HIFI-Barcode SOP – Assembling COI barcodes using high-throughput sequencing Version 3

Shanlin Liu,Chentao Yang,Chengran Zhou,Xin Zhou

Abstract

We developed an Illumina-based pipeline, HIFI-Barcode, to produce full-length COI barcodes from pooled PCR amplicons generated by individual specimens. Using indexed primer sets and high-throughput sequencing platform strategy, and optimized analysis pipeline, the analytical cost and chemistry cost will significantly be reduced. The new protocol includes DNA preparation, amplification, and data analysis pipeline.

Citation: Shanlin Liu,Chentao Yang,Chengran Zhou,Xin Zhou HIFI-Barcode SOP – Assembling COI barcodes using high-throughput sequencing. **protocols.io**
dx.doi.org/10.17504/protocols.io.k9icz4e
Published: 17 May 2018

Before start

Before starting this work, we recommend you read the paper we have published, "Shanlin Liu, Chentao Yang, Chengran Zhou, Xin Zhou; Filling reference gaps via assembling DNA barcodes using high-throughput sequencing—moving toward barcoding the world, *GigaScience*, Volume 6, Issue 12, 1 December 2017, Pages 1–8, https://doi.org/10.1093/gigascience/gix104", and if you have any question, just contact us.

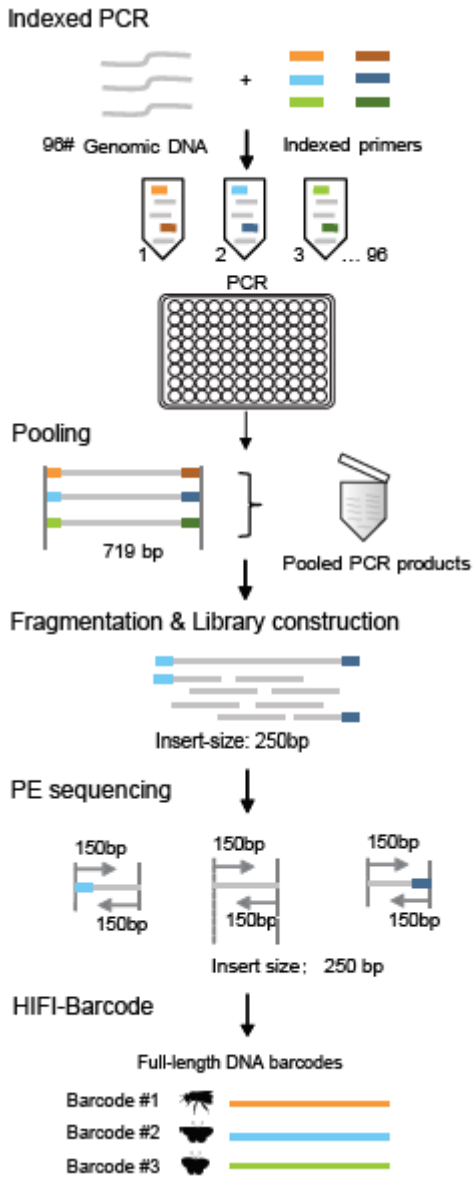
Materials

- 10X PCR buffer by Takara
- exTaq by Takara
- Cap strips by Applied Biosystems
- 10mM dNTP mix by Takara

Protocol

Overview

Step 1.



DNA preparation

Step 2.

Individual Genomic DNA could be extracted using the Glass Fiber Plate method following manufacturer's protocol or other existing method.

http://ccdb.ca/site/wp-content/uploads/2016/09/CCDB_DNA_Extraction.pdf

DNA amplification

Step 3.

3.1 PCR reaction mixture

Consumables & Equipment for PCR amplification

1.10X PCR Buffer for exTaq Store at –20°C.

2.10mM dNTP mix. Store at –20°C

3.molecular grade ddH2O

4.exTaq store at -20°C

5.Microplate(Eppendorf® plates)

6.Cap strips or sealing film

7.Eppendorf 5830R high speed centrifuge

8.primer sets.

Ninety-six pairs of different tags were added to both ends of a common COI barcode primer set (LCO1490 and HCO2198) with each tag containing 5 bps allowing for ≥ 2 bp differences from each other.

Table 1. Indexed Primer sequences

Primer	5'to3'	Primer	5'to3'
Rev001	AAAGCTAAACTTCAGGGTGACCAAAAAATCA	For001	AAAGCGGTCAACAAATCATAAAGATATTGG
Rev002	AACAGTAAACTTCAGGGTGACCAAAAAATCA	For002	AACAGGGTCAACAAATCATAAAGATATTGG
Rev003	AACCTTAAACTTCAGGGTGACCAAAAAATCA	For003	AACCTGGTCAACAAATCATAAAGATATTGG
Rev004	AACTCTAAACTTCAGGGTGACCAAAAAATCA	For004	AACTCGGTCAACAAATCATAAAGATATTGG
Rev005	AAGCATAAACTTCAGGGTGACCAAAAAATCA	For005	AAGCAGGTCAACAAATCATAAAGATATTGG
Rev006	AAGGTTAAACTTCAGGGTGACCAAAAAATCA	For006	AAGGTGGTCAACAAATCATAAAGATATTGG
Rev007	AAGTGTAAACTTCAGGGTGACCAAAAAATCA	For007	AAGTGGGTCAACAAATCATAAAGATATTGG
Rev008	AATGGTAAACTTCAGGGTGACCAAAAAATCA	For008	AATGGGGTCAACAAATCATAAAGATATTGG
Rev009	ACACTTAAACTTCAGGGTGACCAAAAAATCA	For009	ACACTGGTCAACAAATCATAAAGATATTGG
Rev010	ACAGATAAACTTCAGGGTGACCAAAAAATCA	For010	ACAGAGGTCAACAAATCATAAAGATATTGG
Rev011	ACCATTAAACTTCAGGGTGACCAAAAAATCA	For011	ACCATGGTCAACAAATCATAAAGATATTGG
Rev012	ACCTATAAACTTCAGGGTGACCAAAAAATCA	For012	ACCTAGGTCAACAAATCATAAAGATATTGG
Rev013	ACGAATAAACTTCAGGGTGACCAAAAAATCA	For013	ACGAAGGTCAACAAATCATAAAGATATTGG
Rev014	ACGTTTAAACTTCAGGGTGACCAAAAAATCA	For014	ACGTTGGTCAACAAATCATAAAGATATTGG
Rev015	ACTGTTAAACTTCAGGGTGACCAAAAAATCA	For015	ACTGTGGTCAACAAATCATAAAGATATTGG
Rev016	AGAACTAAACTTCAGGGTGACCAAAAAATCA	For016	AGAACGGTCAACAAATCATAAAGATATTGG
Rev017	AGACATAAACTTCAGGGTGACCAAAAAATCA	For017	AGACAGGTCAACAAATCATAAAGATATTGG
Rev018	AGAGTTAAACTTCAGGGTGACCAAAAAATCA	For018	AGAGTGGTCAACAAATCATAAAGATATTGG
Rev019	AGATGTAAACTTCAGGGTGACCAAAAAATCA	For019	AGATGGGTCAACAAATCATAAAGATATTGG
Rev020	AGCTTTAAACTTCAGGGTGACCAAAAAATCA	For020	AGCTTGGTCAACAAATCATAAAGATATTGG
Rev021	AGGATTAAACTTCAGGGTGACCAAAAAATCA	For021	AGGATGGTCAACAAATCATAAAGATATTGG
Rev022	AGTAGTAAACTTCAGGGTGACCAAAAAATCA	For022	AGTAGGGTCAACAAATCATAAAGATATTGG
Rev023	AGTCTTAAACTTCAGGGTGACCAAAAAATCA	For023	AGTCTGGTCAACAAATCATAAAGATATTGG
Rev024	AGTGATAAACTTCAGGGTGACCAAAAAATCA	For024	AGTGAGGTCAACAAATCATAAAGATATTGG
Rev025	AGTTCTAAACTTCAGGGTGACCAAAAAATCA	For025	AGTTCGGTCAACAAATCATAAAGATATTGG
Rev026	ATACCTAAACTTCAGGGTGACCAAAAAATCA	For026	ATACCGGTCAACAAATCATAAAGATATTGG
Rev027	ATCACTAAACTTCAGGGTGACCAAAAAATCA	For027	ATCACGGTCAACAAATCATAAAGATATTGG

Rev028	CAAAGTAAACTTCAGGGTGACCAAAAAATCA	For028	CAAAGGGTCAACAAATCATAAAGATATTGG
Rev029	CAACTTAAACTTCAGGGTGACCAAAAAATCA	For029	CAACTGGTCAACAAATCATAAAGATATTGG
Rev030	CAATCTAAACTTCAGGGTGACCAAAAAATCA	For030	CAATCGGTCAACAAATCATAAAGATATTGG
Rev031	CAGAATAAACTTCAGGGTGACCAAAAAATCA	For031	CAGAAGGTCAACAAATCATAAAGATATTGG
Rev032	CATACTAAACTTCAGGGTGACCAAAAAATCA	For032	CATACGGTCAACAAATCATAAAGATATTGG
Rev033	CATCATAAACTTCAGGGTGACCAAAAAATCA	For033	CATCAGGTCAACAAATCATAAAGATATTGG
Rev034	CCAATTAAACTTCAGGGTGACCAAAAAATCA	For034	CCAATGGTCAACAAATCATAAAGATATTGG
Rev035	CGATTTAAACTTCAGGGTGACCAAAAAATCA	For035	CGATTGGTCAACAAATCATAAAGATATTGG
Rev036	CGTATTAAACTTCAGGGTGACCAAAAAATCA	For036	CGTATGGTCAACAAATCATAAAGATATTGG
Rev037	CGTTATAAACTTCAGGGTGACCAAAAAATCA	For037	CGTTAGGTCAACAAATCATAAAGATATTGG
Rev038	CTAACTAAACTTCAGGGTGACCAAAAAATCA	For038	CTAACGGTCAACAAATCATAAAGATATTGG
Rev039	CTACATAAACTTCAGGGTGACCAAAAAATCA	For039	CTACAGGTCAACAAATCATAAAGATATTGG
Rev040	CTATGTAAACTTCAGGGTGACCAAAAAATCA	For040	CTATGGGTCAACAAATCATAAAGATATTGG
Rev041	CTCAATAAACTTCAGGGTGACCAAAAAATCA	For041	CTCAAGGTCAACAAATCATAAAGATATTGG
Rev042	CTGATTAAACTTCAGGGTGACCAAAAAATCA	For042	CTGATGGTCAACAAATCATAAAGATATTGG
Rev043	CTGTATAAACTTCAGGGTGACCAAAAAATCA	For043	CTGTAGGTCAACAAATCATAAAGATATTGG
Rev044	CTTAGTAAACTTCAGGGTGACCAAAAAATCA	For044	CTTAGGGTCAACAAATCATAAAGATATTGG
Rev045	CTTCTTAAACTTCAGGGTGACCAAAAAATCA	For045	CTTCTGGTCAACAAATCATAAAGATATTGG
Rev046	GAAACTAAACTTCAGGGTGACCAAAAAATCA	For046	GAAACGGTCAACAAATCATAAAGATATTGG
Rev047	GAACATAAACTTCAGGGTGACCAAAAAATCA	For047	GAACAGGTCAACAAATCATAAAGATATTGG
Rev048	GAATGTAAACTTCAGGGTGACCAAAAAATCA	For048	GAATGGGTCAACAAATCATAAAGATATTGG
Rev049	GACTTTAAACTTCAGGGTGACCAAAAAATCA	For049	GACTTGGTCAACAAATCATAAAGATATTGG
Rev050	GAGATTAAACTTCAGGGTGACCAAAAAATCA	For050	GAGATGGTCAACAAATCATAAAGATATTGG
Rev051	GAGTATAAACTTCAGGGTGACCAAAAAATCA	For051	GAGTAGGTCAACAAATCATAAAGATATTGG
Rev052	GATAGTAAACTTCAGGGTGACCAAAAAATCA	For052	GATAGGGTCAACAAATCATAAAGATATTGG
Rev053	GATCTTAAACTTCAGGGTGACCAAAAAATCA	For053	GATCTGGTCAACAAATCATAAAGATATTGG
Rev054	GATGATAAACTTCAGGGTGACCAAAAAATCA	For054	GATGAGGTCAACAAATCATAAAGATATTGG
Rev055	GATTCTAAACTTCAGGGTGACCAAAAAATCA	For055	GATTCGGTCAACAAATCATAAAGATATTGG
Rev056	GCAAATAAACTTCAGGGTGACCAAAAAATCA	For056	GCAAAGGTCAACAAATCATAAAGATATTGG
Rev057	GCTATTAAACTTCAGGGTGACCAAAAAATCA	For057	GCTATGGTCAACAAATCATAAAGATATTGG
Rev058	GCTTATAAACTTCAGGGTGACCAAAAAATCA	For058	GCTTAGGTCAACAAATCATAAAGATATTGG
Rev059	GGAATTAAACTTCAGGGTGACCAAAAAATCA	For059	GGAATGGTCAACAAATCATAAAGATATTGG
Rev060	GGATATAAACTTCAGGGTGACCAAAAAATCA	For060	GGATAGGTCAACAAATCATAAAGATATTGG
Rev061	GGTTTTAAACTTCAGGGTGACCAAAAAATCA	For061	GGTTTGGTCAACAAATCATAAAGATATTGG
Rev062	GTAGATAAACTTCAGGGTGACCAAAAAATCA	For062	GTAGAGGTCAACAAATCATAAAGATATTGG
Rev063	GTCATTAAACTTCAGGGTGACCAAAAAATCA	For063	GTCATGGTCAACAAATCATAAAGATATTGG
Rev064	GTGAATAAACTTCAGGGTGACCAAAAAATCA	For064	GTGAAGGTCAACAAATCATAAAGATATTGG
Rev065	GTGTTTAAACTTCAGGGTGACCAAAAAATCA	For065	GTGTTGGTCAACAAATCATAAAGATATTGG
Rev066	GTTACTAAACTTCAGGGTGACCAAAAAATCA	For066	GTTACGGTCAACAAATCATAAAGATATTGG
Rev067	GTTCATAAACTTCAGGGTGACCAAAAAATCA	For067	GTTCAGGTCAACAAATCATAAAGATATTGG
Rev068	TAAGGTAAACTTCAGGGTGACCAAAAAATCA	For068	TAAGGGGTCAACAAATCATAAAGATATTGG
Rev069	TACTGTAAACTTCAGGGTGACCAAAAAATCA	For069	TACTGGGTCAACAAATCATAAAGATATTGG
Rev070	TAGGATAAACTTCAGGGTGACCAAAAAATCA	For070	TAGGAGGTCAACAAATCATAAAGATATTGG
Rev071	TAGTCTAAACTTCAGGGTGACCAAAAAATCA	For071	TAGTCGGTCAACAAATCATAAAGATATTGG

Rev072	TATCGTAAACTTCAGGGTGACCAAAAAATCA	For072	TATCGGGTCAACAAATCATAAAGATATTGG
Rev073	TATGCTAAACTTCAGGGTGACCAAAAAATCA	For073	TATGCGGTCAACAAATCATAAAGATATTGG
Rev074	TCACATAAACTTCAGGGTGACCAAAAAATCA	For074	TCACAGGTCAACAAATCATAAAGATATTGG
Rev075	TCAGTTAAACTTCAGGGTGACCAAAAAATCA	For075	TCAGTGGTCAACAAATCATAAAGATATTGG
Rev076	TCATGTAAACTTCAGGGTGACCAAAAAATCA	For076	TCATGGGTCAACAAATCATAAAGATATTGG
Rev077	TCCAATAAACTTCAGGGTGACCAAAAAATCA	For077	TCCAAGGTCAACAAATCATAAAGATATTGG
Rev078	TCCTTTAAACTTCAGGGTGACCAAAAAATCA	For078	TCCTTGGTCAACAAATCATAAAGATATTGG
Rev079	TCGATTAAACTTCAGGGTGACCAAAAAATCA	For079	TCGATGGTCAACAAATCATAAAGATATTGG
Rev080	TCGTATAAACTTCAGGGTGACCAAAAAATCA	For080	TCGTAGGTCAACAAATCATAAAGATATTGG
Rev081	TCTCTTAAACTTCAGGGTGACCAAAAAATCA	For081	TCTCTGGTCAACAAATCATAAAGATATTGG
Rev082	TGAAGTAAACTTCAGGGTGACCAAAAAATCA	For082	TGAAGGGTCAACAAATCATAAAGATATTGG
Rev083	TGACTTAAACTTCAGGGTGACCAAAAAATCA	For083	TGACTGGTCAACAAATCATAAAGATATTGG
Rev084	TGAGATAAACTTCAGGGTGACCAAAAAATCA	For084	TGAGAGGTCAACAAATCATAAAGATATTGG
Rev085	TGCTATAAACTTCAGGGTGACCAAAAAATCA	For085	TGCTAGGTCAACAAATCATAAAGATATTGG
Rev086	TGGAATAAACTTCAGGGTGACCAAAAAATCA	For086	TGGAAGGTCAACAAATCATAAAGATATTGG
Rev087	TGTACTAAACTTCAGGGTGACCAAAAAATCA	For087	TGTACGGTCAACAAATCATAAAGATATTGG
Rev088	TGTCATAAACTTCAGGGTGACCAAAAAATCA	For088	TGTCAGGTCAACAAATCATAAAGATATTGG
Rev089	TGTGTTAAACTTCAGGGTGACCAAAAAATCA	For089	TGTGTGGTCAACAAATCATAAAGATATTGG
Rev090	TTACGTAAACTTCAGGGTGACCAAAAAATCA	For090	TTACGGGTCAACAAATCATAAAGATATTGG
Rev091	TTAGCTAAACTTCAGGGTGACCAAAAAATCA	For091	TTAGCGGTCAACAAATCATAAAGATATTGG
Rev092	TTCTCTAAACTTCAGGGTGACCAAAAAATCA	For092	TTCTCGGTCAACAAATCATAAAGATATTGG
Rev093	TTGACTAAACTTCAGGGTGACCAAAAAATCA	For093	TTGACGGTCAACAAATCATAAAGATATTGG
Rev094	TTGCATAAACTTCAGGGTGACCAAAAAATCA	For094	TTGCAGGTCAACAAATCATAAAGATATTGG
Rev095	TTGGTTAAACTTCAGGGTGACCAAAAAATCA	For095	TTGGTGGTCAACAAATCATAAAGATATTGG
Rev096	TTTCCTAAACTTCAGGGTGACCAAAAAATCA	For096	TTTCCGGTCAACAAATCATAAAGATATTGG

PCR reagents per 25 µl reaction:

#of reactions	1x	10x
ddH_2O	16.2ul	1620ul
10uM Primer forward*	1ul	100ul
10uM Primer reverse*	1ul	100ul
10X primer buffer	3ul	300ul
DNTP mix	2.5ul	250ul
exTaq	0.3ul	30ul
Total	24ul	2400ul
DNA template	1ul perl well	

General recommendations

- The use of filter tips is recommended for all PCR reagents to avoid contamination. Clean the bench top with alcohol before setting up reactions.

- Always use a sterile tip when removing exTaq polymerase and the other reagents from their tubes.
- Keep DNA templates (i.e. other PCR products) away from the PCR reagents while you are setting up the reaction mixes. Add DNA after all of the reagents have been returned to the freezer.
- Always include a sample without template as a negative control to check for contamination of the reagents. Include a positive control (a DNA sample that has amplified in the past) as well to test the effectiveness of the PCR reagents.

3.2 PCR thermocycle program

The amplification program includes a thermocycling profile of :

- 94°C for 1 min;
- 5 cycles of 94°C for 30 sec, 45°C for 40 sec, and an extension at 72°C for 1 min;
- followed by 35 cycles of 94°C for 30 sec, 51°C for 40 sec, and 72°C for 1 min;
- with a final extension at 72°C for 10 min, and finally holding at 12°C.

3.3 PCR product check

All amplicons could be visualized on a 1.2% 96 Agarose E-gel (Biowest Agarose).

REAGENTS

10X PCR buffer by Takara

10mM dNTP mix by Takara

exTaq by Takara

Sequencing

Step 4.

4.1 Hiseq or BGIseq500

All PCR products from each plate should be pooled using 1 µl per sample resulting in two 96 µl mixtures.

PCR amplicons should be fragmented to construct a library of an insert-size of 250 bp and sequenced with a strategy of 150 PE.

4.2 Pacbio (optional)

- All PCR products from each plate should be pooled using 1 µl per sample resulting in two 96 µl mixtures.
- Sequencing using PacBio RS II.

BIOINFOMATICS DATA ANALYSIS

Step 5.

5.1 HIFI-Barcode-Hiseq assembly

HIFI-Barcode-Hiseq could be downloaded from https://github.com/comery/HIFI-barcode-hiseq.

Description

HIFIBarcode is used to produce full-length COI barcodes from pooled PCR amplicons generated by individual specimens.

Installation

Clone from github

$ git clone https://github.com/comery/HIFI-barcode-hiseq.git

$ tar -zxvf HIFIBarcode.v1.0.tar.gz

Source code

(1) Wrapper script: HIFIBarcode.v1.0.pl: the wrapper script to run other Perl scripts to do the work you choose.

(2) Main scripts: Seven Perl scripts in folder HIDIBarcode.V1.0/bin/, including:

- 1_split_extract.pl
- 2_uniqu_sort_cluster.Pro.pl
- 3_sep_extract_overlap.pl
- 4_cluster_fromend.pl
- 5_forgap_filling.pl
- 6_rename_kmer.pl
- 7_final.pl

(3) Published softwares:

VSEARCH v2.4.4. Rognes, Torbjørn, et al. 'VSEARCH: a versatile open source tool for metagenomics.' PeerJ 4 (2016): e258. COPE CMR v1.0.3. Liu, Binghang, et al. 'COPE: an accurate k-mer-based pair-end reads connection tool to facilitate genome assembly.' Bioinformatics 28.22 (2012): 2870-2874. SOAPBarcode. Liu, Shanlin, et al. 'SOAPBarcode: revealing arthropod biodiversity through assembly of Illumina shotgun sequences of PCR amplicons.' Methods in Ecology and Evolution 4.12 (2013): 1142-1150.

Pre-requisites

PERL v5

Examples

1: run wrapper script to get shell text and then sh runHIFIBarcode.sh to run HIFIBarcode

perl HIFIBarcode.v1.0.pl --fq1 test_1.fq --fq2 test_2.fq --index index_primer.txt --length 5 --cpunum 10 --outdir test --outpre testout

2: run shell

sh test/runHIFIBarcode.sh

NOTE

The proposed method assemble short-read Illumina sequences based on k-mer sequence matches and such misassembly was not observed in their real data, but it's still possible. The pipeline also provides an additional note file with a suffix of "note.txt" with notes alerting users about the possibilities which have similar or same scores comparing to their best alternative.

LATEST RELEASE

Version 1.1 201709

5.2 HIFI-Barcode-Pacbio assembly

HIFI-Barcode-Pacbio could be downloaded from https://github.com/comery/HIFI-barcode-pacbio.

DESCRIPTION

HIFIBarcode is used to produce full-length COI barcodes from pooled PCR amplicons generated by individual specimens.

INSTALLATION

- Clone from github

  $ git clone https://github.com/comery/HIFI-barcode-pacbio.git

- Or go to website https://github.com/comery/HIFI-barcode-pacbio and click 'Download ZIP'

Requirements

(1) software

- PicBio smrtanalysis download from http://www.pacb.com/products-and-services/analytical-software/smrt-analysis/

(2) programming language

- standard perl
- standard python(python2 is ok)

(3) perl module

- Bio::Perl(exactlly Bio::Seq)

(4) main perl and python scripts in bin/

- primer_like_extract.pl
- cluster_count_passes_length.pl
- py | see more information to go to https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/Extracting-Reads-of-Insert-(CCS)-number-of-passes or directly download from https://raw.githubusercontent.com/PacificBiosciences/Bioinformatics-Training/master/scripts/ccs_passes.py
- pl

DATA requirements:

(1) pacbio original H5 file input

- data/*.h5

(2) primers list

- experiment_data/primer.lst

for GGTCAACAAATCATAAAGATATTGG
rev TAAACTTCAGGGTGACCAAAAAATCA

(3) index(barcodes for identifying samples) list

- experiment_data/index.xls

001 AAAGC
002 AACAG

003 AACCT
004 AACTC
005 AAGCA
...

(4) samples_location.tab

- samples name and corresponding location in 96-cell plate

1 A01
2 B01
3 C01
4 D01
5 E01
...

Overview of steps

If you installed PacBio smrtanalysis, I suppose you get the file path of setup.sh, more about Pacbio Data : http://www.pacb.com/wp-content/uploads/SMRT-Link-User-Guide-v4.0.0.pdf

e.g: setup_path='/path/PicBio/smrtanalysis/current/etc/setup.sh'

step 1 extract CCS from h5 files

Input:

my_inputs.fofn *(my_inputs.fofn contains files list of Pacbio H5 file in 01.data/, like this-> ./01.data/m170506_092957_42199_c101149142550000001823255607191735_s1_p0.1.bax.h5)*

Output:

log

data *.ccs.fasta *.ccs.fastq *.ccs.h5 reads_of_insert.fasta reads_of_insert.fastq slots.pickle

workflow

results

Run:

$ source /path/PicBio/smrtanalysis/current/etc/setup.sh

$ fofnToSmrtpipeInput.py my_inputs.fofn > my_inputs.xml

$ smrtpipe.py --params=settings.xml xml:input.xml

step 2 extract passes number from CCS h5 files

Input:

/data/*.ccs.h5

Output:

ccs_passes.lst

Run:

Note: Before run the scripts, please in sure that you have run source /path/PicBio/smrtanalysis/current/etc/setup.sh

$ python bin/ccs_passes.py data/*.ccs.h5 >ccs_passes.lst

step 3 filtering CCS by passes number (>15)

Input:

ccs_passes.lst

data/reads_of_insert.fasta

Output:

ccs_passes_15.fa

Run:

$ awk '$2>=15{print $1}' ccs_passes.lst >ccs_passes_15.lst

$ perl ./bin/fish_ccs.pl ccs_passes_15.lst data/reads_of_insert.fasta >ccs_passes_15.fa

step 4 assigning CCS to samples by index

Input:

experiment_data/primer.fa

experiment_data/index.xls

ccs_passes_15.fa

Output:

Note: 'outdir' name is up to you, here default value is '02.assignment'

02.assignment/

assign.log.txt

ccs.successfully_assigned.fa

check.ccs_passes_15.fa.log

Run:

$ perl ./bin/1.primer_like_extract.pl -p experiment_data/primer.fa -index experiment_data/index.xls -fa ccs_passes_15.fa -cm 2 -cg 1

step 5 clustering CCS of each sample to find best one

Input:

ccs.successfully_assigned.fa

check.ccs_passes_15.fa.log

ccs_passes.lst

Output:

cluster.top1.fas

cluster.id.txt

cluster.all.fa

Run:

$ cd 02.assignment/

$ perl ../bin/2.cluster_count_passes_length.pl -ccs ccs.successfully_assigned.fa -pattern check.ccs_passes_15.fa.log -passes ../ccs_passes.lst

$ perl ../bin/change_name-location.pl cluster.top1.fas >hifi-barcode-pacbio.cluster.top1.fa

ALL DONE!

So, 'hifi-barcode-pacbio.cluster.top1.fa' is final result!