IDP-denovo (v.2) Manual

Version v.2, by Au's lab, 2016/12/08

Website: http://www.healthcare.uiowa.edu/labs/au/IDP-denovo

Requirements:

1. x86_64 Linux operation system (CentOS-6.5, Fedora 21, Ubuntu 16.04 in test)

2. G++ (version 4.4.7, 4.9.2, 5.4.0 and 6.1.0 in test)

3. Boost library (version 1.41.0, 1.55.0 and 1.58 in test)

4. Python 2.7 and NumPy installed

Youtube video tutorial: https://youtu.be/5lkb5pVVqZg

We are welcome any comments, suggestions, questions or bug reports.

For all the enquiries, please contact Kin Fai Au: kinfaiau(at)gmail(dot)com

Abbreviation: SR-short read, LR-long read.

Contents

1. Download the package

wget "http://www.healthcare.uiowa.edu/labs/au/IDP-denovo/files/idpdenovo-v.2.zip"

After download is finished, please uncompress the package

unzip idpdenovo-v.2.zip

It will generate a folder called "idpdenovo-v.2". Please go to the code directory by running

cd idpdenovo-v.2

Inside there are directories and files:

File name	Content
bin	Directory for executable
plugins	Directory for plugins required by IDP-denovo
src	Directory for source code
test_data	Directory containing data for test, including lr_for_test(LR sequence in FASTA format), scaffold_for_test(SR-scaffold in FASTA format, Oases output), sr_1.fa and sr_2.fa (short read data in FASTA format, mate 1s and 2s)
makefile	File for installation
config_file	File including information of paths to input files and parameters
install.sh	Executable for installation
run_check.sh	Executable for checking before running IDP-denovo
run_test.sh	Executable for running IDP-denovo with example data
manual.pdf	**Manual of IDP-denovo**
backup	Directory for old files used in code development

2. Install IDP-denovo

Under code directory, please run

./install.sh

It installs one of plugins GMAP and compiles codes automatically.

When Installation finishes, under code directory, please run

./run_check.sh

It checks whether plugins can successfully run on your machine before running IDP-denovo.

3. Run IDP-denovo

To find manual of IDP-denovo, please run

```
./bin/idpdenovo.py
```

It shows

usage: idpdenovo.py [-h] [-k K_MER_LENGTH] [-f K_MER_FREQUENCY] -o OUTPUT [-t THREADS]
[--tempdir TEMPDIR | --specific_tempdir SPECIFIC_TEMPDIR]
SR_scaffold long_reads SR_left SR_right

Details are shown below.

Positional arguments	
SR_scaffold	Short read scaffolds in FASTA format
long_reads	Long reads in FASTA format
SR_left	1s left mate short reads in FASTA format
SR_right	2s right mate short reads in FASTA format
Optional arguments	
-h	show this help message and exit
-k	k-mer length used in clustering of unaligned long reads (default: 15)
-f	k-mer frequency cutoff of k-mer used in clustering of unaligned long reads (default: 0.05)
-o	REQUIRED output directory (default: None)
-t	INT number of threads to run.(default: 1)
--tempdir	The temporary directory is made and destroyed here. (default:/tmp)
--specific_tempdir	This temporary directory will be used, but will remain after executing. (default: none)

SR-scaffolds are generated by Oases with SRs, with output name of “transcripts.fa”.

There are data for test in IDP-denovo package, within directory "test_data", including files below.

File name	Content	Size
lr_for_test	LR sequence file in FASTA format	518K
scaffold_for_test	SR-scaffold sequence file in FASTA format	16K
sr_1.fa	mate 1s short read data in FASTA format	773M
sr_2.fa	mate 2s short read data in FASTA format	773M

Output files in output directory include files: lr_input, combine_seq, seq_cluster, report_seq, report.gpd, confirmed_gap, lr_quantify and sr_quantify. See details in “Explanation of output files” section.

4. Explanation of output files

Users can save temporary files by option ‘—speficic_tempdir’.

Details of final outputs are shown below.

File name	Content
lr_input	LR input in FASTA format, with original tags and new tags used in IDP-denovo
combine_seq	output of extended SR-scaffold as well as LRs and SR-scaffold unused in extension in FASTA format
seq_cluster	output of sequence clustering file
report_seq	output of pseudo-reference sequence file in FASTA format
report.gpd	splicing annotation on pseudo-reference in **GPD format**
confirmed_gap	gaps confirmed by SR alignment
lr_quantify	abundance estimation on long read counting
sr_quantify	abundance estimation of SRs, three columns separated by tab: isoform ID, isoform abundance, gene abundance

In clustering output file "seq_cluster", the format is shown below.
ID_of_representative_sequence head:
ID_of_member1:0 ID_of_member2:1 ...
"0" means member sequence is one the same strand of representative sequence, "1" means on reverse complement strand.

For example,

lr1 head:

lr2:1 lr3:0

That means: in a cluster with lr1 as presentative sequence, member lr2 is on the reverse complement strand of lr1 which lr3 is one the same strand of lr1.

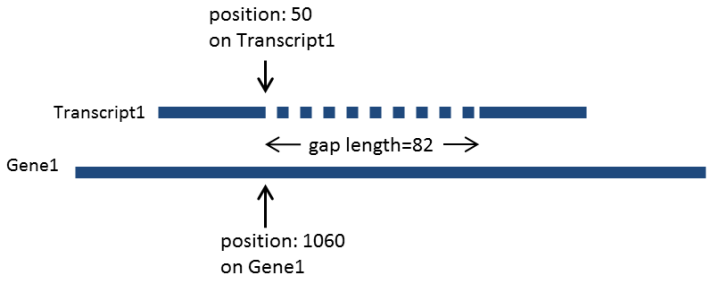
The file “confirmed_gap” contains gaps mentioned in “report.gpd” and confirmed by SR alignment. Some gaps are not confirmed by SR alignment, but they can be true with low SR coverage. The format of "confirmed_gap" is shown below:

GeneID: transcriptID <position_of_transriptID>: <skipped_length_of_pseudoRef>-<position_of_pseudoRef>

For example:

Gene1: transcript1 50:82-1060

That is, on pseudo-reference Gene1, transript1 skips region from 1060 to 1142 on pseudo-reference on position of 50 from transcript1, which suggests an alternative usage of exons.



5. Tutorial

1) Please download IDP-denovo-v.2.zip

wget “http://www.healthcare.uiowa.edu/labs/au/IDP-denovo/files/idpdenovo-v.2.zip”

2) Please uncompress “IDP-denovo-v.2.zip”.

unzip idpdenovo-v.2.zip

And go the code directory, please run

cd idpdenovo-v.2

3) Under code directory, please install IDP-denovo

./install.sh

It will install one plugin GMAP and compiles codes automatically.

4) Under code directory, please run

./run_check.sh

It checks whether plugins can run successfully on your machine before running IDP-denovo. The detailed command is

"./bin/idpdenovo.py test_data/scaffold_for_test test_data/lr_for_test test_data/sr_1.fa test_data/sr_2.fa --test -o TESTING".

5) Under code directory, please run

./run_test.sh

It runs IDP-denovo on example data in directory of "test_data".

Test data are put in directory "test_data", including files shown in table below.

File name	Content	Size
lr_for_test	LR sequence file in FASTA format	518K
scaffold_for_test	SR-scaffold sequence file in FASTA format	16K
sr_1.fa	mate 1s short read data in FASTA format	773M
sr_2.fa	mate 2s short read data in FASTA format	773M

6) Check output

The output of IDP-denovo on test data includes files shown below in directory "test_output".

File name	Content
lr_input	LR input in FASTA format, with original tags and new tags used in IDP-denovo
combine_seq	output of extended SR-scaffold as well as LRs and SR-scaffold unused in extension in FASTA format
seq_cluster	output of sequence clustering file
report_seq	output of pseudo-reference sequence file in FASTA format
report.gpd	splicing annotation on pseudo-reference in **GPD format**
confirmed_gap	gaps confirmed by SR alignment
lr_quantify	abundance estimation on long read counting
sr_quantify	abundance estimation of SRs, three columns separated by tab: isoform ID, isoform abundance, gene abundance

6. FAQs

Q1: Why did I fail to install IDP-denovo?

A1: IDP-denovo cannot be applied to all operation systems. Please forgive us for any inconvenience. We are keeping making progress to apply IDP-denovo to as many operation systems as possible. Please check system requirements. IDP-denovo has been tested in x86_64 Linux operation systems including CentOS-6.5-x86_64, Fedora 21 (Linux 3.19.7-200.fc21.x86_64+debug x86_64 and Ubuntu 16.04. NumPy need to be installed beforehand. Python version used in test is 2.7. G++ (version 4.4.7, 4.9.2 and 6.1.0 in test) and Boost library (version 1.41.0, 1.55.0 and 1.58 in test) are required. Memory required depends on size of input data. It is possible your G++ compiler cannot compile IDP-denovo codes. If your machine is x86_64 system, please run "`./backup/usebackup.sh`" to copy existed executables to directory of executables, please run "`run_check.sh`" to check whether it works.

Q2: Why did I fail to run IDP-denovo?

A2: First, please check plugins can run successfully. Please run "`./run_check.sh`" to check plugins. Second, please check whether IDP-denovo is successfully applied to example data. Please run "`./run_test.sh`" to apply IDP-denovo to example data. If it runs successfully, please check format of your input files.

Q3: Can I apply IDP-denovo to my own data?

A3: Sure! Please make sure formats of input files are correct.

SR-scaffold file in FASTA format is generated by Oases. The file name of SR-scaffold from Oases is called "transcripts.fa".

It is recommended to **correct LRs** before running IDP-denovo.

Q4: What parameters can I set in IDP-denovo?

A4: To find parameter details of IDP-denovo, please run

```
./bin/idpdenovo.py -h
```

It shows

```
usage: idpdenovo.py [-h] [-k K_MER_LENGTH] [-f K_MER_FREQUENCY] -o OUTPUT
                    [-t THREADS]
                    [--tempdir TEMPDIR | --specific_tempdir SPECIFIC_TEMPDIR]
                    SR_scaffold long_reads SR_left SR_right

IDP-denovo

positional arguments:
  SR_scaffold           Short read scaffold
  long_reads            Long reads in FASTA format
  SR_left               1s Left mate short reads reads in FASTA format
  SR_right              2s Right mate short reads reads in FASTA format

optional arguments:
  -h, --help            show this help message and exit
  -k K_MER_LENGTH, --k_mer_length K_MER_LENGTH
                        k-mer length used in clustering of unaligned long
                        reads (default: 15)
  -f K_MER_FREQUENCY, --k_mer_frequency K_MER_FREQUENCY
                        k-mer frequency cutoff (default: 0.05)
  -o OUTPUT, --output OUTPUT
                        REQUIRED Output directory (default: None)
  -t THREADS, --threads THREADS
                        INT number of threads to run. (default: 1)
  --tempdir TEMPDIR     The temporary directory is made and destroyed here.
                        (default: /tmp)
  --specific_tempdir SPECIFIC_TEMPDIR
                        This temporary directory will be used, but will remain
                        after executing. (default: None)
```

Details are shown below.

Positional arguments	
SR_scaffold	Short read scaffolds in FASTA format
long_reads	Long reads in FASTA format
SR_left	1s left mate short reads in FASTA format
SR_right	2s right mate short reads in FASTA format
Optional arguments	
-h	show this help message and exit
-k	k-mer length used in clustering of unaligned long reads (default: 15)
-f	k-mer frequency cutoff of k-mer used in clustering of unaligned long reads (default: 0.05)
-o	REQUIRED output directory (default: None)
-t	INT number of threads to run.(default: 1)
--tempdir	The temporary directory is made and destroyed here. (default:/tmp)
--specific_tempdir	This temporary directory will be used, but will remain after executing. (default: none)