

DIGEST

API Documentation

October 1, 2014

Contents

Contents	1
1 Module DIGEST	2
1.1 Functions	2
1.2 Variables	2
2 Module DIGEST functions	3
2.1 Functions	3
2.2 Variables	6
2.3 Class MyDialect	6
2.3.1 Methods	6
2.3.2 Class Variables	6
2.4 Class jobLauncher	7
2.4.1 Methods	7
2.4.2 Properties	8
2.4.3 Class Variables	8
2.5 Class sequence	9
2.5.1 Methods	9
2.5.2 Properties	9
2.5.3 Class Variables	9
2.6 Class ORF	10
2.6.1 Methods	10
2.6.2 Properties	10
2.6.3 Class Variables	10
2.7 Class ContigORF	11
2.7.1 Methods	11
2.7.2 Properties	11
2.7.3 Class Variables	12
2.8 Class Cigar	12
2.8.1 Methods	12
2.8.2 Properties	12
2.8.3 Class Variables	13
2.9 Class alignmentSAM	13
2.9.1 Methods	13
2.9.2 Properties	13
2.9.3 Class Variables	14
2.10 Class cluster	14

2.10.1 Methods	15
2.10.2 Properties	15
2.10.3 Class Variables	15
2.11 Class clusterSequence	16
2.11.1 Methods	16
2.11.2 Properties	16
2.11.3 Class Variables	16
3 Module ExtractFastaFromSAM	18
3.1 Functions	18
3.2 Variables	18
4 Module SAMparser	19
4.1 Functions	19
4.2 Variables	19
5 Module extendTargets_allPossibility	20
5.1 Functions	20
5.2 Variables	20
6 Module extractORFsequences	21
6.1 Functions	21
6.2 Variables	21
7 Module removeIdenticalSeq	22
7.1 Functions	22
7.2 Variables	22
8 Script script-cd_hit_para_CCRT_py	23
8.1 Functions	23
8.2 Variables	24

1 Module DIGEST

Main script for DIGEST workflow CCRT version

Requires:

- python¹ (tested with 2.7.3)
- hashlib²
- BWA³
- RayMeta⁴
- Metagene⁵
- cd-hit⁶ (tested with v4.5.8-2012-03-24)
- DIGEST_functions.py (PYTHONPATH)
- SAMparser.py
- ExtractFastaFromSAM.py
- extendTargets_allPossibility.py
- hashlib⁷fasta-splitter.pl
- extractORFsequences.py
- removeIdenticalSeq.py
- cd-hit-para-CCRT.py
- DIGEST_clear.sh
- DIGEST_check.sh

1.1 Functions

```
get_parser()
```

```
main()
```

1.2 Variables

Name	Description
__doc__	Value: ...
__package__	Value: None
e	Value: 2.71828182846
pi	Value: 3.14159265359

¹<https://www.python.org/downloads/>

²<https://pypi.python.org/pypi/hashlib>

³<http://sourceforge.net/projects/bio-bwa/files/>

⁴<http://sourceforge.net/projects/denovoassembler/files/>

⁵http://metagene.nig.ac.jp/metagene/download_mga.html

⁶<https://code.google.com/p/cdhit/downloads/list>

⁷<http://kirill-kryukov.com/study/tools/fasta-splitter/>

2 Module DIGEST_functions

Class and functions use by the DIGEST workflow

Requires:

- jobArrayLSFlauncher_modif.sh
- mpirun-genoscope-modif.sh

2.1 Functions

exist(*fname*)

Check the existence of a file.

Parameters

fname: file name
(type=string)

Return Value

1 if the file is present, 0 otherwise
(type=integer)

clstrParser(*file*)

parse .clstr file from cd hit

Parameters

file: file name
(type=string)

Return Value

list of cluster object
(type=list)

fastaReader(*file*)

Fasta parser

Parameters

file: file name
(type=string)

Return Value

dictionary of sequence object with sequence ID as key
(type=dictionary)

geneExtended(*orfSTART*, *orfEND*, *alignSTART*, *alignEND*, *orfSTATUT*)

Check if a gene has been extended

Parameters

- orfSTART*:** ORF start position in sequence extended
(type=integer)
- orfEND*:** ORF end position in sequence extended
(type=integer)
- alignSTART*:** alignment start position of sequence on contig
(type=integer)
- alignEND*:** alignment end position of sequence on contig
(type=integer)
- orfSTATUT*:** ORF complete or partial
(type=string)

Return Value

- True if the gene is completed, False otherwise
(type=boolean)

geneSeen(*orfSTART*, *orfEND*, *alignSTART*, *alignEND*)

Check if a gene has been seen

Parameters

- orfSTART*:** ORF start position in sequence extended
(type=integer)
- orfEND*:** ORF end position in sequence extended
(type=integer)
- alignSTART*:** alignment start position of sequence on contig
(type=integer)
- alignEND*:** alignment end position of sequence on contig
(type=integer)

Return Value

- True if the gene is seen, False otherwise
(type=boolean)

reverseComplement(*sequen*)

make the reverse complement of a sequence

Parameters

- sequen*:** nucleotide sequence
(type=string)

Return Value

- reverse complement of sequence
(type=string)

metageneParser(*file*)

Sotck ORFs of metagene file

Parameters

file: file name

(*type=string*)

Return Value

a a dictionnary with contigs IDs as key and contig object as value

(*type=dictionary*)

subjectStartStop(*alignment*, *subjectLength*)

From an alignment and a length, compute the start and stop alignment position

Parameters

alignment: alignmentSAM object
(*type=alignmentSAM*)

subjectLength: subject sequence length
(*type=integer*)

Return Value

a list with the position start and stop of the alignment (if start = -1 -> alignment start befor the subject sequence ; if stop = -2 -> alignment stop after the subject sequence)

(*type=list*)

fileLineNumber(*file*)

Compute the int number of lines from a file

Parameters

file: file name
(*type=string*)

Return Value

number of lines

(*type=integer*)

nbSequenceFasta(*file*)

Compute the number of sequences in a FASTA file

Parameters

file: file name
(*type=string*)

Return Value

number of sequences

(*type=integer*)

writeORF(ORFlist, prefix, ID, sequence, n)

write ORFs in PREFIX_complete.fasta file or PREFIX_partial.fasta file

Parameters

- ORFlist:** list of ORF object
(*type=list*)
- prefix:** prefix of output file name
(*type=string*)
- ID:** sequence ID
(*type=string*)
- sequence:** nucleotide sequence
(*type=string*)
- n:** limite length for partial ORF
(*type=integer*)

2.2 Variables

Name	Description
__doc__	Value: ...
__package__	Value: None

2.3 Class MyDialect



csv class use to read csv files

2.3.1 Methods

Inherited from csv.Dialect

`__init__()`

2.3.2 Class Variables

Name	Description
delimiter	Value: '\t'
quotechar	Value: None
escapechar	Value: '\\'

continued on next page

Name	Description
doublequote	Value: False
lineterminator	Value: '\n'
quoting	Value: 3
skipinitialspace	Value: False

2.4 Class jobLauncher

object └
DIGEST_functions.jobLauncher

Create class able to launch a list of jobs on SLURM or LSF on a Job Scheduler

2.4.1 Methods

**__init__(self, queue, nbProcesseur, jobName, mode,
option='`select[defined(mem64)],span[hosts=1],rh5`')**

Initialize the jobLauncher class

Parameters

queue: queue name to execute jobs
(*type*=string)

nbProcesseur: number of processor used for each jobs
(*type*=integer)

jobName: job name
(*type*=string)

mode: LSF or SLURM
(*type*=string)

option: resource requirements
(default:'select[defined(mem64)],span[hosts=1]')
(*type*=string)

Overrides: object.__init__

jobArrayLauncher(*self, jobfile*)

Launch a list of jobs on SLURM or LSF on a Job Scheduler

Parameters

jobfile: file name containing one commande by line, one job by line will be launch

(*type=string*)

jobOneLauncher(*self, command*)

Launch a single job on SLURM or LSF on a Job Scheduler

Parameters

command: command line to be launched

(*type=string*)

mpiRun(*self, command*)

Launch a mpi job on SLURM or LSF on a Job Scheduler

Parameters

command: command line to be launched

(*type=string*)

Inherited from object

*__delattr__(*self*), __format__(*self*), __getattribute__(*self*), __hash__(*self*), __new__(*self*), __reduce__(*self*), __reduce_ex__(*self*), __repr__(*self*), __setattr__(*self*), __sizeof__(*self*), __str__(*self*), __subclasshook__(*self*)*

2.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<i>__class__</i>	

2.4.3 Class Variables

Name	Description
jobName	job name (<i>type=string</i>)
mode	LSF or SLURM (<i>type=string</i>)
nbProcesseur	number of processor used for each jobs (<i>type=integer</i>)

continued on next page

Name	Description
option	resource requirements (default:'select[defined(mem64)],span[hosts=1]') (<i>type</i> =string)
queue	queue name to execute jobs (<i>type</i> =string)

2.5 Class sequence

object └
DIGEST_functions.sequence

New sequence object

2.5.1 Methods

`__init__(self, header, nucl)`

Initialize the sequence class

Parameters

header: sequence header
(*type*=string)

nucl: nucleotide sequence
(*type*=string)

Overrides: object.__init__

Inherited from object

`__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
 __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()`

2.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.5.3 Class Variables

Name	Description
ID	sequence ID (<i>type=string</i>)
length	sequence length (<i>type=integer</i>)
seq	nucleotide sequence (<i>type=string</i>)

2.6 Class ORF

object └
DIGEST_functions.ORF

New ORF object from MetaGene output

2.6.1 Methods

`__init__(self, ligne)`

Initialize the ORF class

Parameters

`ligne`: ORF line from a MetaGene output file
(type=list)

Overrides: `object.__init__`

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

2.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.6.3 Class Variables

Name	Description
frame	frame (1,2 or 3) (<i>type=integer</i>)
posEND	ORF stop position in sequence (<i>type=integer</i>)
posSTART	ORF start position in sequence (<i>type=integer</i>)
score	ORF score (<i>type=float</i>)
statut	statut partial or complete (<i>type=string</i>)
strand	strand (+ or -) (<i>type=character</i>)

2.7 Class ContigORF

object └
DIGEST_functions.ContigORF

New contig ORF object from MetaGene output

2.7.1 Methods

__init__(self, ligne1, ligne2)

Initialize the ContigORF class

Parameters

ligne1: 1st line of a block in a MetaGene output file
(*type=list*)

ligne2: 2nd line of a block in a MetaGene output file
(*type=list*)

Overrides: object.__init__

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
 __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

2.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.7.3 Class Variables

Name	Description
GC	GC% in sequence (<i>type=float</i>)
ORFlist	list of ORF objects (<i>type=list</i>)
model	ORF model (bacteria, archaea or eukaryote) (<i>type=string</i>)

2.8 Class Cigar

object └─
DIGEST_functions.Cigar

New alignment CIGAR object

2.8.1 Methods

`__init__(self, strcigar)`

Initialize Cigar class

Parameters

`strcigar`: CIGAR from a SAM line
(*type=string*)

Overrides: object.`__init__`

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

2.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.8.3 Class Variables

Name	Description
<code>AlignLength</code>	number of match and mismatch (<i>type=integer</i>)
<code>Code</code>	code containing int and char compound the CIGAR (<i>type=list</i>)
<code>SumDeletion</code>	number of deletion (<i>type=integer</i>)
<code>SumInsertion</code>	number of insertion (<i>type=integer</i>)

2.9 Class alignmentSAM

object └─
DIGEST_functions.alignmentSAM

New alignmentSAM object from a line of a SAM file, see SAM format for more informations

2.9.1 Methods

<code>__init__(self, ligne)</code>
Initialize the alignmentSAM class
Parameters
<code>ligne</code> : SAM alignment line <code>(type=list)</code>

Overrides: object.`__init__`

Inherited from object

`__delattr__(self, name)`, `__format__(self, format_spec=None)`, `__getattribute__(self, name)`, `__hash__(self)`, `__new__(cls, *args, **kwargs)`, `__reduce__(self)`, `__reduce_ex__(self, reduction_callback=None)`, `__repr__(self)`, `__setattr__(self, name, value)`, `__sizeof__(self)`, `__str__(self)`, `__subclasshook__(self, other)`

2.9.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

2.9.3 Class Variables

Name	Description
CIGAR	Cigar (<i>type=Cigar</i>)
Flag	SAM flag (<i>type=integer</i>)
Length	alignment length (<i>type=integer</i>)
MAPQ	mapping quality score (<i>type=integer</i>)
PNEXT	Position of the primary alignment of the NEXT read (<i>type=string</i>)
Pos	start alignment position on the subject sequence (<i>type=integer</i>)
Query	query sequence ID (<i>type=string</i>)
QuerySequence	query nucleotide sequence (<i>type=string</i>)
RNEXT	Reference sequence name of the primary alignment of the NEXT read (<i>type=string</i>)
Subject	Subject sequence ID (<i>type=string</i>)

2.10 Class cluster

object —
DIGEST_functions.cluster

New cluster object

2.10.1 Methods

`__init__(self)`

Initialize the cluster class

Overrides: object.__init__

`computeLen(self)`

Compute the length of the cluster

`addSequence(self, clusterSequence)`

Add a clusterSequence in the list

Parameters

clusterSequence: a clusterSequence object
 $(type=clusterSequence)$

`computeMeanPourc(self)`

Compute the pourcent mean of similarity

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

2.10.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.10.3 Class Variables

Name	Description
<code>length</code>	cluster length $(type=integer)$
<code>listseq</code>	list of clusterSequence object compound the cluster $(type=list)$
<code>meanPourcSimilarity</code>	pourcent mean of similarity in the cluster $(type=float)$

continued on next page

Name	Description
ref	ID of the reference sequence (<i>type</i> =string)

2.11 Class clusterSequence

object └
DIGEST_functions.clusterSequence

New sequence in a cluster object

2.11.1 Methods

<code>__init__(self, length, header, pourc)</code>
x.__init__(...) initializes x; see help(type(x)) for signature
Parameters
length: sequence length <i>(type</i> =integer)
header: sequence header <i>(type</i> =string)
pourc: pourcent of similarity of the sequence against the cluster reference sequence <i>(type</i> =float)
Overrides: object.__init__

Inherited from object

`__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()`

2.11.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.11.3 Class Variables

Name	Description
header	sequence header <i>(type=string)</i>
length	sequence length <i>(type=integer)</i>
pourcentSimilarity	pourcent of similarity of the sequence against the cluster reference sequence <i>(type=float)</i>

3 Module ExtractFastaFromSAM

Extract overlapped paired-end reads and unmapped paired-end reads from a SAM file sorted by reads names.

Requires: DIGEST functions.py (PYTHONPATH)

Input : SAM file sorted by reads names Output : overlapped paired-end reads and unmapped paired-end reads in FASTA format

3.1 Functions

`get_parser()`

`main()`

3.2 Variables

Name	Description
<code>--doc--</code>	Value: ...
<code>--package--</code>	Value: None

4 Module SAMparser

Extract mapped or unmapped lines of SAM file with or without header and can sort result by query or subject

Requires: SamTools⁸

4.1 Functions

`get_parser()`

`main()`

4.2 Variables

Name	Description
<code>--doc--</code>	Value: ...
<code>--package--</code>	Value: None

⁸<http://sourceforge.net/projects/samtools/files/>

5 Module extendTargets_allPossibility

Extension of each sequence target with the mapped contig, if a target is mapped on several contigs, each possibility are kept.

Unmapped contig are lost.

Requires: DIGEST_functions.py (PYTHONPATH)

Input : SAM file without header only with mapped line and sort by query Output : target extended FASTA file

5.1 Functions

`get_parser()`

`main()`

5.2 Variables

Name	Description
<code>--doc--</code>	Value: ...
<code>--package--</code>	Value: None

6 Module extractORFsequences

Extract ORFs sequences in FASTA format from MetaGene output file and target extended FASTA file

Requires: DIGEST_functions.py (PYTHONPATH)

Output : PREFIX_complete.fasta and PREFIX_partial.fasta

6.1 Functions

```
get_parser()
```

```
main()
```

6.2 Variables

Name	Description
__doc__	Value: ...
__package__	Value: None

7 Module removeIdenticalSeq

Remove identical sequences from a FASTA file

Requires: hashlib⁹ python library

Input : SEQUENCE.fasta (1 line sequence) Output : PREFIX.fasta

7.1 Functions

`get_parser()`

`main()`

7.2 Variables

Name	Description
<code>--doc--</code>	Value: ...
<code>--package--</code>	Value: None

⁹<https://docs.python.org/2/library/hashlib.html>

8 Script *script_cd_hit_para_CCRT.py*

Launch cd-hit program in a parallel mode compatible with the CCRT architecture.

Requires:

- python¹⁰ (tested with 2.7.5)
- cd-hit¹¹ (tested with v4.5.8-2012-03-24)
Input : a FASTA file Output : a clustering FASTA

8.1 Functions

get_parser()

launchJobCCRT(*filename*, *jobDependency*=None)

launch the ccc_msub commande with a bash script as arguments and optionnally a job ID for the job dependency. Then this function recovers and return the job ID generated.

Parameters

filename: bash script file name
(type=string)

jobDependency: put an "after" dependency on the set of jobs defined as the argument.\ The submitted job will only be started when the jobs corresponding to the provided ids are terminated.
(type=string)

Return Value

job ID
(type=string)

¹⁰<https://github.com/lh3/seqtk>

¹¹<https://code.google.com/p/cdhit/downloads/list>

writeJobScript(*filename*, *cores*, *task*, *queue*, *projid*)

Create the job parser script header

Parameters

***filename*:** bash script file name
(type=string)

***cores*:** cores number reserved
(type=integer)

***task*:** max task number
(type=integer)

***queue*:** requested queue
(type=string)

main()

8.2 Variables

Name	Description
<code>--doc--</code>	Value: ...
<code>--package--</code>	Value: None

Index

DIGEST (*module*), 2
 DIGEST.get_parser (*function*), 2
 DIGEST.main (*function*), 2
DIGEST_functions (*module*), 3–17
 DIGEST_functions.alignmentSAM (*class*), 13–14
 DIGEST_functions.Cigar (*class*), 12–13
 DIGEST_functions.clstrParser (*function*), 3
 DIGEST_functions.cluster (*class*), 14–16
 DIGEST_functions.cluster.addSequence (*method*), 15
 DIGEST_functions.cluster.computeLen (*method*), 15
 DIGEST_functions.cluster.computeMeanFourier (*method*), 15
 DIGEST_functions.clusterSequence (*class*), 16–17
 DIGEST_functions.ContigORF (*class*), 11–12
 DIGEST_functions.exist (*function*), 3
 DIGEST_functions.fastaReader (*function*), 3
 DIGEST_functions.fileLineNumber (*function*), 5
 DIGEST_functions.geneExtended (*function*), 3
 DIGEST_functions.geneSeen (*function*), 4
 DIGEST_functions.jobLauncher (*class*), 7–9
 DIGEST_functions.jobLauncher.jobArrayLauncher (*method*), 7
 DIGEST_functions.jobLauncher.jobOneLauncher (*method*), 8
 DIGEST_functions.jobLauncher.mpiRun (*method*), 8
 DIGEST_functions.metageneParser (*function*), 4
 DIGEST_functions.MyDialect (*class*), 6–7
 DIGEST_functions.nbSequenceFasta (*function*), 5
 DIGEST_functions.ORF (*class*), 10–11
 DIGEST_functions.reverseComplement (*function*), 4
 DIGEST_functions.sequence (*class*), 9–10
 DIGEST_functions.subjectStartStop (*function*), 5
 DIGEST_functions.writeORF (*function*), 5
extendTargets_allPossibility (*module*), 20
 extendTargets_allPossibility.get_parser (*function*), 20
 extendTargets_allPossibility.main (*function*), 20
ExtractFastaFromSAM (*module*), 18
 ExtractFastaFromSAM.get_parser (*function*), 18
 ExtractFastaFromSAM.main (*function*), 18
extractORFsequences (*module*), 21
 extractORFsequences.get_parser (*function*), 21
 extractORFsequences.main (*function*), 21
removeIdenticalSeq (*module*), 22
 removeIdenticalSeq.get_parser (*function*), 22
 removeIdenticalSeq.main (*function*), 22
SAMparser (*module*), 19
 SAMparser.get_parser (*function*), 19
 SAMparser.main (*function*), 19
script_cd_hit_para_CCRT_py (*script*), 23–24
 script_cd_hit_para_CCRT_py.get_parser (*function*), 23
 script_cd_hit_para_CCRT_py.launchJobCCRT (*function*), 23
 script_cd_hit_para_CCRT_py.main (*function*), 24
 script_cd_hit_para_CCRT_py.writeJobScript (*function*), 23