DIGEST

API Documentation

October 1, 2014

Contents

1 Module DIGEST

Main script for DIGEST workflow CCRT version

Requires:

- python[1] (tested with 2.7.3)
- hashlib[2]
- BWA[3]
- RayMeta[4]
- Metagene[5]
- cd-hit[6] (tested with v4.5.8-2012-03-24)
- DIGEST_functions.py (PYTHONPATH)
- SAMparser.py
- ExtractFastaFromSAM.py
- extendTargets_allPossibility.py
- hashlib[7]fasta-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/
[5] http://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

dictionnary 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:` alignement start position of sequence on contig
(type=integer)

`alignEND:` alignement end position of sequence on contig
(type=integer)

`orfSTATUT:` ORF complete or partial
(type=string)

Return Value

True if the gene is completed, Fasle otherwise
(type=booleen)

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:` alignement start position of sequence on contig
(type=integer)

`alignEND:` alignement end position of sequence on contig
(type=integer)

Return Value

True if the gene is seen, Fasle otherwise
(type=booleen)

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:` limte length for partial ORF
(type=integer)

2.2 Variables

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

2.3 Class MyDialect

csv.Dialect
└ **DIGEST_functions.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__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

2.4.2 Properties

Name	Description
Inherited from object	
__class__	

2.4.3 Class Variables

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

continued on next page

Name	Description
option	resource requirements (default:'select[defined(mem64)],span[hosts=1]') *(type=string)*
queue	queue name to execute jobs *(type=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
Inherited from object	
__class__	

2.5.3 Class Variables

Name	Description
ID	sequence ID *(type=string)*
length	sequence length *(type=integer)*
seq	nucleotide sequence *(type=string)*

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
Inherited from object __class__	

2.6.3 Class Variables

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

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
Inherited from object	
__class__	

2.7.3 Class Variables

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

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
Inherited from object	
__class__	

2.8.3 Class Variables

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

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

__init__(*self*, *ligne*)

Initialize the alignmentSAM class

Parameters

`ligne`: SAM alignment line

(type=list)

Overrides: object.__init__

Inherited from object

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

2.9.2 Properties

Name	Description
Inherited from object	
__class__	

2.9.3 Class Variables

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

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
Inherited from object	
__class__	

2.10.3 Class Variables

Name	Description
length	cluster length *(type=integer)*
listseq	list of clusterSequence object compound the cluster *(type=list)*
meanPourcSimilarity	pourcent mean of similarity in the cluster *(type=float)*

continued on next page

Name	Description
ref	ID of the reference sequence *(type=string)*

2.11 Class clusterSequence

object
└ **DIGEST_functions.clusterSequence**

New sequence in a cluster object

2.11.1 Methods

__init__(*self*, *length*, *header*, *pourc*)

x.__init__(...) initializes x; see help(type(x)) for signature

Parameters

`length:` sequence length
(type=integer)

`header:` sequence header
(type=string)

`pourc:` pourcent of similarity of the sequence against the cluster reference sequence
(type=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
Inherited from object __class__	

2.11.3 Class Variables

Name	Description
header	sequence header *(type=string)*
length	sequence length *(type=integer)*
pourcSimilarity	pourcent of similarity of the sequence against the cluster reference sequence *(type=float)*

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
__doc__	**Value:** ...
__package__	**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: SamTools[8]

4.1 Functions

get_parser()

main()

4.2 Variables

Name	Description
__doc__	**Value:** ...
__package__	**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
__doc__	**Value:** ...
__package__	**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: hashlib[9] python library

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

7.1 Functions

get_parser()

main()

7.2 Variables

Name	Description
__doc__	**Value:** ...
__package__	**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:

- python[10] (tested with 2.7.5)
- cd-hit[11] (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
__doc__	**Value:** ...
__package__	**Value:** `None`

Index