation **7**
- 2.1 Prerequisites . . . 7
- 2.2 Installing Moa using easy_install . . . 7
- 2.3 Bioinformatics tools . . . 8

3 Installation from source **9**
- 3.1 Configuration . . . 9

4 Creating a pipeline **11**
- 4.1 Guiding principles . . . 11
- 4.2 Setting up new jobs - `moa new` . . . 11

5 Running a pipeline **13**
- 5.1 Running one job . . . 13
- 5.2 Running a series of jobs . . . 13

6 Filesets **15**

7 How to write a template **17**

8 Command reference **21**
- 8.1 moa **!** . . . 21
- 8.2 moa **adhoc** . . . 21
- 8.3 moa **archive** . . . 21
- 8.4 moa **blog** . . . 22
- 8.5 moa **cp** . . . 22
- 8.6 moa **err** . . . 23
- 8.7 moa **files** . . . 23
- 8.8 moa **help** . . . 23
- 8.9 moa **history** . . . 23
- 8.10 moa **kill** . . . 24
- 8.11 moa **list** . . . 24
- 8.12 moa **lock** . . . 24
- 8.13 moa **log** . . . 24
- 8.14 moa **map** . . . 25
- 8.15 moa **new** . . . 25

8.16 moa **out** . . . 25
8.17 moa **pause** . . . 26
8.18 moa **raw_commands** . . . 26
8.19 moa **raw_parameters** . . . 26
8.20 moa **refresh** . . . 26
8.21 moa **ren** . . . 27
8.22 moa **resume** . . . 27
8.23 moa **set** . . . 27
8.24 moa **show** . . . 27
8.25 moa **simple** . . . 28
8.26 moa **status** . . . 28
8.27 moa **tag** . . . 28
8.28 moa **template** . . . 29
8.29 moa **template_dump** . . . 29
8.30 moa **template_set** . . . 29
8.31 moa **test** . . . 29
8.32 moa **tree** . . . 29
8.33 moa **unittest** . . . 30
8.34 moa **unlock** . . . 30
8.35 moa **unset** . . . 30
8.36 moa **version** . . . 30
8.37 moa **welcome** . . . 30

9 Templates **31**
9.1 adhoc . . . 31
9.2 bamextract . . . 33
9.3 bartab . . . 34
9.4 bfast_aln . . . 36
9.5 bfast_db . . . 41
9.6 bidibebla . . . 44
9.7 blast . . . 46
9.8 blast . . . 49
9.9 blastdb . . . 52
9.10 blat . . . 53
9.11 bowtie . . . 55
9.12 bowtie_pe . . . 57
9.13 bowtie_se . . . 59
9.14 bowtiedb . . . 61
9.15 bwa_aln . . . 62
9.16 bwa_index . . . 67
9.17 bwa_sampe . . . 68
9.18 bwa_samse . . . 72
9.19 clean_fasta . . . 73
9.20 clustalgroup . . . 74
9.21 clustalpair . . . 75
9.22 clustalw . . . 76
9.23 concatenate . . . 77
9.24 dotself . . . 78
9.25 dottup . . . 80
9.26 empty . . . 81
9.27 f2b . . . 81

9.28 fasta2gff . . . 82
9.29 fastqc . . . 84
9.30 fastx_clipper . . . 85
9.31 fastx_qual_stats . . . 88
9.32 fq2bq . . . 91
9.33 gather . . . 92
9.34 genemarks . . . 94
9.35 getorf . . . 95
9.36 getorf . . . 98
9.37 glimmer3 . . . 100
9.38 gmap . . . 102
9.39 gmapdb . . . 105
9.40 gsmap . . . 106
9.41 h_blast . . . 107
9.42 kanga . . . 110
9.43 kangar_pe . . . 114
9.44 kangar_se . . . 118
9.45 kangax . . . 121
9.46 lftp . . . 124
9.47 map . . . 127
9.48 maq_pe . . . 128
9.49 maq_se . . . 133
9.50 maqpair . . . 137
9.51 moatest . . . 138
9.52 mummer . . . 139
9.53 ncbi . . . 141
9.54 newbler . . . 142
9.55 newjobtest . . . 144
9.56 nstretch . . . 145
9.57 orthomcl . . . 146
9.58 pregap . . . 148
9.59 primersearch . . . 150
9.60 project . . . 151
9.61 r . . . 152
9.62 repmask . . . 153
9.63 sam2bam . . . 155
9.64 samtools_pileup . . . 156
9.65 scaf . . . 161
9.66 sffinfo . . . 162
9.67 simple . . . 165
9.68 soap_aln_pe . . . 166
9.69 soap_aln_se . . . 171
9.70 soapdb . . . 174
9.71 statsidx . . . 175
9.72 unittest . . . 176
9.73 upload2gbrowse . . . 178
9.74 varscan . . . 180
9.75 vmatch . . . 181
9.76 vmatchdb . . . 182
9.77 vpcr . . . 183
9.78 vpcr_list . . . 185

10 Moa API **187**
10.1 moa.actor . . . 187
10.2 moa.commands . . . 187
10.3 moa.job . . . 187
10.4 moa.jobConf . . . 190
10.5 moa.sysConf . . . 190
10.6 moa.ui . . . 191
10.7 moa.utils . . . 191
10.8 moa.template . . . 192
10.9 moa.template.provider . . . 193
10.10 moa.backend . . . 193
10.11 moa.plugin . . . 193
10.12 Yaco . . . 200
10.13 fist . . . 202

11 More information **205**

12 Indices and tables **207**

Bibliography **209**

Python Module Index **211**

Index **213**

lightweight workflows for bioinformatics

Contents:

NOTE: both the software and the manual are under development. Things might change.

CHAPTER

ONE

INTRODUCTION

A bioinformatics project commonly consists of a number of separate steps that chain (3rd party) tools together. To finetune the behaviour of of such an analysis pipeline, a number of custom steps are often necessary. Particularly when such projects get bigger, it becomes vital to organize, automate and store analysis pipelines.

There are many different ways to organize bioinformatics projects. Many bioinformaticians use the command line or tailor made scripts to organize and automate their work. This approach has obvious advantages, most importantly flexibility. Potential downsides to scripting are that a project easily becomes disorganized and untrackable unless measures are taken.

***Moa** to assist in organizing, automating and maintaining a command line bioinformatics project without loss of flexibility.**

The best way to understand how Moa aims to achieve this is by a simple example, imagine you would like to run a BLAST analysis:

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

In the first line, a new BLAST job (titled "demo run") is created in the current directory. What really happens is that Moa copies the *blast* template to the current directory. The *blast* template knows how to execute a [BLAST] job, but needs at least two variables defined, defined in the next two lines. *moa run* executes the analysis and 'blasts' the input sequences (in `../sequences/*.fasta`) against the database in `/data/blast/db/nt`. BLAST output files (XML) are generated and converted to GFF (GFF conversion is an extra of the template, not part of the BLAST suite). The one to last statement is probably most typical of the flexibility provided by Moa; it is a single shell commmand that will be executed after BLAST is executed (there is a corresponding `moa_preprocess`). This shell comamand filters all BLAST hits that have the word "polymerase" in their description into a separater GFF file.

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.
- ***Create reusable 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`).

1.1 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:

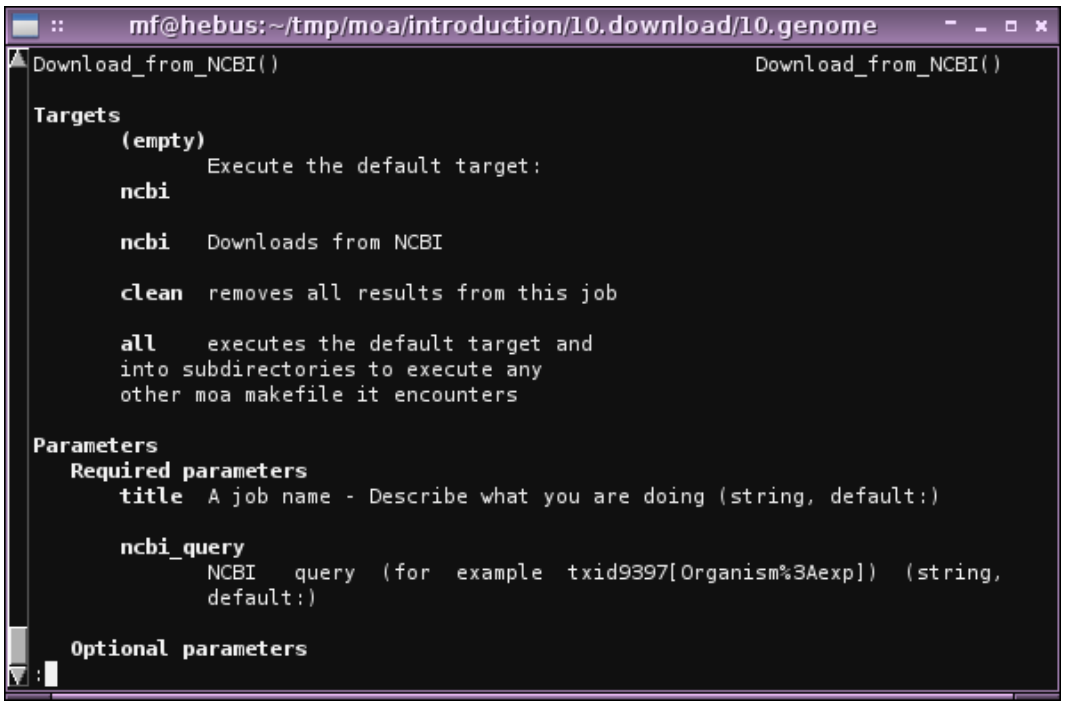
- Each moa job is contained in one directory. Output files of a job are typically stored in that directory. It is not possible to have more than one Moa jobs in a directory.
- Moa is able to automatically execute all jobs in a directory and the underlying sub-directories (using `moa all`).

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

TWO

INSTALLATION

2.1 Prerequisites

Moa is developed and tested on Ubuntu and RHEL 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). Necessary either to download the Moa software from github, or, to make use of the integrated version control.
- **Python (2.6). Moa is not tested with other** versions of Python
- **Bash (4.1.2). Many of the** embedded scripts expect the Bash shell.
- **Gnu Make Standard Library (GSML). A** set of standard routines for Gnu Make. GSML is distributed together with Moa.
- A number of support scripts & templates depend on Biopython. Consider installing it before starting to use Moa.
- *Python-dev*: the Python development package. A number of the prerequisites to be installed by easy_install try to compile C libraries, and need this to be installed. Although all of them have backup, python only, alteratives; from a performace perspective it is probably smart to have this installed:

  ```
  sudo apt-get install python-dev
  ```

- *python-yaml*: Again - this is not really necessary, but might improve performace. If omitted, easy_install will try to install and complile it - and use a python only version if that fails:

  ```
  sudo apt-get install python-dev
  ```

- Python easy_install is the preferred way to install Moa and a number of further prerequisites.

2.2 Installing Moa using easy_install

Easy:

```
sudo easy_install moa
```

The commandline will install moa and a number of other python libraries

There is a number of other prerequisites Moa requires the following modules to be installed:

- pyyaml
- Jinja2
- Ruffus
- gitpython
- Yaco
- fist

These can be installed using install Moa:

```
easy_install-2.6 moa
```

2.3 Bioinformatics tools

Each of the wrapped tools requires the tools to be present. Usually, Moa expects all tools to be present & executable on the system PATH. The standard Moa distribution comes with wrappers for:

- Blast
- BWA
- Bowtie
- Soap

and many more

CHAPTER

THREE

INSTALLATION FROM SOURCE

Moa is hosted on and can be installed from github:

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

3.1 Configuration

Configuration of Moa is simple, and can be done by sourcing the *moainit* script:

```
. ~/moa/bin/moainit
```

(Note the dot!, alternatively use: `source ~/moa/bin/moainit`)

It is probably a good idea to add this line to your `~/.bashrc` for future sessions.

Moa should now work, try *moa –help* or, for a more extensive test: *moa unittest*

CHAPTER

FOUR

CREATING A PIPELINE

4.1 Guiding principles

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

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

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

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

4.2 Setting up new jobs - `moa new`

Creating a new job is done with the `moa new` command.

CHAPTER

FIVE

RUNNING A PIPELINE

5.1 Running one job

5.2 Running a series of jobs

CHAPTER

SIX

FILESETS

Filesets are an important part of Moa - they are used to define in- and output files of Moa jobs

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

CHAPTER

SEVEN

HOW TO WRITE A TEMPLATE

A MOA template is made up of a `.moa file` and a `.jinja2` (or `.mk`) file.

The `.moa` file mainly contains input-output file sets and parameter options used for the bash command(s). Some of these options have default values which the user can change while constructing the job.

The `.jinja2` file includes information to structure the command(s). It is written in jinja, which is a templating language for python and is simple to write and easy to understand.

These files are used by the backend, currently *ruffus*, that manages file set and parameter dependencies to make pipelines and render commands to the bash prompt. Initially, *GNU make* was the backend used. It is very powerful but some of its limitations and its complexity led to including *ruffus* as an option for the backend as well.

The easiest way to write a moa template is to edit an existing template to suit your requirements. This involves understanding the parts of an existing template.

The `bwa_aln` template is used as an example below. Just as a background, the *bwa aln* command takes a FASTQ file as input and aligns it to a reference genome that was previously indexed. The output is a .sai file with the alignments.

The `bwa_aln.moa` file has some main components:

- *Backend*

  ```
  backend: ruff
  ```

  This is 'ruff' which means that ruffus is used in the python script at a lower level to read the template .moa and .jinja2 file, and render the corresponding commands to the bash prompt.

- *Commands*

  ```
  commands:
    run:
      mode: map
      help:  run bwa aln
    clean:
      mode: simple
      help: Remove all job data, not the Moa job itself, note that this must be imple
  ```

  This indicates the function names that you will later define. In the example above, there are 2 commands- run and clean, so `moa run` or `moa clean` on the command prompt in the job directory would execute these functions.

- *Filesets*

```
filesets:
  input:
    category: input
    extension: fq
    help: Fastq input files
    glob: '*'
    optional: false
    type: set
  output:
    category: output
    dir: .
    extension: sai
    glob: '{{ input_glob }}'
    source: input
    type: map
```

Like the name, each filesets refer to a set of files in a single directory. The bwa_aln template shows 2 filesets: `input` and `output`.

- *Category*: is essentially used to separate input from output.
- *Extension*: refers to the type of file(s) required or generated.
- *Glob*: searches for files with a specified pattern. Moa, by default (glob= *) automatically processes all files of the specified input extension in the directory specified. By specifying a glob, Moa will only process those files whose name pattern matches what is in the glob.
- *Type*: refers to the data type of the fileset or parameter.

  A fileset can either be of `set` or `map` type. The type `set` refers to a simple set of files in a directory. The type `map` refers to a set of files that are linked to what their `source` value is. In the above code, the output fileset is mapped to the input fileset.
- *Dir*: the directory of the output fileset is '.', which means that the output files will be placed in the current working directory.

- *Parameter category order*

```
parameter_category_order:
  - ''
  - input
  - system
  - advanced
```

- *Parameters*

```
mismatch_penalty:
  category: ''
  default: 3
  help: mismatch penalty
  optional: true
  type: integer
```

They are the variables/options that specify a command.

- *Category*:
- *Default*: is the value that is used by default if not changed by the user.

  - *Optional*: specifies if it is necessary for the user to fill in a value for the variable. If `optional` is false, the user has to indicate a value for the parameter in order to execute the job.
  - *Type*: specifies the data type of the variable eg. integer, string, boolean.

- *Moa_id*

  ```
  moa_id: bwa_aln
  ```

  is supposed to be the same as the filename. Ideally something descriptive (eg. bwa_aln). This is used to later link to the other template file.

The other template file is ''bwa_aln.jinja2" which is written in jinja, a templating language for python. *Note that the jinja2 file name is the same as the moa file name.*

Important features of the bwa_aln.jinja2 file are:

- The three hash's (###) specify the start of a function and are followed by the function name. In our bwa_aln example, we have defined 2 funtions: `run` and `clean`.

  ```
  ### run
  ```

- This defination is followed by a set of commands which you would want to be executed when you type `moa run` or `moa_clean` in the bwa_aln job directory. The commands in our example file look the same as what you would put in the command prompt but the values of the parameters are bought from the .moa file and hence it's value is replaced by the parameter name.

  ```
  bwa aln {{db}}                                  \
      -n {{edit_dist_missing_prob}}               \
      .                                           \
      .                                           \
      .                                           \
      {{ input }}                                 \
      -f {{ output}}
  ```

- It is also possible to add if-else statements or other computing blocks in accordance with the design language.

  ```
  {% if color_space %} -c {% endif %}
  ```

CHAPTER

EIGHT

COMMAND REFERENCE

8.1 moa !

Moa-fy the last (bash) command issued

Usage:

```
moa !
```

Description:

Set the 'process' parameter to the last issued command. If no moa job exists, create a 'simple'job.

Usage of this command will be logged

8.2 moa adhoc

Create an adhoc analysis

Usage:

```
moa adhoc -t "title" -- echo "do something"
```

Description:

Creates an adhoc job.

Usage of this command will be logged

8.3 moa archive

Archive a job,

Description:

Archive a job, or tree with jobs for later execution.

This command stores only those files that are necessary for execution of this job, that is: templates & configuration. In & output files, and any other file are ignored. An exception to this are all files that start with 'moa.'

Usage:

```
moa archive
```

or

```
moa archive -r
```

The latter archives all jobs in subdirs of the current directory.

Note that only those directories that contain a moa job are included into the archive.

Usage of this command will be logged

8.4 moa blog

record a short note

Usage:

```
moa blog
```

Description:

Allows a user to enter a short note that is appended to moa.description (including a timestamp). Use it as follows:

```
$ moa blog
Here you can enter a short, conscise, multi-
line message describing what you have been
doing
[ctrl-d]
```

Note: the ctrl-d needs to be given on an empty line. The text is appended to moa.desciption. In the web interface this is converted to Markdown.

Usage of this command will be logged

8.5 moa cp

Copy a moa job

Description:

Copy a moa job, or a tree with jobs.

moa cp copies only those files defining a job: the template files and the job configuration. Additionaly, all files in the moa directory that start with *moa.* (for example *moa.description* are copied as well. Data and log files are not copied!

The command has two modes of operation. The first is:

```
moa cp 10.from 20.to
```

copies the moa job in 10.from to a newly created 20.to directory. If the *20.to* directory already exists, a new directory is created in *20.to/10.from*. As an shortcut one can use:

```
moa cp 10.from 20
```

in which case the job will be copied to the *20.from* directory.

If the source (*10.from*) directory is not a Moa job, the command exits with an error.

The second mode of operation is recursive copying:

```
moa cp -r 10.from 20.to
```

in which case all subdirectories under 10.from are traversed and copied - if a directory contains a Moa job.

::TODO.. Warn for changing file & dir links

Usage of this command will be logged

8.6 moa err

Returns stderr of the last moa run

*Usage of this command will **NOT* be logged*

8.7 moa files

Show an overview of the files for this job

Description:

moa files - Display discovered & inferred files for this job

Usage:

```
moa files
```

Display a list of all files discovered (for input & prerequisite type filesets) and inferred from these for map type filesets.

Usage of this command will be logged

8.8 moa help

Display help for a template

Usage of this command will be logged

8.9 moa history

display a version control log

Description:

Print a log to screen

Usage of this command will be logged

8.10 moa kill

Kill a job

Description:

See if a job is running, if so - kill it

Usage of this command will be logged

8.11 moa list

Print a list of all known templates

Description:

moa list - Print a list of all known templates

Usage:

```
moa list
moa list -l
```

Print a list of all templates known to this moa installation. If the option '-l' is used, a short description for each tempalte is printed as well.

Usage of this command will be logged

8.12 moa lock

Lock this job - prevent execution

Usage of this command will be logged

8.13 moa log

Show the logs for this job

Description:

moa log - show a log of the most recent moa calls

Usage:

```
moa log [LINES]
```

Shows a log of moa commands executed. Only commands with an impact on the pipeline are logged, such as *moa run* & *moa set*. The number of log entries to display can be controlled with the optional LINES parameter.

*Usage of this command will **NOT* be logged*

8.14 moa map

Create a "map" adhoc analysis

Usage:

```
moa map -t "title" -- echo "do something"
```

Description:

Create a 'map' adhoc job.

There are a number of ways this command can be used:

```
$ moa map -t 'a title' -- echo 'define a command'
```

Anything after – will be the executable command. If omitted, Moa will query the user for a command.

Moa will also query the user for input & output files. An example session:

```
$ moa map -t 'something intelligent'
process:
> echo 'processing {{ input }} {{ output }}'
input:
> ../10.input/*.txt
output:
> ./*.out
```

Assuming you have a number of text files in the *../10/input/* directory, you will see, upon running:

```
processing ../10.input/test.01.txt ./test.01.out
processing ../10.input/test.02.txt ./test.02.out
processing ../10.input/test.03.txt ./test.03.out
...
```

Usage of this command will be logged

8.15 moa new

Create a new Moa job

Description:

moa new

Usage:

```
moa new TEMPLATE_NAME -t 'a descriptive title'
```

Usage of this command will be logged

8.16 moa out

Returns stdout of the last moa run

*Usage of this command will **NOT* be logged*

8.17 moa pause

Pause a job

Description:

pause a running job

Usage of this command will be logged

8.18 moa raw_commands

{}

Description:

(private) **moa raw_commands** - Print a list of all known commands

Usage:

```
moa raw_commands
```

Print a list of known Moa commands, both global, plugin defined commands as template specified ones. This command is mainly used by software interacting with Moa.

*Usage of this command will **NOT* be logged*

8.19 moa raw_parameters

{}

Description:

(private) **moa raw_parameters** - Print out a list of all known parameters

Usage:

```
moa raw_parameters
```

print a list of all defined or known parameters

*Usage of this command will **NOT* be logged*

8.20 moa refresh

Reload the template

Description:

Refresh the template - i.e. reload the template from the central repository.

Usage of this command will be logged

8.21 moa ren

Rename/renumber a job

Description:

Renumber or rename a moa job..

Usage of this command will be logged

8.22 moa resume

Resume a job

Description:

pause a running job

Usage of this command will be logged

8.23 moa set

Set, change or remove variables

Usage:

```
moa set [KEY] [KEY=VALUE]
```

Description:

This command can be used in a number of ways:

```
moa set PARAMETER_NAME=PARAMETER_VALUE
moa set PARAMETER_NAME='PARAMETER VALUE WITH SPACES'
moa set PARAMETER_NAME
```

In the first two forms, moa sets the parameter *PARAMETER_NAME* to the *PARAMETER_VALUE*. In the latter form, Moa will present the user with a prompt to enter a value. Note that the first two forms the full command lines will be processed by bash, which can either create complications or prove very useful. Take care to escape variables that you do not want to be expandend and use single quotes where you can.

Usage of this command will be logged

8.24 moa show

Show configured variables

Usage:

```
moa show
```

Description:

Show all parameters know to this job. Parameters in **bold** are specifically configured for this job (as opposed to those parameters that are set to their default value). Parameters in red are not configured, but need to be for the template to operate. Parameters in blue are not configured either, but are optional.

*Usage of this command will **NOT* be logged*

8.25 moa simple

Create a "simple" adhoc analysis

Usage:

```
moa simple -t "title" -- echo "do something"
```

Description:

Create a 'simple' adhoc job. Simple meaning that no in or output files are tracked.

There are a number of ways this command can be used:

```
moa simple -t 'a title' -- echo 'define a command'
```

Anything after – will be the executable command. Note that bash will attempt to process the command line. A safer method is:

```
moa simple -t 'a title'
```

Moa will query you for a command to execute (the parameter *process*).

Usage of this command will be logged

8.26 moa status

Show the state of the current job

Description:

moa status - print out a short status status message

Usage:

```
moa status
```

*Usage of this command will **NOT* be logged*

8.27 moa tag

Tag the current version

Usage of this command will be logged

8.28 moa template

Display the template name

Description:

moa template - Print the template name of the current job

Usage:

```
moa template
```

Usage of this command will be logged

8.29 moa template_dump

Display the raw template description

Description:

moa template_dump - Show raw template information

Usage:

```
moa template_dump [TEMPLATE_NAME]
```

Show the raw template data.

Usage of this command will be logged

8.30 moa template_set

Set a template parameters

Description:

moa template_set - set a template parameter.

This only works for top level template parameters

Usage of this command will be logged

8.31 moa test

Test the currennt configuration

Usage of this command will be logged

8.32 moa tree

display a directory tree

*Usage of this command will **NOT* be logged*

8.33 moa unittest

Run Moa unittests

Usage of this command will be logged

8.34 moa unlock

Unlock this job

Usage of this command will be logged

8.35 moa unset

Remove a variable

Usage:

```
moa unset KEY
```

Description:

Remove a configured parameter from this job. In the parameter was defined by the job template, it reverts back to the default value. If it was an ad-hoc parameter, it is lost from the configuration.

Usage of this command will be logged

8.36 moa version

Print the moa version

Description:

moa version - Print the moa version number

*Usage of this command will **NOT* be logged*

8.37 moa welcome

Display a welcome text

Description:

print a welcome message

Usage of this command will be logged

CHAPTER

NINE

TEMPLATES

Contents:

9.1 adhoc

Execute an ad hoc analysis

:: The *adhoc* template assists in running one-liners - possibly on a set of input files

9.1.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.1.2 Filesets

input::

Input files for adhoc

type: *map*
source: *{}*
category: *input*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.1.3 Parameters

mode::

operation mode: *seq*, sequential: process the input files one by one; *par*, parallel: process the input files in parallel (use with *-j*); *all*: process all input files at once (use *$^* in *adhoc_process*) and *simple*: Ignore input files, just execute *adhoc_process* once.

type: *set*
default: *simple*
optional: *True*

name_sed::

A sed expression which can be used to derive the output file name for each input file (excluding the path). The sed expression is executed for each input file name, and the result is available as $t in the $(adhoc_process) statement. Make sure that you use single quotes when specifying this on the command line

type: *string*
default: *s/a/a/*
optional: *True*

output_dir::

Output subdirectory

type: *directory*
default: .
optional: *True*

process::

Command to execute for each input file. The path to the input file is available as $< and the output file as $t. (it is not mandatory to use both parameters, for example "cat $< > output" would concatenate all files into one big file

type: *string*
default: *echo "needs a sensbile command"*
optional: *True*

touch::

use touch files to track if input files have changed.

type: *set*
default: *T*
optional: *True*

9.1.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.2 bamextract

Bamextract

:: Extract one sequence from a bam file

9.2.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.2.2 Parameters

bam_input::

BAM input file

type: *file*
default: ‘‘
optional: *False*

fasta_file::

Fasta file with the reference sequence data

type: *file*
default: ‘‘
optional: *False*

gff_file::

GFF annotation file to extract data from

type: *file*
default: ‘‘
optional: *True*

haplotypes::

No of haplotypes in the sample

type: *integer*
default: *2*
optional: *True*

seq_id::

List of sequence ids to extract

type: *string*
default: ‘‘
optional: *False*

9.2.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.3 bartab

Bartab

:: BARTAB - a tool to process sff files

9.3.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run

9.3.2 Parameters

extra_parameters::

extra parameters to feed bartab

type: *string*
default: ‘‘
optional: *True*

forward_primer::

remove forward primer

type: *string*
default: ‘‘
optional: *True*

in::

input file for bartab

type: *file*
default: ‘‘
optional: *False*

map::

A file mapping barcodes to metadata

type: *file*
default: ‘‘
optional: *True*

min_length::

minimun acceptable sequence length

type: *integer*
default: *50*
optional: *True*

out::

base output name

type: *integer*
default: *bartab*
optional: *True*

qin::

Quality scores for the input fasta file

type: *file*
default: ‘‘
optional: *True*

reverse_primer::

remove reverse primer

type: *string*
default: ‘‘
optional: *True*

trim::

Trim barcode

type: *set*
default: *T*
optional: *True*

9.3.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.4 bfast_aln

:: Generate bam format alignments using bfast

9.4.1 Commands

clean Remove all job data, not the Moa job itself

run run bfast match, localalign, postprocess commands

9.4.2 Filesets

fa_input::

fasta input file

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_input::

fastq input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output_aln::

{}

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

output_bam::

{}

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.4.3 Parameters

algorithm_colour_space::

true -> colour space, false -> NT space

type: *boolean*
default: *False*
optional: *True*

avg_mism_qual::

Specifies the average mismatch quality

type: *integer*
default: *10*
optional: *True*

extra_params_localalign::

Any extra parameters for the localalign command

type: *string*
default: ‘‘
optional: *True*

extra_params_match::

Any extra parameters for the match command

type: *string*
default: ‘‘
optional: *True*

extra_params_postprocess::

Any extra parameters for the postprocess command

type: *string*
default: ‘‘
optional: *True*

min_mapping_qual::

Specifies to remove low mapping quality alignments

type: *integer*
default: *-2147483648*
optional: *True*

min_norm_score::

Specifies to remove low (alignment) scoring alignments

type: *integer*
default: *-2147483648*
optional: *True*

output_format::

0 - BAF, 1 - SAM

type: *integer*
default: *1*
optional: *True*

paired_opp_strands::

Specifies that paired reads are on opposite strands

type: *boolean*
default: *False*
optional: *True*

pairing_std_dev::

Specifies the pairing distance standard deviation to examine when recuing

type: *float*
default: *2.0*
optional: *True*

print_params::

print program parameters

type: *boolean*
default: *False*
optional: *True*

thread_num::

Specifies the number of threads to use

type: *integer*
default: *1*
optional: *True*

timing_information::

specifies output timing information

type: *boolean*
default: *True*
optional: *True*

ungapped_aln::

Do ungapped local alignment

type: *boolean*
default: *False*
optional: *True*

ungapped_pairing_rescue::

Specifies that ungapped pairing rescue should be performed

type: *boolean*
default: *False*
optional: *True*

unpaired_reads::

True value specifies that pairing should not be performed

type: *boolean*
default: *False*

optional: *True*

usage_summary::

Display usage summary (help)

type: *boolean*
default: *False*
optional: *True*

which_strand::

0 - consider both strands, 1 - forwards strand only, 2 - reverse strand only

type: *integer*
default: *0*
optional: *True*

9.4.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Feb 15 10:06:48 2011

Modification date

9.5 bfast_db

:: Generate db index files for aligning reads with bfast

9.5.1 Commands

clean Remove all job data, not the Moa job itself

run run bfast fasta2brg and index commands

9.5.2 Filesets

fa_input::

fasta input file

type: *map*

source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.5.3 Parameters

algorithm_colour_space::

true -> colour space, false -> NT space

type: *boolean*
default: *False*
optional: *True*

depth::

The depth of the splitting(d). The index will be split into 4^d parts.

type: *integer*
default: *0*
optional: *True*

extra_params::

Any extra parameters

type: *string*
default: ‘‘
optional: *True*

hash_width::

The hash width for the index (recommended from manual = 14)

type: *integer*
default: *{}*
optional: *False*

index_num::

Specifies this is the ith index you are creating

type: *integer*
default: *1*
optional: *True*

mask::

The mask or spaced seed to use.

type: *string*
default: *{}*
optional: *False*

print_params::

print program parameters

type: *boolean*
default: *False*
optional: *True*

thread_num::

Specifies the number of threads to use

type: *integer*
default: *1*
optional: *True*

timing_information::

specifies output timing information

type: *boolean*
default: *True*
optional: *True*

usage_summary::

Display usage summary (help)

type: *boolean*
default: *False*
optional: *True*

9.5.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Feb 15 10:06:48 2011

Modification date

9.6 bidibebla

Bidirectional best BLAST hit

:: Discover the bidirectional best blast hit between two sets of sequences

9.6.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run generate a list of bidirectional best blast hits

9.6.2 Filesets

input::

Fasta input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

Lists of bidirectional best blasts

type: *map*
source: *input*
category: *output*

optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

reference::

Reference fasta file to compare against

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.6.3 Parameters

eval::

e value cutoff

type: *float*
default: *1e-10*
optional: *True*

nothreads::

Threads to run blast with with

type: *integer*
default: *4*
optional: *True*

protein::

Is this a protein set

type: *boolean*
default: *F*
optional: *True*

9.6.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.7 blast

Basic Local Alignment Tool

:: Wraps BLAST [[Alt90]], probably the most popular similarity search tool in bioinformatics.

9.7.1 Commands

blast_report Generate a text BLAST report.

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run 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 to the standard text based output) in the *./out* directory. The output XML is subsequently converted to GFF3 by the custom *blast2gff* script (using BioPython). Additionally, a simple text report is created.

9.7.2 Parameters

blast_gff_blasthit::

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

type: *set*
default: *F*
optional: *{}*

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

type: *file*
default: ‘‘
optional: *True*

eval::

e value cutoff

type: *float*
default: *1e-10*
optional: *True*

gff_source::

source field to use in the gff

type: *string*
default: *BLAST*
optional: *True*

input_dir::

Directory with the BLAST input files

type: *directory*
default: ‘‘
optional: *True*

input_extension::

file extension for the files in blast_input_dir

type: *string*
default: *fasta*
optional: *True*

input_file::

Input fasta file to BLAST

type: *file*
default: ‘‘
optional: *True*

input_glob::

glob to select a subset of files from blast_input_dir

type: *string*
default: *
optional: *True*

input_limit::

Number of files to use, if not defined: all files

type: *integer*
default: ‘‘
optional: *True*

input_sort::

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

type: *set*
default: *u*
optional: *True*

nohits::

number of hits to report

type: *integer*
default: *50*
optional: *True*

nothreads::

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

type: *integer*
default: *2*
optional: *True*

program::

blast program to use (default: blastn)

type: *set*

default: *blastn*
optional: *True*

9.7.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.8 blast

Basic Local Alignment Tool

:: Wraps BLAST [[Alt90]], probably the most popular similarity search tool in bioinformatics.

9.8.1 Commands

blast_report Generate a text BLAST report.

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run 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 to the standard text based output) in the *./out* directory. The output XML is subsequently converted to GFF3 by the custom *blast2gff* script (using BioPython). Additionally, a simple text report is created.

9.8.2 Parameters

blast_gff_blasthit::

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

type: *set*
default: *F*
optional: *{}*

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

type: *file*
default: ‘‘
optional: *True*

eval::

e value cutoff

type: *float*
default: *1e-10*
optional: *True*

gff_source::

source field to use in the gff

type: *string*
default: *BLAST*
optional: *True*

input_dir::

Directory with the BLAST input files

type: *directory*
default: ‘‘
optional: *True*

input_extension::

file extension for the files in blast_input_dir

type: *string*
default: *fasta*
optional: *True*

input_file::

Input fasta file to BLAST

type: *file*
default: ‘‘

optional: *True*

input_glob::

glob to select a subset of files from blast_input_dir

type: *string*
default: *
optional: *True*

input_limit::

Number of files to use, if not defined: all files

type: *integer*
default: ‘‘
optional: *True*

input_sort::

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

type: *set*
default: *u*
optional: *True*

nohits::

number of hits to report

type: *integer*
default: *50*
optional: *True*

nothreads::

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

type: *integer*
default: *2*
optional: *True*

program::

blast program to use (default: blastn)

type: *set*
default: *blastn*
optional: *True*

9.8.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.9 blastdb

9.9.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

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

9.9.2 Parameters

fasta_file::

The file with all input FASTA sequences for the blastdb.

type: *file*
default: *{}*
optional: *False*

name::

Name of the BLAST database to create.

type: *string*
default: *blastdb*
optional: *True*

protein::

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

type: *set*
default: *F*
optional: *True*

9.9.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.10 blat

Blat

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

9.10.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.10.2 Parameters

db::

type of the database (dna, prot or dnax)

type: *set*
default: ‘‘
optional: *False*

db_id_list::

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

type: *file*
default: ‘‘
optional: *True*

db_type::

type of the database (dna, prot or dnax)

type: *set*
default: *dna*
optional: *True*

eval::

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

type: *float*
default: *1e-10*
optional: *True*

gff_source::

Source field for the generated GFF files

type: *string*
default: ‘‘
optional: *False*

input_dir::

source field in the generated gff

type: *directory*
default: ‘‘
optional: *False*

input_extension::

extension of the input files

type: *string*
default: *fasta*
optional: *True*

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

type: *file*
default: ‘‘
optional: *False*

query_type::

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

type: *set*
default: *dna*
optional: *True*

9.10.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.11 bowtie

Bowtie

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

9.11.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template

run *no help defined*

9.11.2 Filesets

input::

Fasta/fastq input files for bowtie

type: *map*
source: *{}*
category: *input*

optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

Output files

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.11.3 Parameters

db::

The (basename of the) bowtie database to use.

type: *string*
default: *{}*
optional: *False*

extra_params::

extra parameters to feed bowtie

type: *string*
default: ‘‘
optional: *True*

input_format::

Format of the input files

type: *set*
default: *fastq*
optional: *True*

9.11.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.12 bowtie_pe

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

9.12.1 Commands

clean Remove all job data, not the Moa job itself

report Create a report on the results

run *no help defined*

run2 Execute bowtie in paired-end mode

9.12.2 Filesets

db::

The (basename of the) bowtie database to use.

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_forward_input::

fastq input files directory - forward

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*

dir: *{}*

fq_reverse_input::

fastq input files directory - reverse

type: *map*
source: *fq_forward_input*
category: *input*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

output::

Bam output file

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.12.3 Parameters

extra_params::

extra parameters to feed to bowtie

type: *string*
default: ‘‘
optional: *True*

input_format::

Format of the input files

type: *set*
default: *fastq*
optional: *True*

max_insertsize::

Maximum allowed insertsize

type: *integer*
default: *250*
optional: *True*

min_insertsize::

Minimum allowed insertsize

type: *integer*
default: *1*
optional: *True*

output_format::

Format of the output file

type: *set*
default: *bam*
optional: *True*

9.12.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.13 bowtie_se

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

9.13.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template

run *no help defined*

9.13.2 Filesets

fq_input::

fastq input files directory

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

Bam output file

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.13.3 Parameters

ebwt_base::

The (basename of the) bowtie database to use.

type: *string*
default: *{}*
optional: *False*

extra_params::

extra parameters to feed to bowtie

type: *string*
default: ‘‘
optional: *True*

input_format::

Format of the input files

type: *set*
default: *fastq*
optional: *True*

output_format::

Format of the output file

type: *set*
default: *bam*
optional: *True*

9.13.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.14 bowtiedb

Bowtie index builder

:: Builds a bowtie index from a reference sequence

9.14.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Create the bowtie database

9.14.2 Filesets

input::

Input fasta file for the bowtie database

type: *map*
source: *{}*
category: *input*

optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

database name to create

type: *map*
source: *{}*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.14.3 Parameters

extra_params::

any option parameters

type: *string*
default: ‘‘
optional: *True*

9.14.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Dec 09 07:56:48 2010

9.15 bwa_aln

:: Use BWA to align a set of fastq reads against a db

9.15.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run bwa aln

9.15.2 Filesets

input::

Fastq input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.15.3 Parameters

best_hits_stop::

stop searching when there are >INT equally best hits

type: *integer*
default: *30*
optional: *True*

color_space::

input sequences are in the color space

type: *boolean*
default: *False*
optional: *True*

db::

bwa database to align against

type: *string*
default: ‘‘
optional: *False*

edit_dist_missing_prob::

max

type: *integer*
default: *0.04*
optional: *True*

gap_ext_max::

maximum number of gap extensions, -1 for disabling long gaps

type: *integer*
default: *-1*
optional: *True*

gap_ext_penalty::

gap extension penalty

type: *integer*
default: *4*
optional: *True*

gap_open_penalty::

gap open penalty

type: *integer*
default: *11*
optional: *True*

gap_opens_max::

maximum number or fraction of gap opens

type: *integer*
default: *1*
optional: *True*

log_gap_penalty_del::

log-scaled gap penalty for long deletions

type: *boolean*
default: *False*
optional: *True*

max_ext_long_del::

maximum occurrences for extending a long deletion

type: *integer*
default: *10*
optional: *True*

max_queue_entry::

maximum entries in the queue

type: *integer*
default: *2000000*
optional: *True*

mismatch_penalty::

mismatch penalty

type: *integer*
default: *3*
optional: *True*

no_indel_from_ends::

do not put an indel within INT bp towards the ends

type: *integer*
default: *5*

optional: *True*

non_iterative::

non-iterative mode search for all n-difference hits (slow)

type: *boolean*
default: *False*
optional: *True*

quality_step::

quality threshold for read trimming down to 35bp

type: *integer*
default: *0*
optional: *True*

seed_len::

Seed length

type: *integer*
default: *30*
optional: *True*

seed_max_diff::

Maximum differences in the seed

type: *integer*
default: *2*
optional: *True*

thread_num::

number of threads

type: *integer*
default: *1*
optional: *True*

9.15.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.16 bwa_index

Bwa index builder

:: Builds a bwa index from a reference sequence

9.16.1 Commands

clean Remove all job data

run Create the index

9.16.2 Parameters

algorithm::

Algorithm for constructing BWT index. Available options are 'is' and 'bwtsw'

type: *string*
default: *is*
optional: *True*

color_space::

input sequences are in the color space

type: *boolean*
default: *False*
optional: *True*

input_fasta::

input fasta file for the database

type: *file*
default: *{}*
optional: *False*

prefix::

Name of the bwa index to create

type: *string*
default: *{}*
optional: *False*

9.16.3 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.17 bwa_sampe

:: Generate alignments in SAM format given paired end reads

9.17.1 Commands

clean Remove all job data, not the Moa job itself

run run bwa sampe

9.17.2 Filesets

fq_forward_input::

fastq input files directory - forward

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_reverse_input::

fastq input files directory - reverse

type: *map*
source: *fq_forward_input*
category: *input*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

output_bam::

{}

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

sai_forward_input::

sai input files - forward

type: *map*
source: *fq_forward_input*
category: *input*
optional: *False*
extension: *sai*
glob: *{}*
dir: *{}*

sai_reverse_input::

sai input files - reverse files

type: *map*
source: *sai_forward_input*
category: *input*
optional: *True*
extension: *sai*
glob: *{}*
dir: *{}*

9.17.3 Parameters

db::

bwa database to align against

type: *string*
default: ‘‘
optional: *False*

disable_insert_size::

disable insert size estimate (force -s)

type: *boolean*
default: *False*
optional: *True*

disable_SW::

disable Smith-Waterman for the unmapped mate

type: *boolean*
default: *False*
optional: *True*

max_aln_out::

maximum hits to output for paired reads

type: *integer*
default: *3*
optional: *True*

max_insert_size::

maximum insert size

type: *integer*
default: *500*
optional: *True*

max_occ_read::

maximum occurrences for one end

type: *integer*
default: *100000*
optional: *True*

max_out_discordant_pairs::

maximum hits to output for discordant pairs

type: *integer*
default: *10*
optional: *True*

preload_index::

preload index into memory (for base-space reads only)

type: *boolean*
default: *False*
optional: *True*

prior_chimeric_rate::

prior of chimeric rate (lower bound)

type: *integer*
default: *1e-05*
optional: *True*

9.17.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Nov 25 17:06:48 2010

Modification date

9.18 bwa_samse

:: Generate alignments in SAM format given single end reads, using both 'bwa samse'.

9.18.1 Commands

clean Remove all job data, not the Moa job itself

run run bwa samse

9.18.2 Filesets

fq_input::

fastq input file

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output_bam::

output bam file

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

sai_input::

sai input directory - filenames must correspond to the fastq input files

type: *map*
source: *fq_input*
category: *input*
optional: *False*

extension: *{}*
glob: *{}*
dir: *{}*

9.18.3 Parameters

db::

bwa database to align against

type: *string*
default: ‘‘
optional: *False*

max_aln_out::

Maximum number of alignments to output in the XA tag for reads paired properly

type: *integer*
default: *3*
optional: *True*

9.18.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Nov 25 17:06:48 2010

Modification date

9.19 clean_fasta

clean Fasta

:: Convert files to unix format and convert all characters that are not an A,C,G,T or N to N.

9.19.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Cleanup of a FASTA file (in place!)

9.19.2 Parameters

cf_input_dir::

Directory with the sequences to run cleanfasta on

type: *directory*
default: ‘‘
optional: *False*

cf_input_extension::

input file extension

type: *string*
default: *fasta*
optional: *True*

sed_command::

{}

type: *string*
default: */^>/!s/[^ACGTNacgtn]/N/g*
optional: *True*

9.19.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.20 clustalgroup

clustalw

:: Run clustalw on two sets of sequences

9.20.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run clustalw

9.20.2 Parameters

cwg_input_dir::

This set of sequences to run clustalw on

type: *directory*
default: ‘‘
optional: *False*

cwg_input_extension::

Input file extension

type: *string*
default: *fasta*
optional: *True*

9.20.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.21 clustalpair

clustalw

:: Run clustalw on two sets of sequences

9.21.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run clustalw

9.21.2 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)

type: *directory*
default: ‘‘
optional: *False*

input_dir_b::

The set to compare against

type: *directory*
default: ‘‘
optional: *False*

input_extension::

Extension of the input files

type: *string*
default: *fasta*
optional: *True*

9.21.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.22 clustalw

clustalw

:: Run clustalw on two sets of sequences

9.22.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run clustalw

9.22.2 Parameters

input_dir_a::

This set is compared to the sequences in input_dir_b.

type: *directory*
default: ‘‘
optional: *False*

input_dir_b::

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

type: *directory*
default: ‘‘
optional: *False*

input_extension::

Extension of the input files

type: *string*
default: *fasta*
optional: *True*

9.22.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.23 concatenate

Concatenate

:: Concatenate a set of fasta files into one.

9.23.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.23.2 Parameters

input_dir::

Directory with the input data

type: *directory*
default: ‘‘
optional: *False*

input_extension::

Extension of the input files

type: *string*
default: *fasta*
optional: *True*

name::

name of the file, the outputfile will become ./name.fasta

type: *string*
default: ‘‘
optional: *False*

9.23.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.24 dotself

Dottup self

:: Run dottup with a sequence against itself

9.24.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run clustalw

9.24.2 Parameters

input_dir::

Set of sequences to use

type: *directory*
default: ‘‘
optional: *False*

input_extension::

Extension of input files

type: *string*
default: *fasta*
optional: *True*

wordsize::

Wordsize used for recognizing similarity

type: *integer*
default: *6*
optional: *True*

9.24.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.25 dottup

EMBOSS Dottup

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

9.25.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Run dottup

9.25.2 Parameters

input_dir_a::

This set is compared to the sequences in input_dir_b.

type: *directory*
default: ‘‘
optional: *False*

input_dir_b::

The set to compare against

type: *directory*
default: ‘‘
optional: *True*

input_extension::

Extension of the dottup input files

type: *string*
default: *fasta*
optional: *True*

wordsize::

Wordsize used to discover similarities between sequences

type: *integer*
default: *8*

optional: *True*

9.25.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.26 empty

empty

:: Do nothing...

9.26.1 Commands

9.26.2 Parameters

9.26.3 Other

Backend ruff

Author Mark Fiers

Creation date Mon Apr 04 16:02:58 2011

Modification date Mon Apr 04 16:03:18 2011

9.27 f2b

Convert fasta to bfa

:: Converts a FASTA file to MAQ format for use with a BFA a maq_fasta2bfa index from a reference sequence

9.27.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.27.2 Filesets

bfa::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

input FASTA files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.27.3 Parameters

9.27.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.28 fasta2gff

GFF from FASTA

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

9.28.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.28.2 Parameters

f2g_gffsource::

Source to be used in the gff

type: *string*
default: ‘‘
optional: *False*

f2g_input_dir::

Directory with the input fasta files

type: *directory*
default: ‘‘
optional: *False*

f2g_input_extension::

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

type: *string*
default: *fasta*
optional: *True*

f2g_options::

options to be passed to the fasta2gff script

type: *string*
default: ‘‘
optional: *True*

f2g_output_dir::

Directory with the output gff

type: *directory*
default: *./gff*
optional: *True*

9.28.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.29 fastqc

Run FastQC for fastq QC

:: Run FastQC on a set a fastq files - quality assessment

9.29.1 Commands

report Generate a simple fastqc report

run *no help defined*

run2 Run Fastqc

9.29.2 Filesets

input::

fastqc input files'

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

touch files - track if a file has been processed - do not touch this unless you know what you're doing.

type: *map*
source: *input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.29.3 Parameters

output_dir::

output directory for the fastQC report

type: *dir*
default: .
optional: *True*

9.29.4 Other

Backend ruff

Author Mark Fiers

Creation date Thu, 28 Apr 2011 09:27:17 +1200

Modification date Thu, 28 Apr 2011 14:19:04 +1200

9.30 fastx_clipper

:: run fastx_clipper

9.30.1 Commands

clean Remove all job data, not the Moa job itself

run run fastx_clipper

9.30.2 Filesets

input::

fastq input files directory

type: *map*
source: *{}*

category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.30.3 Parameters

adaptor::

ADAPTER string. default is CCTTAAGG (dummy adapter).

type: *string*
default: *CCTTAAGG*
optional: *True*

adaptor_and_bases::

Keep the adapter and N bases after it.

type: *integer*
default: *0*
optional: *True*

compress_output::

Compress output with GZIP.

type: *boolean*
default: *False*
optional: *True*

debug_output::

DEBUG output.

type: *boolean*
default: *False*
optional: *True*

help::

help screen

type: *boolean*
default: *False*
optional: *True*

keep_unknown_nuc_seq::

keep sequences with unknown (N) nucleotides. default is to discard such sequences.

type: *boolean*
default: *False*
optional: *True*

out_adaptor_only_seq::

Report Adapter-Only sequences.

type: *boolean*
default: *False*
optional: *True*

rm_clipped_seq::

Discard clipped sequences (i.e. - keep only sequences which did not contained the adapter).

type: *boolean*
default: *False*
optional: *True*

rm_non_clipped_seq::

Discard non-clipped sequences (i.e. - keep only sequences which contained the adapter).

type: *boolean*
default: *False*
optional: *True*

rm_short_seq::

discard sequences shorter than N nucleotides. default is 5.

type: *integer*
default: *5*
optional: *True*

verbose::

Verbose - report number of sequences. If [-o] is specified, report will be printed to STDOUT. If [-o] is not specified (and output goes to STDOUT), report will be printed to STDERR.

type: *boolean*
default: *False*
optional: *True*

9.30.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Dec 06 17:06:48 2010

Modification date

9.31 fastx_qual_stats

:: run fastx_quality_stats, fastq_quality_boxplot_graph.sh and fastx_nucleotide_distribution_graph.sh

9.31.1 Commands

clean Remove all job data, not the Moa job itself

run run fastx_quality_stats, fastq_quality_boxplot_graph.sh and fastx_nucleotide_distribution_graph.sh

9.31.2 Filesets

boxplot_output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

fastq input files directory

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

nuc_distr_output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

qual_output::

{}

type: *map*

source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.31.3 Parameters

gen_postScript_file::

Generate PostScript (.PS) file. Default is PNG image.

type: *boolean*
default: *False*
optional: *True*

graph_title::

Title - will be plotted on the graph.

type: *string*
default: *{{ input_glob }}*
optional: *True*

help::

help screen

type: *boolean*
default: *False*
optional: *True*

new_out_format::

New output format (with more information per nucleotide/cycle)

type: *boolean*
default: *False*
optional: *True*

9.31.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Dec 03 17:06:48 2010

Modification date

9.32 fq2bq

Convert FASTQ to BFQ

:: Converts a FASTQ file to MAQ BFQ format.

9.32.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.32.2 Filesets

bfq::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

input FASTA files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.32.3 Parameters

9.32.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.33 gather

gather files

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

9.33.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run gather files

9.33.2 Parameters

g_input_dir::

list of directories with the input files

type: *directory*
default: ‘‘
optional: *False*

g_input_pattern::

glob pattern to download

type: *string*
default: *
optional: *True*

g_limit::

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

type: *integer*
default: *1000000*
optional: *True*

g_name_sed::

SED expression to be executed on each file name - allows you to change file names

type: *string*
default: *s/a/a/*
optional: *True*

g_output_dir::

Output subdirectory, defaults to .

type: *directory*
default: .
optional: *True*

g_parallel::

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

type: *set*
default: *F*
optional: *True*

g_powerclean::

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

type: *set*
default: *F*
optional: *True*

g_process::

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

type: *string*
default: *ln -f $$< $$(g_target)*
optional: *True*

9.33.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.34 genemarks

geneMarkS

:: predict genes using geneMarkS

9.34.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.34.2 Filesets

input::

Directory with the input files for Genemarks

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.34.3 Parameters

gff_source::

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

type: *string*

default: *genemarkS*
optional: *True*

matrix::

the matrix to use

type: *file*
default: ‘‘
optional: *True*

9.34.4 Other

Backend ruff

Author

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.35 getorf

Revseq

:: This Moa template takes a set of input FASTA sequences and determines the reverse complement using the EMBOSS revseq utility.

9.35.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.35.2 Filesets

gff::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

Input files for getorf

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.35.3 Parameters

circular::

Is the sequence linear?

type: *set*
default: *N*
optional: *True*

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

type: *set*
default: *3*
optional: *True*

gff_source::

source field to use in the gff.

type: *string*
default: *getorf*
optional: *True*

maxsize::

maximal nucleotide size of the predicted ORF.

type: *integer*
default: *1000000*
optional: *True*

minsize::

minimal nucleotide size of the predicted ORF.

type: *integer*
default: *30*
optional: *True*

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.

type: *set*
default: *11*
optional: *True*

9.35.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.36 getorf

Revseq

:: This Moa template takes a set of input FASTA sequences and determines the reverse complement using the EMBOSS revseq utility.

9.36.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.36.2 Filesets

gff::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

Input files for getorf

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.36.3 Parameters

circular::

Is the sequence linear?

type: *set*
default: *N*
optional: *True*

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

type: *set*
default: *3*
optional: *True*

gff_source::

source field to use in the gff.

type: *string*
default: *getorf*
optional: *True*

maxsize::

maximal nucleotide size of the predicted ORF.

type: *integer*
default: *1000000*
optional: *True*

minsize::

minimal nucleotide size of the predicted ORF.

type: *integer*
default: *30*
optional: *True*

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.

type: *set*
default: *11*
optional: *True*

9.36.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.37 glimmer3

Glimmer3

:: Predicts (prokaryotic) using glimmer3.

9.37.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Glimmer3 is a open reading frame discovery program from the EMBOSS [[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.

9.37.2 Filesets

input::

Directory with the input files for Glimmer3

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.37.3 Parameters

gene_len::

Minimum gene length (glimmer3 -g/–gene_len)

type: *integer*
default: *110*
optional: *True*

gff_source::

source field to use in the gff. Defaults to “glimmer3”

type: *string*
default: *glimmer3*
optional: *True*

max_overlap::

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

type: *integer*
default: *50*
optional: *True*

treshold::

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

type: *integer*
default: *30*
optional: *True*

9.37.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.38 gmap

Gmap

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

9.38.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.38.2 Filesets

align::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

genepred::

{}

type: *map*
source: *input*
category: *output*

optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

gff::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

gff_invert::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

Sequences to map

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

raw::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.38.3 Parameters

db::

Gmap db

type: *file*
default: ‘‘
optional: *False*

extra_parameters::

extra parameters to feed to gmap

type: *string*
default: ‘‘
optional: *True*

gff_source::

Source field to use in the output GFF

type: *string*
default: *gmap*
optional: *True*

invert_gff::

Invert the GFF (T/*F*)

type: *set*
default: *T*

optional: *True*

9.38.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.39 gmapdb

gmapdb index builder

:: Builds gmapdb index from a reference sequence

9.39.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.39.2 Filesets

input::

The reference sequence to build a gmap database with.

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.39.3 Parameters

name::

Name of the gmap index to create

type: *string*
default: *gmapdb*
optional: *True*

9.39.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.40 gsmap

GSMapper

:: Run the Roche GS Reference mapper

9.40.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.40.2 Parameters

annotation::

Gene annotation file in the UCSC GenePred format

type: *file*
default: ‘‘
optional: *True*

min_overlap_ident::

Minimum identity length in the assembly step

type: *integer*
default: *90*
optional: *True*

min_overlap_len::

Minimum overlap length in the assembly step

type: *integer*
default: *40*
optional: *True*

name::

Name identifying this mapping in the output gff

type: *string*
default: ‘‘
optional: *False*

reference_fasta::

A multifasta file with the reference sequence(s)with the library id.

type: *file*
default: ‘‘
optional: *True*

sfffile::

SFF files with reads to map against the reference sequences

type: *file*
default: ‘‘
optional: *True*

9.40.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.41 h_blast

Hadoop Blast

:: Runs BLAST on a hadoop cluster

9.41.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

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

9.41.2 Parameters

db::

Location of the blast database

type: *file*
default: ‘‘
optional: *False*

eval::

e value cutoff

type: *float*
default: *1e-10*
optional: *True*

hadoop_base::

location of the hadoop installation

type: *directory*
default: ‘‘
optional: *False*

hdfs_base::

htfs://SERVER:PORT for the hdfs filesystem, defaults to “hdfs://localhost:9000”

type: *string*
default: *hdfs://localhost:9000*
optional: *True*

input_dir::

location of the hadoop installation

type: *directory*
default: ‘‘
optional: *False*

input_extension::

input file extension

type: *string*
default: *fasta*
optional: *True*

nohits::

number of hits to report

type: *integer*
default: *50*
optional: *True*

nothreads::

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

type: *integer*
default: *1*
optional: *True*

program::

blast program to use (default: blastn)

type: *set*
default: *blastn*
optional: *True*

9.41.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.42 kanga

:: use kanga to align short reads to a reference genome

9.42.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run kanga

9.42.2 Filesets

input_fasta::

Fasta input file

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

output files

type: *map*
source: *rds_input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

output_bam::

output files

type: *map*
source: *rds_input*
category: *output*
optional: *True*

extension: *{}*
glob: *{}*
dir: *{}*

output_log::

output log file

type: *map*
source: *rds_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

rds_input::

rds (preprocessed) input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

sfx_input::

sfx array lookup file

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.42.3 Parameters

color_space::

process for colorspace (SOLiD)

type: *boolean*
default: *False*
optional: *True*

extra_params::

any extra parameters

type: *string*
default: ‘‘
optional: *True*

help::

print this help and exit

type: *boolean*
default: *False*
optional: *True*

max_Ns::

maximum number of intermediate N’s in reads before treating read as unalignable

type: *integer*
default: *1*
optional: *True*

max_pair_len::

accept paired end alignments with apparent length of at most this

type: *integer*
default: *300*
optional: *True*

min_pair_len::

accept paired end alignments with apparent length of at least this

type: *integer*
default: *100*
optional: *True*

no_multireads::

do not accept multiple reads aligning to the same loci

type: *boolean*
default: *False*
optional: *True*

out_format::

0 - CSV loci only, 1 - CSV loci + match sequence, 2 - CSV loci + read sequence, 3 - CSV loci + read + match sequence, 4 - UCSC BED, 5 - SAM format

type: *integer*
default: *0*
optional: *True*

pe_mode::

0 - none, 1 - paired ends with recover orphan ends, 2 - paired end no orphan recovery

type: *integer*
default: *0*
optional: *True*

quality::

fastq quality scoring- 0 - sanger, 1m - Illumina 1.3+, 2 - Solexa < 1.3, 3 - Ignore quality

type: *integer*
default: *3*
optional: *True*

thread_num::

number of processing threads (0 sets threads to number of CPU cores)

type: *integer*

default: *0*
optional: *True*

trim3::

trim this number of bases from 3' end of reads when loading raw reads

type: *integer*
default: *0*
optional: *True*

trim5::

trim this number of bases from 5' end of reads when loading raw reads

type: *integer*
default: *0*
optional: *True*

version::

print version information and exit

type: *boolean*
default: *False*
optional: *True*

9.42.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.43 kangar_pe

:: use kangar to pre process raw fq reads

9.43.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run kangar

9.43.2 Filesets

fq_forward_input::

fastq input files - forward - containing the 5' end

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_reverse_input::

fastq input files directory - reverse - containing the 3' end

type: *map*
source: *fq_forward_input*
category: *input*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

output_log::

output log file

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

rds_output::

output rds file

type: *map*
source: *fq_forward_input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.43.3 Parameters

extra_params::

any extra parameters

type: *string*
default: ‘‘
optional: *True*

help::

print this help and exit

type: *boolean*
default: *False*
optional: *True*

mode::

processing mode 0 - single end create, 1 - paired end create, 2 - output statistics 3 - dump as fasta

type: *integer*
default: *0*
optional: *True*

quality::

fastq quality scoring- 0 - sanger, 1m - Illumina 1.3+, 2 - Solexa < 1.3, 3 - Ignore quality

type: *integer*
default: *3*
optional: *True*

reads_num::

limit number of reads (or dumps) in each input file to this many, 0 if no limit

type: *integer*
default: *0*
optional: *True*

rm_duplicates::

remove duplicate reads retaining only one

type: *boolean*
default: *False*
optional: *True*

trim3::

trim this number of bases from 3' end of sequence

type: *integer*
default: *0*
optional: *True*

trim5::

trim this number of bases from 5' end of sequence

type: *integer*
default: *0*
optional: *True*

version::

print version information and exit

type: *boolean*
default: *False*

optional: *True*

9.43.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.44 kangar_se

:: use kangar to pre process raw fq single end reads

9.44.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run kangar

9.44.2 Filesets

fq_input::

fastq input files - forward - containing the 5' end

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output_log::

output log file

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*

dir: *{}*

rds_output::

output rds file

type: *map*
source: *fq_input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.44.3 Parameters

extra_params::

any extra parameters

type: *string*
default: ‘‘
optional: *True*

help::

print this help and exit

type: *boolean*
default: *False*
optional: *True*

mode::

processing mode 0 - single end create, 1 - paired end create, 2 - output statistics 3 - dump as fasta

type: *integer*
default: *0*
optional: *True*

quality::

fastq quality scoring- 0 - sanger, 1m - Illumina 1.3+, 2 - Solexa < 1.3, 3 - Ignore quality

type: *integer*
default: *3*
optional: *True*

reads_num::

limit number of reads (or dumps) in each input file to this many, 0 if no limit

type: *integer*
default: *0*
optional: *True*

rm_duplicates::

remove duplicate reads retaining only one

type: *boolean*
default: *False*
optional: *True*

trim3::

trim this number of bases from 3' end of sequence

type: *integer*
default: *0*
optional: *True*

trim5::

trim this number of bases from 5' end of sequence

type: *integer*
default: *0*
optional: *True*

version::

print version information and exit

type: *boolean*
default: *False*
optional: *True*

9.44.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.45 kangax

:: use kangax to create the suffix array lookup database for the reference genome

9.45.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run run kangax

9.45.2 Filesets

input_fasta::

Fasta input file

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output_log::

output log file

type: *map*
source: *input_fasta*
category: *output*
optional: *{}*

extension: *{}*
glob: *{}*
dir: *{}*

output_sfx::

output suffix array lookup

type: *map*
source: *input_fasta*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.45.3 Parameters

block_seq_len::

generated suffix blocks to hold at most this length (MB) concatenated sequences

type: *integer*
default: *3300*
optional: *True*

color_space::

generate for colorspace (SOLiD)

type: *boolean*
default: *False*
optional: *True*

extra_params::

any extra parameters

type: *string*
default: ‘‘
optional: *True*

help::

print this help and exit

type: *boolean*
default: *False*
optional: *True*

reference_species::

reference species

type: *string*
default: ‘‘
optional: *False*

target_dep::

generate target file only if missing or older than any independent source files

type: *boolean*
default: *False*
optional: *True*

version::

print version information and exit

type: *boolean*
default: *False*
optional: *True*

9.45.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date

9.46 lftp

lftp

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

9.46.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run execute the download

9.46.2 Parameters

dos2unix::

Run dos2unix to prevent problems with possible dos text files

type: *set*
default: *F*
optional: *True*

get_name::

target name of the file to download

type: *string*
default: ‘‘
optional: *True*

lftp_output_dir::

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

type: *directory*
default: .
optional: *{}*

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

type: *set*
default: *T*
optional: *True*

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.

type: *set*
default: *get*
optional: *True*

noclean::

set of files not to be deleted by the powerclean

type: *string*
default: *moa.mk Makefile*
optional: *True*

pass::

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

type: *password*
default: ‘‘
optional: *True*

pattern::

glob pattern to download

type: *string*
default: *‘*’
optional: *True*

powerclean::

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

type: *set*
default: *F*
optional: *True*

timestamp::

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

type: *set*
default: *F*
optional: *True*

url::

The base url to download from

type: *string*
default: ‘‘
optional: *True*

user::

username for the remote site

type: *string*
default: ‘‘
optional: *True*

9.46.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.47 map

Execute a "map" ad-hoc analysis

:: Execute one command, on a number of input files.

9.47.1 Commands

run *no help defined*

9.47.2 Filesets

input::

"map" input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

"map" output files

type: *map*
source: *input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.47.3 Parameters

process::

The command to execute

type: *string*
default: *True*

optional: *False*

9.47.4 Other

Backend ruff

Author Mark Fiers

Creation date Tue Mar 29 16:34:19 2011

Modification date Wed Mar 30 06:02:01 2011

9.48 maq_pe

:: Generate alignments in SAM format given paired end reads using Maq.

9.48.1 Commands

clean Remove all job data, not the Moa job itself

run run maq's fasta2bfa, fastq2bfq and map.

9.48.2 Filesets

bam_output::

bam alignment output file

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

bfa_output::

BFA Index name

type: *map*
source: *{}*
category: *other*
optional: *{}*
extension: *{}*
glob: *{}*

dir: *{}*

bfq_forward_output::

bfq files - forward files

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

bfq_reverse_output::

bfq files - reverse files

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

fa_input::

directory with reference fasta file name

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_forward_input::

fastq input files directory - forward files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_reverse_input::

fastq input files directory - reverse files

type: *map*
source: *fq_forward_input*
category: *input*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

map_output::

maq map output files

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.48.3 Parameters

disable_sw::

disable Smith-Waterman alignment

type: *boolean*
default: *False*
optional: *True*

extra_parameters::

Any extra parameters

type: *string*
default: ‘‘
optional: *True*

first_read_len::

length of the first read (<=127)s

type: *integer*
default: *0*
optional: *True*

match_in_colorspace::

match in the colorspace

type: *boolean*
default: *False*
optional: *True*

max_dist_read_pairs::

max distance between two paired reads s

type: *integer*
default: *250*
optional: *True*

max_dist_RF_read_pairs::

max distance between two RF paired reads s

type: *integer*
default: *0*
optional: *True*

max_mismatch_qual_sum::

maximum allowed sum of qualities of mismatches

type: *integer*
default: *70*
optional: *True*

max_num_hits_out::

max number of hits to output. >512 for all 01 hits.

type: *integer*
default: *250*
optional: *True*

num_mismatch_24bp::

number of mismatches in the first 24bp

type: *integer*
default: *2*
optional: *True*

read_ref_diff_rate::

rate of difference between reads and references

type: *float*
default: *0.001*
optional: *True*

sec_read_len::

length of the second read (<=127)s

type: *integer*
default: *0*
optional: *True*

trim_all_reads::

trim all reads (usually not recommended)

type: *boolean*
default: *False*

optional: *True*

9.48.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Dec 03 17:06:48 2010

Modification date

9.49 maq_se

:: Generate alignments in SAM format given single end reads using Maq.

9.49.1 Commands

clean Remove all job data, not the Moa job itself

run run maq's fasta2bfa, fastq2bfq and map.

9.49.2 Filesets

bam_output::

bam alignment output file

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

bfa_output::

BFA Index name

type: *map*
source: *{}*
category: *other*
optional: *{}*
extension: *{}*
glob: *{}*

dir: *{}*

bfq_output::

bfq files - forward files

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

fa_input::

directory with reference fasta file name

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_input::

fastq input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

map_output::

maq map output files

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.49.3 Parameters

disable_sw::

disable Smith-Waterman alignment

type: *boolean*
default: *False*
optional: *True*

extra_parameters::

other parameters

type: *string*
default: ‘‘
optional: *True*

match_in_colorspace::

match in the colorspace

type: *boolean*
default: *False*
optional: *True*

max_mismatch_qual_sum::

maximum allowed sum of qualities of mismatches

type: *integer*
default: *70*
optional: *True*

max_num_hits_out::

number of mismatches in the first 24bp

type: *integer*
default: *250*
optional: *True*

num_mismatch_24bp::

number of mismatches in the first 24bp

type: *integer*
default: *2*
optional: *True*

read_ref_diff_rate::

rate of difference between reads and references

type: *float*
default: *0.001*
optional: *True*

trim_all_reads::

trim all reads (usually not recommended)

type: *boolean*
default: *False*
optional: *True*

9.49.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Dec 02 17:06:48 2010

Modification date

9.50 maqpair

MAQ paired ends mapper

:: Map paired ends to a reference sequence using MAQ

9.50.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.50.2 Parameters

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

type: *string*
default: *_f.bfq*
optional: *True*

maxdist::

max outer distance for a (non RF) readpair. This applies to illumina matepairs - i.e. short inserts

type: *integer*
default: *250*
optional: *True*

read_dir::

directory containing the forward reads

type: *string*
default: ‘‘
optional: *False*

reference::

Reference bfa file to map the reads to

type: *string*

default: ‘‘
optional: *False*

reverse_suffix::

suffix of reverse files

type: *string*
default: *_r.bfq*
optional: *True*

RF_maxdist::

max outer distance for an RF readpair (corresponds to the -A parameter). This applies to long insert illumina pairs

type: *integer*
default: *15000*
optional: *True*

9.50.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.51 moatest

Unittest template

:: Not to be used - is used by unitmoatests

9.51.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Do nothing - no need to call this.

9.51.2 Parameters

test_opt::

test variable

type: *string*
default: *konijntje*
optional: *True*

txt::

test variable

type: *string*
default: ‘‘
optional: *False*

9.51.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.52 mummer

mummer

:: Run mummer between two sequences

9.52.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.52.2 Filesets

mum_input_a::

Set 1 input fasta files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

mum_input_b::

Set 1 input fasta files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.52.3 Parameters

mum_breaklen::

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

type: *integer*
default: *200*
optional: *True*

mum_matchmode::

use all matching fragments (max) or only unique matchers (mum)

type: *set*
default: *mum*
optional: *True*

mum_plot_raw::

plot an alternative visualization where mummer does not attempt to put the sequences in the correct order

type: *set*
default: *F*
optional: *True*

mum_self::

mummer against self

type: *set*
default: *T*
optional: *True*

9.52.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.53 ncbi

Download data from NCBI

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

9.53.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Download from NCBI

9.53.2 Parameters

db::

NCBI database

type: *string*
default: *nuccore*
optional: *True*

query::

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

type: *string*
default: ‘‘
optional: *True*

rename_sequence::

try to rename the sequence - note, this does not work if you are downloading more than one sequence

type: *boolean*
default: *False*
optional: *True*

sequence_name::

Name of the file to write the downloaded sequences to. Use 'from_dir' to have the sequence name extracted from the directory name

type: *string*
default: *out*
optional: *True*

9.53.3 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.54 newbler

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.

9.54.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.54.2 Filesets

input::

input SFF files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.54.3 Parameters

largecontig_cutoff::

min length of a contig in 454LargeContigs.fna

type: *integer*
default: ‘‘
optional: *True*

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.

type: *string*
default: *$(shell echo ‘basename $(CURDIR)* | sed “s/[////]//g”)‘
optional: *True*

mid_configuration::

Mid configuration file to use

type: *file*
default: ‘‘
optional: *True*

mids::

mids to use for this assembly

type: *string*
default: ‘‘
optional: *True*

min_identity::

Minimal overalp identity used during assembly

type: *integer*
default: ‘‘
optional: *True*

9.54.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.55 newjobtest

Execute a “simple” ad hoc analysis

:: Execute one command, No in or output files are tracked by Moa.

9.55.1 Commands

run *no help defined*

9.55.2 Parameters

process::

The command to execute

type: *string*
default: *True*
optional: *False*

9.55.3 Other

Backend ruff

Author Mark Fiers

Creation date Tue Mar 29 16:34:19 2011

Modification date Wed Mar 30 06:02:01 2011

9.56 nstretch

Nstretch

:: Run NSTRETCH on an set of input files

9.56.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.56.2 Parameters

input_dir::

input dir with the fasta files

type: *directory*
default: ‘‘
optional: *False*

input_extension::

extension of the input files

type: *string*
default: *fasta*
optional: *True*

len::

minimal number of Ns before its reported (default 10)

type: *integer*
default: *10*
optional: *True*

9.56.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.57 orthomcl

Bwa index builder

:: Run orhthomcl on a set of input fasta files

9.57.1 Commands

clean Remove all job data

run Run orthomcl

9.57.2 Filesets

input::

Directory with the input fasta files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.57.3 Parameters

db::

DB name

type: *string*
default: *orthomcl*
optional: *True*

host::

DB host

type: *string*
default: *localhost*
optional: *True*

login::

DB login

type: *string*
default: *{}*
optional: *False*

pass::

DB password

type: *string*
default: *{}*
optional: *False*

port::

DB port

type: *string*
default: *3306*
optional: *True*

prefix::

prefix for separating tables & output fields

type: *string*
default: *run1*
optional: *True*

9.57.4 Other

Backend ruff

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.58 pregap

Pregap

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

9.58.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.58.2 Parameters

cloning_vector::

File containing the cloning vector

type: *file*
default: ‘‘
optional: *False*

ecoli_screenseq::

File containing ecoli screen sequences

type: *file*
default: ‘‘
optional: *False*

input_dir::

Directory with the input data

type: *string*
default: ‘‘
optional: *False*

input_pattern::

file name pattern

type: *string*
default: ‘‘

optional: *False*

quality_value_clip::

quality cutoff

type: *integer*
default: *10*
optional: *True*

repeat_masker_lib::

File with a repeatmasker library

type: *file*
default: ‘‘
optional: *False*

sequencing_vector::

File containing the sequencing vector

type: *file*
default: ‘‘
optional: *False*

template::

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

type: *file*
default: *./files/pregap.config.*
optional: *True*

vector_primerfile::

File with the vector primers

type: *file*
default: ‘‘
optional: *False*

9.58.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.59 primersearch

Run EMBOSS primerpair

:: Search DNA sequences for matches with primer pairs

9.59.1 Commands

run *no help defined*

9.59.2 Filesets

input::

primersearch input sequence files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

primersearch output files

type: *map*
source: *input*
category: *output*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

primers::

Primer pairs file

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.59.3 Parameters

johns_postprocess::

Run John's Post processing

type: *boolean*
default: *True*
optional: *True*

mismatch::

Allowed percent mismatch

type: *integer*
default: *0*
optional: *True*

9.59.4 Other

Backend ruff

Author John McCallum

Creation date Mon Apr 04 08:51:23 2011

Modification date Mon Apr 04 09:00:42 2011

9.60 project

Create a project

:: Create a new project, a placeholder for project settings, and used by several plugins.

9.60.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

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

9.60.2 Parameters

description::

A description of what this project is supposed to achieve, how to use it, and what parameters are most important to set

type: *string*
default: ‘‘
optional: *True*

9.60.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.61 r

Run R

:: run the R script in moa.R, with the specified input files

9.61.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.61.2 Filesets

input::

Input files for r

type: *map*
source: *{}*
category: *input*

optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

9.61.3 Parameters

touch::

use touch files to track if input files have changed. If you set this to False, the touch files will still be generated, but will have no effect on whether or not an inputfile will be processed

type: *set*
default: *T*
optional: *True*

9.61.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.62 repmask

Repeatmasker

:: Run a default repeatmask on the input sequences

9.62.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.62.2 Filesets

input::

Input files for repmask

type: *map*
source: *{}*

category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

OUTPUT_FILESET_ID::

{}

type: *map*
source: *INPUT_FILESET_ID*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.62.3 Parameters

parallel::

No of threads to run in parallel

type: *integer*
default: *4*
optional: *True*

quick::

Quick job

type: *set*
default: *F*
optional: *True*

simple::

Mask *only* low complex/simple repeats, not interspersed repeats (Repeatmasker -(no)int parameter)

type: *set*
default: *F*
optional: *True*

species::

Repeatmasker species

type: *string*
default: ‘‘
optional: *True*

9.62.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.63 sam2bam

Convert SAM to BAM using samtools

:: Converts a FASTQ file to MAQ BFQ format.

9.63.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.63.2 Filesets

input::

input SAM files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.63.3 Parameters

9.63.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.64 samtools_pileup

:: Print the alignment in the pileup format.

9.64.1 Commands

clean Remove all job data, not the Moa job itself

run run samtools pileup command

9.64.2 Filesets

fasta::

reference fasta file

type: *map*
source: *{}*
category: *prerequisite*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

input::

bam or sam files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

output_bam::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.64.3 Parameters

cap_mapQ_at::

cap mapping quality at INT

type: *integer*
default: *60*
optional: *True*

extra_params::

any extra parameters

type: *string*
default: ‘‘
optional: *True*

filter_read_bits::

filtering reads with bits in INT

type: *integer*
default: *1796*
optional: *True*

input_is_SAM::

the input is in SAM

type: *boolean*
default: *False*
optional: *True*

num_haplotypes::

number of haplotypes in the sample (for -c/-g)

type: *integer*
default: *2*
optional: *True*

out_2nd_best::

output the 2nd best call and quality

type: *boolean*
default: *False*
optional: *True*

out_GLFv3_format::

output in the GLFv3 format (suppressing -c/-i/-s)

type: *boolean*
default: *False*
optional: *True*

out_maq_consensus::

output the maq consensus sequence

type: *boolean*
default: *False*
optional: *True*

phred_prob_indel::

phred prob. of an indel in sequencing/prep. (for -c/-g)

type: *integer*
default: *40*
optional: *True*

print_variants_only::

print variants only (for -c)

type: *boolean*
default: *False*
optional: *True*

prior_diff_haplotypes::

phred prob. of an indel in sequencing/prep. (for -c/-g)

type: *float*
default: *0.001*
optional: *True*

prior_indel_haplotypes::

number of haplotypes in the sample (for -c/-g)

type: *float*
default: *0.00015*
optional: *True*

show_lines_indels::

only show lines/consensus with indels

type: *boolean*
default: *False*
optional: *True*

simple_pileup_format::

simple (yet incomplete) pileup format

type: *boolean*
default: *False*
optional: *True*

theta_maq_model::

number of haplotypes in the sample (for -c/-g)

type: *float*
default: *0.85*

optional: *True*

use_SOAPsnp_model::

use the SOAPsnp model for SNP calling

type: *boolean*
default: *False*
optional: *True*

9.64.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Dec 15 17:06:48 2010

Modification date

9.65 scaf

Scaffolder

:: Scaffold a set of input files based on a blast against a reference sequence. This software is written around bambus

9.65.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.65.2 Parameters

input_file::

input file with the sequences to scaffold

type: *file*
default: ‘‘
optional: *False*

prefix::

prefix for scaffolding output files

type: *string*
default: *scaffolds*
optional: *True*

reference_file::

blast database of the reference set

type: *file*
default: ‘‘
optional: *True*

9.65.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.66 sffinfo

sffinfo

:: Roche sffinfor tool - extract information from sff files

9.66.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

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

9.66.2 Filesets

accession::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*

dir: *{}*

flowgram::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

input::

Sff input files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

quality::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

sequence::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.66.3 Parameters

accessions::

Output the accessions

type: *set*
default: *T*
optional: *True*

flowgrams::

output the flowgrams

type: *set*
default: *F*
optional: *True*

quality::

Output quality scores

type: *set*
default: *T*
optional: *True*

sequences::

Output the sequences

type: *set*
default: *T*
optional: *True*

untrimmed::

output untrimmed sequences & qualities

type: *set*
default: *F*
optional: *True*

9.66.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.67 simple

Execute a "simple" ad hoc analysis

:: Execute one command, No in or output files are tracked by Moa.

9.67.1 Commands

run *no help defined*

9.67.2 Parameters

process::

The command to execute

type: *string*
default: *True*
optional: *False*

9.67.3 Other

Backend ruff

Author Mark Fiers

Creation date Tue Mar 29 16:34:19 2011

Modification date Wed Mar 30 06:02:01 2011

9.68 soap_aln_pe

:: Use SOAP to align a set of paired fastq reads against a db

9.68.1 Commands

clean Remove all job data, not the Moa job itself

run run soap to align paired end reads

9.68.2 Filesets

bam_output::

{}

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

fa_input::

directory with reference fasta file name

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_forward_input::

fastq input files directory - forward files

type: *map*
source: *{}*
category: *input*
optional: *False*

extension: *{}*
glob: *{}*
dir: *{}*

fq_reverse_input::

fastq input files directory - reverse files

type: *map*
source: *fq_forward_input*
category: *input*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

soap_output::

{}

type: *map*
source: *fq_forward_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.68.3 Parameters

db_index_files::

Prefix name for reference index [*.index]

type: *string*
default: ‘‘
optional: *False*

edge_bp_no_gaps::

will not allow gap exist inside n-bp edge of a read

type: *integer*

default: *5*
optional: *True*

gap_size::

one continuous gap size allowed on a read

type: *integer*
default: *0*
optional: *True*

how_report_hits::

How to report repeat hits, 0=none; 1=random one; 2=all

type: *integer*
default: *1*
optional: *True*

long_read_seed_len::

For long reads with high error rate at 3'-end, those can't align whole length, then first align 5' INT bp subsequence as a seed, [256] use whole length of the read

type: *integer*
default: *256*
optional: *True*

match_mode::

Match mode for each read or the seed part of read, which shouldn't contain more than 2 mismaches, 0 exact match only 1 1 mismatch match only 2 2 mismatch match only 3 [gap] (coming soon) 4 find the best hits

type: *integer*
default: *4*
optional: *True*

max_insert_size::

maximal insert size allowed

type: *integer*
default: *600*
optional: *True*

min_insert_size::

minimal insert size allowed

type: *integer*
default: *400*
optional: *True*

mismatches_per_read::

Totally allowed mismatches in one read

type: *integer*
default: *6*
optional: *True*

out_file_unpaired_aln::

output file of unpaired alignment hits

type: *string*
default: *unpaired_aln.txt*
optional: *True*

out_read_id::

Output reads id instead of reads name

type: *boolean*
default: *False*
optional: *True*

out_unmapped_reads_file::

Output file name for unmapped reads

type: *string*
default: *unmapped_reads.txt*

optional: *True*

report_read_mismatches::

report all mismatched reads in SOAP Format

type: *boolean*
default: *False*
optional: *True*

rm_low_qual_reads::

Filter low quality reads contain more INT bp Ns

type: *integer*
default: *5*
optional: *True*

thread_num::

Multithreads, n threads

type: *integer*
default: *1*
optional: *True*

type_of_pe::

for long insert size of pair end reads RF (default means FR pair)

type: *boolean*
default: *False*
optional: *True*

9.68.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 30 07:56:48 2010

Modification date

9.69 soap_aln_se

:: Use SOAP to align a set of fastq reads against a db

9.69.1 Commands

clean Remove all job data, not the Moa job itself

run run soap to align single end reads

9.69.2 Filesets

bam_output::

{}

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

fa_input::

directory with reference fasta file name

type: *map*
source: *{}*
category: *prerequisite*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

fq_input::

Fastq input file

type: *map*
source: *{}*
category: *input*
optional: *False*

extension: *{}*
glob: *{}*
dir: *{}*

soap_output::

{}

type: *map*
source: *fq_input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.69.3 Parameters

db_index_files::

Prefix name for reference index [*.index]

type: *string*
default: ‘‘
optional: *False*

edge_bp_no_gaps::

will not allow gap exist inside n-bp edge of a read

type: *integer*
default: *5*
optional: *True*

gap_size::

one continuous gap size allowed on a read

type: *integer*
default: *0*
optional: *True*

how_report_hits::

How to report repeat hits, 0=none; 1=random one; 2=all

type: *integer*
default: *1*
optional: *True*

long_read_seed_len::

For long reads with high error rate at 3'-end, those can't align whole length, then first align 5' INT bp subsequence as a seed, [256] use whole length of the read

type: *integer*
default: *256*
optional: *True*

match_mode::

Match mode for each read or the seed part of read, which shouldn't contain more than 2 mismaches, 0 exact match only 1 1 mismatch match only 2 2 mismatch match only 3 [gap] (coming soon) 4 find the best hits

type: *integer*
default: *4*
optional: *True*

mismatches_per_read::

Totally allowed mismatches in one read

type: *integer*
default: *6*
optional: *True*

out_read_id::

Output reads id instead of reads name

type: *boolean*
default: *False*
optional: *True*

out_unmapped_reads_file::

Output file name for unmapped reads

type: *string*
default: *unmapped_reads.txt*
optional: *True*

rm_low_qual_reads::

Filter low quality reads contain more INT bp Ns

type: *integer*
default: *5*
optional: *True*

thread_num::

Multithreads, n threads

type: *integer*
default: *1*
optional: *True*

9.69.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 29 07:56:48 2010

Modification date

9.70 soapdb

Bowtie index builder

:: Builds a bowtie index from a reference sequence

9.70.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.70.2 Filesets

input_fasta::

input fasta file for the SOAP database

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.70.3 Parameters

9.70.4 Other

Backend ruff

Author Mark Fiers, Yogini Idnani

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.71 statsidx

:: Retrieve and print stats from BAM file to an index file

9.71.1 Commands

clean Remove all job data, not the Moa job itself

run run samtools idxstats

9.71.2 Filesets

input::

bam input files directory - forward files

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*

glob: *{}*
dir: *{}*

output::

{}

type: *map*
source: *input*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.71.3 Parameters

9.71.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Dec 08 17:06:48 2010

Modification date

9.72 unittest

:: Template used in testing - has no other purpose

9.72.1 Commands

clean Remove all job data

prepare prepare for the unittest

run Prepare & Run

run2 actually run

9.72.2 Filesets

input_1::

Input file set 1

type: *map*
source: *{}*
category: *input*
optional: *True*
extension: *{}*
glob: *{}*
dir: *{}*

input_2::

Input file set 2

type: *map*
source: *input_1*
category: *input*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

output::

output files

type: *map*
source: *input_1*
category: *output*
optional: *{}*
extension: *{}*
glob: *{}*
dir: *{}*

9.72.3 Parameters

test_string::

Test string values

type: *string*
default: *{}*
optional: *True*

9.72.4 Other

Backend ruff

Author Yogini Idnani, Mark Fiers

Creation date Wed Nov 25 17:06:48 2010

Modification date

9.73 upload2gbrowse

Library for uploading data to GBrowse

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

9.73.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

gupgo *no help defined*

initGbrowse *no help defined*

run *no help defined*

9.73.2 Parameters

gup_db::

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

type: *string*
default: ‘‘
optional: *False*

gup_fasta_extension::

extension of the FASTA files to upload (.fasta)

type: *string*
default: *fasta*
optional: *True*

gup_force_upload::

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

type: *set*
default: *F*
optional: *True*

gup_gff_extension::

extension of the GFF files to upload (.gff)

type: *string*
default: *gff*
optional: *True*

gup_upload_fasta::

upload fasta to gbrowse (T/F)

type: *set*
default: *F*
optional: *True*

gup_upload_gff::

upload gff to gbrowse (T/F)

type: *set*
default: *F*
optional: *True*

gup_user::

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

type: *string*
default: ‘‘
optional: *False*

marks_extensions::

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

type: *set*
default: *F*
optional: *True*

9.73.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.74 varscan

Varscan

:: Run VARSCAN to detect snps

9.74.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.74.2 Parameters

extra_params::

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

type: *string*
default: ‘‘
optional: *True*

input_file::

Varscan input alignments file

type: *file*
default: ‘‘
optional: *True*

output_name::

Base name of the output files

type: *string*
default: *out*
optional: *True*

perl_file::

the varscan (perl) executable

type: *file*
default: ‘‘
optional: *True*

9.74.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.75 vmatch

Vmatch

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

9.75.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.75.2 Parameters

db::

vmatch db to compare against

type: *file*
default: ‘‘
optional: *True*

extra_parameters::

extra parameters to feed to vmatch

type: *string*
default: ‘‘
optional: *True*

input_file::

input file with the sequences to map

type: *file*
default: ‘‘
optional: *True*

9.75.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.76 vmatchdb

vmatch database builder

:: Builds a vmatchdb index from a sequence

9.76.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run *no help defined*

9.76.2 Filesets

input::

Input files for vmatch

type: *map*
source: *{}*
category: *input*
optional: *False*
extension: *{}*
glob: *{}*
dir: *{}*

9.76.3 Parameters

name::

Name of the vmatch index to create

type: *string*
default: ‘‘
optional: *True*

pl::

Prefix length

type: *integer*
default: ‘‘
optional: *True*

9.76.4 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.77 vpcr

VPCR

:: Virtual PCR, based on Bowtie

9.77.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Predict the fragments that would be generated by a PCR

9.77.2 Parameters

bowtie_db::

Location of the bowtie database used for the vpcr

type: *file*

default: ‘‘
optional: *True*

insert_max::

maximum insert size for a vpcr fragment

type: *integer*
default: *10000*
optional: *True*

insert_min::

minimal insert size for a fragment

type: *integer*
default: *10*
optional: *True*

primer_1::

First primer to use

type: *string*
default: ‘‘
optional: *False*

primer_2::

Second primer to use

type: *string*
default: ‘‘
optional: *False*

9.77.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

9.78 vpcr_list

:: Virtual PCR, based on Bowtie

9.78.1 Commands

clean Remove all job data, not the Moa job itself, note that this must be implemented by the template.

run Predict the fragments that would be generated by a PCR

9.78.2 Parameters

bowtie_db::

Location of the bowtie database used for the vpcr

type: *file*
default: ‘‘
optional: *False*

insert_max::

maximum insert size for a vpcr fragment

type: *integer*
default: *10000*
optional: *True*

insert_min::

minimal insert size for a fragment

type: *integer*
default: *10*
optional: *True*

primer_list::

List of primers to check

type: *file*
default: *{}*
optional: *False*

9.78.3 Other

Backend gnumake

Author Mark Fiers

Creation date Wed Nov 10 07:56:48 2010

Modification date Wed Nov 10 07:56:48 2010

CHAPTER

TEN

MOA API

10.1 moa.actor

'Simple' wrapper around subprocess to execute code

moa.actor.**getLastStderr**(*job*)
Get the last stderr

moa.actor.**getLastStdout**(*job*)
Get the last stdout

moa.actor.**getRecentOutDir**(*job*)
Return the most recent output directory

moa.actor.**simpleRunner**(*wd*, *cl*, *silent=False*)

- put env in the environment
- Execute the commandline (in cl)
- store stdout & stderr in log files
- return the rc

10.2 moa.commands

Handle Moa commands (i.e. anything that you can run as *moa COMMAND* on the commandline

10.3 moa.job

class moa.job.**Job**(*wd*)
Class defining a single job

```
>>> wd = tempfile.mkdtemp()
>>> job = Job(wd)
>>> assert(isinstance(job, Job))
>>> assert(job.template.name == 'nojob')
```

Parameters

- **wd** – The directory containing the job

- **template** – The template a job should have. If undefined, read the template from *./.moa/template*
- **options** – Additional options to feed to this job

checkCommands(*command*)

Check command, and rearrange if there are delegates.

```
>>> job = newTestJob('unittest')
>>> assert(job.template.commands.run.delegate == ['prepare', 'run2'])
>>> assert(job.checkCommands('run2') == ['run2'])
>>> assert(job.checkCommands('run') == ['prepare', 'run2'])
>>> assert(job.checkCommands('prepare') == ['prepare'])
```

Parameters **commands** (*list of strings*) – The list of commands to check

Returns The checked list of commands

Return type list of strings

checkConfDir()

Check if the configuration directory exists. If not create it.

```
>>> job = newTestJob('unittest')
>>> confdir = os.path.join(job.wd, '.moa')
>>> assert(os.path.exists(confdir))
>>> import shutil
>>> shutil.rmtree(confdir)
>>> assert(os.path.exists(confdir) == False)
>>> job.checkConfDir()
>>> assert(os.path.exists(confdir))
```

defineOptions(*parser*)

Set command line options - deferred to the backends

```
>>> job = newTestJob('unittest')
>>> import optparse
>>> parser = optparse.OptionParser()
>>> job.defineOptions(parser)
```

execute(*command*, *verbose=False*, *silent=False*)

Execute *command* in the context of this job. Execution is alwasy deferred to the backend

#Note: Uncertain how to test verbose & silent

Parameters

- **command** (*string*) – the command to execute
- **verbose** (*Boolean*) – output lots of data
- **silent** (*Boolean*) – output nothing

getFiles()

Return all moa files - i.e. all files crucial to this job.

hasCommand(*command*)

Check if this job defines a certain command

Warning: THIS METHOD DOES NOT WORK PROPERLY YET

```
>>> job = newTestJob('unittest')
>>> assert(job.hasCommand('run'))
>>> assert(job.hasCommand('run2'))
>>> assert(not job.hasCommand('dummy'))
```

initialize()
Initialize a new job in the current wd

isMoa()
Check if this is a Moa directory - Currently, this needs to be overridden #weird; uncertain if this ever gets called

loadBackend()
load the backend

loadTemplate()
Load the template for this job, based on what configuration can be found

prepare()
Give this job a chance to prepare for execution - deferred to the backend.

```
>>> job = newTestJob('unittest')
>>> job.prepare()
```

refreshTemplate()
Reload the template into the local .moa/template.d directory

```
>>> job = newTestJob('unittest')
>>> templateFile = os.path.join(job.confDir, 'template.d', 'unittest.jinja2')
>>> assert(os.path.exists(templateFile))
>>> os.unlink(templateFile)
>>> assert(not os.path.exists(templateFile))
>>> job.refreshTemplate()
>>> assert(os.path.exists(templateFile))
```

setTemplate(*name*, *provider=None*)
Set a new template for this job

```
>>> job = newTestJob('unittest')
>>> job.setTemplate('adhoc')
>>> afile = os.path.join(job.confDir, 'template.d', 'adhoc.mk')
>>> assert(os.path.exists(afile))
```

moa.job.**newJob**(*wd*, *template*, *title*, *parameters=[]*, *provider=None*)
Create a new job in the wd and return the proper job object currently only makefile jobs are supported - later we'll scan the template, and instantiate the proper job type

```
>>> wd = tempfile.mkdtemp()
>>> job = newJob(wd, template='blast', title='test')
>>> assert(isinstance(job, Job))
>>> assert(job.template.name == 'blast')
>>> assert(job.conf.title == 'test')
```

Parameters

- **wd** – Directory to create this job in, note that this directory must already exists
- **template** (*String*) – Template name for this job
- **parameters** (*list of (key, value) tuples*) – A list of parameters to set for this job

Return type instance of `moa.job.Job`

`moa.job.`**`newTestJob`**(*template*, *title='Test job'*, *provider=None*)

for testing purposes - creates a temporary directory and uses that to instantiate a job. This function returns the job object created

```
>>> job = newTestJob(template = 'adhoc', title='test title')
>>> assert(isinstance(job, Job))
>>> assert(os.path.exists(job.wd))
>>> assert(job.conf.title == 'test title')
>>> assert(os.path.exists(os.path.join(job.wd, '.moa')))
>>> assert(os.path.exists(os.path.join(job.wd, '.moa', 'template')))
>>> assert(job.template.name == 'adhoc')
```

Returns the created job

Return type instance of `moa.job.Job`

10.4 moa.jobConf

moa job configuration

class `moa.jobConf.`**`JobConf`**(*job*)

to distinguish between attributes of this object & proper job configuration parameters

`doNotCheck`

these fields are not be type-checked

`doNotSave`

these fields are not to be saved

`keys`()

return a dict with all known parameters and values, either defined in the job configuration of the template

`load`(*confFile*, *delta=None*)

Load a configuration file

Parameters **delta** – if a value appears to be a relative path, try to correct for this. Currently this only works for files that exist. i.e.

`private`

these fields are private (i.e. not to be displayed by default)

10.5 moa.sysConf

Store Moa wide configuration

10.6 moa.ui

communicate information to the user

10.7 moa.utils

A set of random utilities used by Moa

moa.utils.**deprecated**(*func*)
Decorator function to flag a function as deprecated

Parameters **func** – any function

moa.utils.**flog**(*f*)
A simple logger - uses the `moa.logger` code to log the calling function. Use as a decorator:

```
@moa.utils.flog
def any_function(*args);
    ...
```

This is for debugging purposes (obviously)

Parameters **func** – Any python function

moa.utils.**getMoaBase**()
Return MOABASE - the directory where Moa is installed. This function also sets an environment variable *MOABASE*

```
>>> d = getMoaBase()
>>> assert(os.path.isdir(d))
>>> assert(os.path.isfile(os.path.join(d, 'README')))
>>> assert(os.path.isdir(os.path.join(d, 'lib')))
```

Return type string (path)

moa.utils.**getProcessInfo**(*pid*)
Return some info on a process

moa.utils.**getResource**(*what*)
Gets a data file from the moa package.

There are two possible locations where any resource could be, either three dirs up, or only one. This depends on if this a pypi (one dir up) package or the git package (three dirs up)

moa.utils.**listResource**(*what*)
List a directory

moa.utils.**moaDirOrExit**(*job*)
Check if the job contains a proper Moa job, if not, exit with an error message and a non-zero exit code.

Parameters **job** – An instance of `moa.job.Job`

moa.utils.**simple_decorator**(*decorator*)
This decorator can be used to turn simple functions into well-behaved decorators, so long as the decorators are fairly simple. If a decorator expects a function and returns a function (no

descriptors), and if it doesn't modify function attributes or docstring, then it is eligible to use this. Simply apply @simple_decorator to your decorator and it will automatically preserve the docstring and function attributes of functions to which it is applied.

Note; I got this code from somehwere, but forgot where exactly. This seems the most likely source:

http://svn.navi.cx/misc/trunk/djblets/djblets/util/decorators.py

10.8 moa.template

10.8.1 moa.template

Store information on a template. This module is also responsible for retrieving template information.

moa.template.**initTemplate**(**args*, ***kwargs*)

moa.template.**installTemplate**(*wd*, *tName*, *provider=None*)

Initialize the template - this means - try to figure out where the template came from & copy the template files into *job/.moa/template* & *job/.moa/template.d/extra*.

Currently all templates come from the moa repository. In the future, multiple sources must be possible

```
>>> import tempfile
>>> wd = tempfile.mkdtemp()
>>> installTemplate(wd, 'adhoc')
>>> templateFile = os.path.join(wd, '.moa', 'template')
>>> adhocFile = os.path.join(wd, '.moa', 'template.d', 'adhoc.mk')
>>> assert(os.path.exists(templateFile))
>>> assert(os.path.exists(adhocFile))
```

moa.template.**refresh**(*wd*)

Refresh the template - try to find out what the template is from {{wd}}/.moa/template.d/meta. If that doesn't work, revert to the default template. If default is not specified - exit with an error

```
>>> import tempfile
>>> wd = tempfile.mkdtemp()
>>> installTemplate(wd, 'adhoc')
>>> templateFile = os.path.join(wd, '.moa', 'template')
>>> adhocFile = os.path.join(wd, '.moa', 'template.d', 'adhoc.mk')
>>> os.unlink(adhocFile)
>>> os.unlink(templateFile)
>>> assert(not os.path.exists(templateFile))
>>> assert(not os.path.exists(adhocFile))
>>> refresh(wd)
>>> assert(os.path.exists(templateFile))
>>> assert(os.path.exists(adhocFile))
```

10.8.2 moa.template.template

Store information on a template. This module is also responsible for retrieving template information.

class moa.template.template.**Template**(*templateFile*)

Template extends Yaco

getRaw()
Return a Yaco representation of the yaml-template, without any of this Template processing. This is really useful when processing a template that needs to be written back to disk

```
>>> import moa.job
>>> job = moa.job.newTestJob(template='adhoc')
>>> raw = job.template.getRaw()
>>> assert(isinstance(raw, Yaco.Yaco))
>>> assert(raw.has_key('parameters'))
```

10.9 moa.template.provider

10.9.1 moa.provider.core

Provides templates from the Moa package.

10.10 moa.backend

10.10.1 Gnumake

10.10.2 Ruff

Ruffus/Jinja Backend

members

10.11 moa.plugin

10.11.1 adhoc - create jobs from adhoc bash code

moa.plugin.adhoc.**createAdhoc**(*job*)
Creates an adhoc job.

moa.plugin.adhoc.**createMap**(*job*)
Create a 'map' adhoc job.

There are a number of ways this command can be used:

```
$ moa map -t 'a title' -- echo 'define a command'
```

Anything after – will be the executable command. If omitted, Moa will query the user for a command.

Moa will also query the user for input & output files. An example session:

```
$ moa map -t 'something intelligent'
process:
> echo 'processing {{ input }} {{ output }}'
input:
> ../10.input/*.txt
```

```
output:
> ./*.out
```

Assuming you have a number of text files in the *../10/input/* directory, you will see, upon running:

```
processing ../10.input/test.01.txt ./test.01.out
processing ../10.input/test.02.txt ./test.02.out
processing ../10.input/test.03.txt ./test.03.out
...
```

moa.plugin.adhoc.**createSimple**(*job*)
Create a 'simple' adhoc job. Simple meaning that no in or output files are tracked.

There are a number of ways this command can be used:

```
moa simple -t 'a title' -- echo 'define a command'
```

Anything after – will be the executable command. Note that bash will attempt to process the command line. A safer method is:

```
moa simple -t 'a title'
```

Moa will query you for a command to execute (the parameter *process*).

moa.plugin.adhoc.**exclamate**(*job*)
Set the 'process' parameter to the last issued command. If no moa job exists, create a 'simple'job.

moa.plugin.adhoc.**exclamateInJob**(*job*)
Reuse the last issued command: set it as the 'process' parameters in the current job

moa.plugin.adhoc.**exclamateNoJob**(*job*)
Create a "simple" job & set the last command to the 'process' parameter

10.11.2 configure - Configure jobs

Control job configuration

moa.plugin.configure.**configSet**(*job*)
This command can be used in a number of ways:

```
moa set PARAMETER_NAME=PARAMETER_VALUE
moa set PARAMETER_NAME='PARAMETER VALUE WITH SPACES'
moa set PARAMETER_NAME
```

In the first two forms, moa sets the parameter *PARAMETER_NAME* to the *PARAMETER_VALUE*. In the latter form, Moa will present the user with a prompt to enter a value. Note that the first two forms the full command lines will be processed by bash, which can either create complications or prove very useful. Take care to escape variables that you do not want to be expandend and use single quotes where you can.

moa.plugin.configure.**configShow**(*job*)
Show all parameters know to this job. Parameters in **bold** are specifically configured for this job (as opposed to those parameters that are set to their default value). Parameters in red are not configured, but need to be for the template to operate. Parameters in blue are not configured either, but are optional.

moa.plugin.configure.**configUnset**(*job*)
Remove a configured parameter from this job. In the parameter was defined by the job template, it reverts back to the default value. If it was an ad-hoc parameter, it is lost from the configuration.

moa.plugin.configure.**defineCommands**(*data*)
Set the moa commands for this plugin

10.11.3 extraCommands - Pre & Post commands

Allow execution of a bash oneline before & after job completion

moa.plugin.extraCommands.**postRun**(*data*)
If defined, execute the postCommand

moa.plugin.extraCommands.**preRun**(*data*)
If defined, execute the precommand

10.11.4 fileset - define sets of in&output files

moa.plugin.fileset.**defineCommands**(*data*)
Set the moa commands for this plugin

moa.plugin.fileset.**preFiles**(*data*)
Run before execution of any command (backend or plugin)

moa.plugin.fileset.**pre_command**(*data*)
Run before execution of any command (backend or plugin)

moa.plugin.fileset.**preparefilesets**(*data*)
prepare all filesets

moa.plugin.fileset.**showFiles**(*job*)
moa files - Display discovered & inferred files for this job

Usage:

```
moa files
```

Display a list of all files discovered (for input & prerequisite type filesets) and inferred from these for map type filesets.

10.11.5 help - generate help

moa.plugin.help.**pager**(*template*, *templateData*)
render the template & send it to the pager

moa.plugin.help.**templateHelp**(*job*)

moa.plugin.help.**welcome**(*job*)
print a welcome message

10.11.6 info - Job information

Print info on Moa jobs and Moa

moa.plugin.info.**defineCommands**(*data*)
Set the moa commands for this plugin

moa.plugin.info.**rawCommands**(*job*)
(private) **moa raw_commands** - Print a list of all known commands

Usage:

```
moa raw_commands
```

Print a list of known Moa commands, both global, plugin defined commands as template specified ones. This command is mainly used by software interacting with Moa.

moa.plugin.info.**rawParameters**(*job*)
(private) **moa raw_parameters** - Print out a list of all known parameters

Usage:

```
moa raw_parameters
```

print a list of all defined or known parameters

moa.plugin.info.**status**(*job*)
moa status - print out a short status status message

Usage:

```
moa status
```

moa.plugin.info.**version**(*job*)
moa version - Print the moa version number

10.11.7 lock - Lock/Unlock moa jobs

10.11.8 logger - Log Moa activity

moa.plugin.logger.**niceRunTime**(*d*)
Nice representation of the run time d is time duration string

moa.plugin.logger.**showLog**(*job*)
moa log - show a log of the most recent moa calls

Usage:

```
moa log [LINES]
```

Shows a log of moa commands executed. Only commands with an impact on the pipeline are logged, such as *moa run* & *moa set*. The number of log entries to display can be controlled with the optional LINES parameter.

10.11.9 logo - Print a big, in your face, moa logo

moa.plugin.logo.**preRun**(*data*)
Print the logo just before a moa run

10.11.10 moaGit - maintain a git repository with job information

moa.plugin.moaGit.**gitlog**(*job*)
Print a log to screen

moa.plugin.moaGit.**postNew**(*data*)
To be executed just after the 'moa new' command

moa.plugin.moaGit.**postSet**(*data*)
Execute just after setting a parameter

10.11.11 moautil - Some extra utilities - copy/move jobs

moa.plugin.moautil.**archive**(*job*)
Archive a job, or tree with jobs for later execution.

This command stores only those files that are necessary for execution of this job, that is: templates & configuration. In & output files, and any other file are ignored. An exception to this are all files that start with 'moa.'

Usage:

```
moa archive
```

or

moa archive -r

The latter archives all jobs in subdirs of the current directory.

Note that only those directories that contain a moa job are included into the archive.

moa.plugin.moautil.**moacp**(*job*)
Copy a moa job, or a tree with jobs.

moa cp copies only those files defining a job: the template files and the job configuration. Additionaly, all files in the moa directory that start with *moa.* (for example *moa.description* are copied as well. Data and log files are not copied!

The command has two modes of operation. The first is:

```
moa cp 10.from 20.to
```

copies the moa job in 10.from to a newly created 20.to directory. If the *20.to* directory already exists, a new directory is created in *20.to/10.from*. As an shortcut one can use:

```
moa cp 10.from 20
```

in which case the job will be copied to the *20.from* directory.

If the source (*10.from*) directory is not a Moa job, the command exits with an error.

The second mode of operation is recursive copying:

```
moa cp -r 10.from 20.to
```

in which case all subdirectories under 10.from are traversed and copied - if a directory contains a Moa job.

::TODO.. Warn for changing file & dir links

moa.plugin.moautil.**moaren**(*job*)
Renumber or rename a moa job..

10.11.12 newjob - Instantiate new jobs

moa.plugin.newjob.**newJob**(*job*)
moa new

Usage:

```
moa new TEMPLATE_NAME -t 'a descriptive title'
```

10.11.13 Pack - saves pipelines, or parts of pipelines for reuse

moa.plugin.pack.**pack**(*data*)
Create an adhoc job

moa.plugin.pack.**packArgs**(*data*)
Get the arguments that a packfile was created with

moa.plugin.pack.**prepare**(*data*)
Check if the packpath exists

10.11.14 parameterCheck - check parameters

moa.plugin.parameterCheck.**defineCommands**(*data*)
Define the parameters test commands

moa.plugin.parameterCheck.**promptSnippet**(*data*)
Function used by the prompt plugin to generate snippets for inlusion in the prompt

10.11.15 prompt - Moa BASH prompt enhancer

moa.plugin.prompt.**defineCommands**(*data*)
Set the moa commands for this plugin

10.11.16 status - Job Status

Possible job states:

- waiting - not yet executed
- running - is currently being executed
- success - finished succesfully
- error - finished with an error
- interrupted - manual interruption

moa.plugin.status.**defineCommands**(*data*)
Set the moa commands for this plugin

moa.plugin.status.**kill**(*job*)
See if a job is running, if so - kill it

moa.plugin.status.**pause**(*job*)
pause a running job

moa.plugin.status.**resume**(*job*)
pause a running job

moa.plugin.status.**status**(*job*)
moa status - print out a short status status message

Usage:

```
moa status
```

10.11.17 template - information on templates

moa.plugin.template.**defineCommands**(*data*)
Set the moa commands for this plugin

moa.plugin.template.**dumpTemplate**(*job*)
moa template_dump - Show raw template information

Usage:

```
moa template_dump [TEMPLATE_NAME]
```

Show the raw template data.

moa.plugin.template.**listTemplates**(*job*)
moa list - Print a list of all known templates

Usage:

```
moa list
moa list -l
```

Print a list of all templates known to this moa installation. If the option '-l' is used, a short description for each tempalte is printed as well.

moa.plugin.template.**refresh**(*job*)
Refresh the template - i.e. reload the template from the central repository.

moa.plugin.template.**template**(*job*)
moa template - Print the template name of the current job

Usage:

```
moa template
```

moa.plugin.template.**templateSet**(*job*)
moa template_set - set a template parameter.

This only works for top level template parameters

10.11.18 test - Run unittests

10.11.19 twit - Tweet results

Use twitter to send a message upon job completion

moa.plugin.twit.**postRun**(*job*)
Send a tweet out upon completing the default run

10.12 Yaco

Yaco provides a *dict* like structure that can be serialized to & from yaml. Yaco objects behave as dictionaries but also allow attribute access (loosely based on this **'recipe < http://code.activestate.com/recipes/473786/>'_**). Sublevel dictionaries are automatically converted to Yaco objects, allowing sublevel attribute access, for example:

```
>>> x = Yaco()
>>> x.test = 1
>>> x.sub.test = 2
>>> x.sub.test
2
```

Note that sub-dictionaries do not need to be initialized. This has as a consequence that requesting uninitialized items automatically return an empty Yaco object (inherited from a dictionary).

Yaco can be found in the Python package index and is also part of the Moa source distribution

10.12.1 Autogenerating keys

An important feature (or annoyance) of Yaco is the auto generation of keys that are not present (yet). For example:

```
>>> x = Yaco()
>>> x.a.b.c.d = 1
>>> assert(x.a.b.c.d == 1)
```

works - *a*, *b* and *c* are assumed to be *Yaco* dictionaries and d is give value *1*. This makes populating data structures easy.

It might also generate some confusion when querying for keys in the Yaco structure - if a key does not exists, it automatically comes back as an empy *dict* or *Yaco* object (renders as *{}*). This means that if it is easy to check if a certain 'branch' of a Yaco datastructure exists:

```
>>> x = Yaco()
>>> assert (not x.a.b)
```

but now the following works as well:

```
>>> assert(x.has_key('a'))
>>> assert(x.a.has_key('b'))
```

So, a safe way to test a data structure, without introducing extra branches is:

```
>>> x = Yaco()
>>> assert(not x.has_key('a'))
```

Todo: Need to find a more elegant way of testing without introducing data structures

class Yaco.**Yaco**(*data={}*)

Rather loosely based on http://code.activestate.com/recipes/473786/ (r1)

```
>>> v= Yaco()
>>> v.a = 1
>>> assert(v.a == 1)
>>> assert(v['a'] == 1)
>>> v= Yaco({'a':1})
>>> assert(v.a == 1)
>>> assert(v['a'] == 1)
```

get_data()

Prepare & parse data for export

```
>>> y = Yaco()
>>> y.a = 1
>>> y.b = 2
>>> y._c = 3
>>> assert(y._c == 3)
>>> d = y.get_data()
>>> assert(d.has_key('a') == True)
>>> assert(d.has_key('b') == True)
>>> assert(d.has_key('_c') == False)
>>> y._private = ['b']
>>> d = y.get_data()
>>> assert(d.has_key('a') == True)
>>> assert(d.has_key('b') == False)
>>> assert(d.has_key('_c') == False)
```

load(*from_file*)

Load this dict from_file

```
>>> import yaml
>>> import tempfile
>>> tf = tempfile.NamedTemporaryFile(delete=False)
>>> tf.write(yaml.dump({'a' : [1,2,3, [1,2,3, {'d' : 4}]], 'b': 4, 'c': '5'}))
>>> tf.close()
>>> y = Yaco()
>>> y.load(tf.name)
>>> assert(y.a[3][3].d == 4)
```

pretty()

Return data as a pprint.pformatted string

save(*to_file*, *doNotSave=[]*)

update(*data*)

```
>>> v = Yaco({'a' : [1,2,3,{'b' : 12}]})
>>> assert(v.a[3].b == 12)

>>> v = Yaco({'a' : [1,2,3,[1,{'b' : 12}]]})
>>> assert(v.a[3][1].b == 12)
```

10.13 fist

Filesets

Handle & manipulate sets of files

This module aims at providing classes to handle and manipulate sets of files. Two simple examples are a simple set containing one file (`fist.fistSingle`) or a *glob* based set of files (`fist.fistFileset`). A more complicated example is `fistMapset` that maps another fileset based on a pattern.

Each fileset inherits from *list* - hence fist filesets behave as lists.

Future work should allow the definition of remote filesets (for example over http or ssh).

Each fist class is istantiated with a url defining the file(set). In the case of `fist.fistFileset` this url contains a globbing characters:

```
fs = fist.fistFileset('/tmp/*,txt')
```

This fileset object contains a list with all **.txt* files in */tmp*. Subsequently it is possible to map this set

class `fist.`**`fistCore`**(*url*)
Core class for all fist classes

class `fist.`**`fistFileset`**(*url*)
Most basic set of files - handle a set of files described by a single URI with wildcards, for example:

```
* `*.txt`
* `../*.txt`
* `file:///home/name/data/*.txt`
```

```
>>> f = fistFileset('*.txt')
>>> assert(f.path=='.')
>>> assert(f.glob=='*.txt')
>>> assert(f.path=='.')
>>> assert(f.glob=='*.txt')
>>> f = fistFileset('/tmp')
>>> assert(f.path=='/tmp')
>>> assert(f.glob=='*')
>>> f = fistFileset('/tmp/*.txt')
>>> assert(f.path=='/tmp')
>>> assert(f.glob=='*.txt')
>>> f = fistFileset('../*.txt')
>>> assert(f.path=='..')
>>> assert(f.glob=='*.txt')
>>> f = fistFileset(os.path.join(wd, 'in', '*.txt'))
>>> f.resolve()
>>> assert(len(f) == 100)
>>> f = fistFileset(os.path.join(wd, 'in', 'in1*.txt'))
>>> f.resolve()
>>> assert(len(f) == 10)
>>> f = fistFileset('~/*')
>>> f.resolve()
>>> assert(len(f) > 0)
```

class `fist.`**`fistMapset`**(*url*)
fistMapset

Map set - map a fileset based on a target uri

```
>>> f = fistFileset(os.path.join(wd, 'in', '*'))
>>> f.resolve()
>>> assert(len(f) == 100)
>>> ##
>>> ## Null mapping
>>> ##
>>> m = fistMapset('*/*')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert(os.path.join(wd, 'in/in18.txt') in m)
>>> ##
>>> ## simple folder mapping
>>> ##
>>> m = fistMapset('out/*')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('out/in18.txt' in m)
>>> ##
>>> ## simple folder mapping
>>> ##
>>> m = fistMapset('./*')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('./in18.txt' in m)
>>> ##
>>> ## simple folder & mapping & extension append
>>> ##
>>> m = fistMapset('out/*.out')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('out/in18.txt.out' in m)
>>> ##
>>> ## New from fileset - now with a pattern defining the extension
>>> ##
>>> f = fistFileset(os.path.join(wd, 'in', '*.txt'))
>>> f.resolve()
>>> ##
>>> ## extension mapping
>>> ##
>>> m = fistMapset('out/*.out')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('out/in18.out' in m)
>>> ##
>>> ## New from fileset - now with a pattern defining file glob &
>>> ## extension
>>> ##
>>> f = fistFileset(os.path.join(wd, 'in', 'in*.txt'))
>>> f.resolve()
>>> ##
>>> ## more complex filename mapping
>>> ##
>>> m = fistMapset('out/test*.out')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('out/test18.out' in m)
```

```
>>> ##
>>> ## mapping keeping the extension the same
>>> ##
>>> m = fistMapset('out/test*.txt')
>>> m.resolve(f)
>>> assert(len(m) == 100)
>>> assert('out/test18.txt' in m)
```

resolve(*mapFrom*)
Resolve the mapped set based on a input fileSet

resolver(*mapFrom*, *list*)
map all files in the incoming list

class fist.**fistSingle**(*url*)
Represents a single file

init()
Assuming the url is a single file

CHAPTER

ELEVEN

MORE INFORMATION

- Browse the Moa source at Github.
- Download a pdf version of the manual.

CHAPTER

TWELVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

BIBLIOGRAPHY

[BLAST] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10. PubMed PMID: 2231712.

PYTHON MODULE INDEX

f

fist, 201

m

moa.actor, 187
moa.backend.gnumake, 193
moa.backend.ruff, 193
moa.commands, 187
moa.job, 187
moa.jobConf, 190
moa.plugin.adhoc, 193
moa.plugin.configure, 194
moa.plugin.extraCommands, 195
moa.plugin.fileset, 195
moa.plugin.help, 195
moa.plugin.info, 195
moa.plugin.lock, 196
moa.plugin.logger, 196
moa.plugin.logo, 196
moa.plugin.moaGit, 196
moa.plugin.moautil, 197
moa.plugin.newjob, 198
moa.plugin.pack, 198
moa.plugin.parameterCheck, 198
moa.plugin.prompt, 198
moa.plugin.status, 198
moa.plugin.template, 199
moa.plugin.test, 199
moa.plugin.twit, 200
moa.sysConf, 190
moa.template, 192
moa.template.provider.core, 193
moa.template.template, 192
moa.ui, 190
moa.utils, 191

y

Yaco, 200

INDEX

A

archive() (in module moa.plugin.moautil), 197

C

checkCommands() (moa.job.Job method), 188
checkConfDir() (moa.job.Job method), 188
configSet() (in module moa.plugin.configure), 194
configShow() (in module moa.plugin.configure), 194
configUnset() (in module moa.plugin.configure), 194
createAdhoc() (in module moa.plugin.adhoc), 193
createMap() (in module moa.plugin.adhoc), 193
createSimple() (in module moa.plugin.adhoc), 194

D

defineCommands() (in module moa.plugin.configure), 195
defineCommands() (in module moa.plugin.fileset), 195
defineCommands() (in module moa.plugin.info), 195
defineCommands() (in module moa.plugin.parameterCheck), 198
defineCommands() (in module moa.plugin.prompt), 198
defineCommands() (in module moa.plugin.status), 198
defineCommands() (in module moa.plugin.template), 199
defineOptions() (moa.job.Job method), 188
deprecated() (in module moa.utils), 191
doNotCheck (moa.jobConf.JobConf attribute), 190
doNotSave (moa.jobConf.JobConf attribute), 190
dumpTemplate() (in module moa.plugin.template), 199

E

exclamate() (in module moa.plugin.adhoc), 194
exclamateInJob() (in module moa.plugin.adhoc), 194
exclamateNoJob() (in module moa.plugin.adhoc), 194
execute() (moa.job.Job method), 188

F

fist (module), 201
fistCore (class in fist), 202
fistFileset (class in fist), 202
fistMapset (class in fist), 202
fistSingle (class in fist), 204
flog() (in module moa.utils), 191

G

get_data() (Yaco.Yaco method), 201
getFiles() (moa.job.Job method), 188
getLastStderr() (in module moa.actor), 187
getLastStdout() (in module moa.actor), 187
getMoaBase() (in module moa.utils), 191
getProcessInfo() (in module moa.utils), 191
getRaw() (moa.template.template.Template method), 192
getRecentOutDir() (in module moa.actor), 187
getResource() (in module moa.utils), 191
gitlog() (in module moa.plugin.moaGit), 197

H

hasCommand() (moa.job.Job method), 188

I

init() (fist.fistSingle method), 204
initialize() (moa.job.Job method), 189
initTemplate() (in module moa.template), 192
installTemplate() (in module moa.template), 192
isMoa() (moa.job.Job method), 189

J

Job (class in moa.job), 187
JobConf (class in moa.jobConf), 190

K

keys() (moa.jobConf.JobConf method), 190
kill() (in module moa.plugin.status), 198

L

listResource() (in module moa.utils), 191
listTemplates() (in module moa.plugin.template), 199
load() (moa.jobConf.JobConf method), 190
load() (Yaco.Yaco method), 201
loadBackend() (moa.job.Job method), 189
loadTemplate() (moa.job.Job method), 189

M

moa.actor (module), 187
moa.backend.gnumake (module), 193
moa.backend.ruff (module), 193
moa.commands (module), 187
moa.job (module), 187
moa.jobConf (module), 190
moa.plugin.adhoc (module), 193
moa.plugin.configure (module), 194
moa.plugin.extraCommands (module), 195
moa.plugin.fileset (module), 195
moa.plugin.help (module), 195
moa.plugin.info (module), 195
moa.plugin.lock (module), 196
moa.plugin.logger (module), 196
moa.plugin.logo (module), 196
moa.plugin.moaGit (module), 196
moa.plugin.moautil (module), 197
moa.plugin.newjob (module), 198
moa.plugin.pack (module), 198
moa.plugin.parameterCheck (module), 198
moa.plugin.prompt (module), 198
moa.plugin.status (module), 198
moa.plugin.template (module), 199
moa.plugin.test (module), 199
moa.plugin.twit (module), 200
moa.sysConf (module), 190
moa.template (module), 192
moa.template.provider.core (module), 193
moa.template.template (module), 192
moa.ui (module), 190
moa.utils (module), 191
moacp() (in module moa.plugin.moautil), 197
moaDirOrExit() (in module moa.utils), 191
moaren() (in module moa.plugin.moautil), 197

N

newJob() (in module moa.job), 189
newJob() (in module moa.plugin.newjob), 198
newTestJob() (in module moa.job), 190
niceRunTime() (in module moa.plugin.logger), 196

P

pack() (in module moa.plugin.pack), 198
packArgs() (in module moa.plugin.pack), 198
pager() (in module moa.plugin.help), 195
pause() (in module moa.plugin.status), 199
postNew() (in module moa.plugin.moaGit), 197
postRun() (in module moa.plugin.extraCommands), 195
postRun() (in module moa.plugin.twit), 200
postSet() (in module moa.plugin.moaGit), 197
pre_command() (in module moa.plugin.fileset), 195
preFiles() (in module moa.plugin.fileset), 195
prepare() (in module moa.plugin.pack), 198
prepare() (moa.job.Job method), 189
preparefilesets() (in module moa.plugin.fileset), 195
preRun() (in module moa.plugin.extraCommands), 195
preRun() (in module moa.plugin.logo), 196
pretty() (Yaco.Yaco method), 201
private (moa.jobConf.JobConf attribute), 190
promptSnippet() (in module moa.plugin.parameterCheck), 198

R

rawCommands() (in module moa.plugin.info), 196
rawParameters() (in module moa.plugin.info), 196
refresh() (in module moa.plugin.template), 199
refresh() (in module moa.template), 192
refreshTemplate() (moa.job.Job method), 189
resolve() (fist.fistMapset method), 204
resolver() (fist.fistMapset method), 204
resume() (in module moa.plugin.status), 199

S

save() (Yaco.Yaco method), 201
setTemplate() (moa.job.Job method), 189
showFiles() (in module moa.plugin.fileset), 195
showLog() (in module moa.plugin.logger), 196
simple_decorator() (in module moa.utils), 191
simpleRunner() (in module moa.actor), 187
status() (in module moa.plugin.info), 196
status() (in module moa.plugin.status), 199

T

Template (class in moa.template.template), 192

template() (in module moa.plugin.template), 199
templateHelp() (in module moa.plugin.help), 195
templateSet() (in module moa.plugin.template), 199

U

update() (Yaco.Yaco method), 201

V

version() (in module moa.plugin.info), 196

W

welcome() (in module moa.plugin.help), 195

Y

Yaco (class in Yaco), 201
Yaco (module), 200