***Pacific Biosciences***

***Software Functional Specification***

***compareSequences.py***

*Secondary Analysis Group*

*Yuan Li*

*Feb 26, 2014*

Contents

[1 Revision History 4](#_Toc381193928)

[2 Introduction 5](#_Toc381193929)

[2.1 Purpose 5](#_Toc381193930)

[2.2 Definitions 5](#_Toc381193931)

[2.3 References 5](#_Toc381193932)

[3 Software Overview 6](#_Toc381193933)

[3.1 Flow Chart Overview 6](#_Toc381193934)

[3.2 Product Description 8](#_Toc381193935)

[3.3 Product Functional Capabilities 10](#_Toc381193936)

[3.4 User Characteristics 10](#_Toc381193937)

[3.5 User Operations and Practices 11](#_Toc381193938)

[3.6 General Constraints 11](#_Toc381193939)

[3.7 Assumptions 11](#_Toc381193940)

[3.8 Prerequisites 11](#_Toc381193941)

[3.9 Software 12](#_Toc381193942)

[4 Functional Requirements 12](#_Toc381193943)

[4.1 PRD 12](#_Toc381193944)

[4.2 Function Specifications 12](#_Toc381193945)

[4.3 Deployment 15](#_Toc381193946)

[4.4 Development Plan 15](#_Toc381193947)

[5 External Interfaces 16](#_Toc381193948)

[5.1 User Interfaces 16](#_Toc381193949)

[5.2 Hardware Interfaces 22](#_Toc381193950)

[5.3 Software Interfaces 22](#_Toc381193951)

[5.4 Communications Interfaces 22](#_Toc381193952)

[6 Performance Requirements 22](#_Toc381193953)

[7 Attributes 22](#_Toc381193954)

[7.1 Security 22](#_Toc381193955)

[7.2 Reliability, Availability, Maintainability 22](#_Toc381193956)

[7.3 Configuration and Compatibility 22](#_Toc381193957)

[7.4 Installation 23](#_Toc381193958)

[7.5 Usability 23](#_Toc381193959)

[8 Additional Requirements 23](#_Toc381193960)

[8.1 User Documentation 23](#_Toc381193961)

[8.2 Other Requirements 23](#_Toc381193962)

[9 Appendix 23](#_Toc381193963)

[9.1 Prerequisite tools and libraries 23](#_Toc381193964)

[9.2 Substitution matrix used in phmmer 24](#_Toc381193965)

[9.3 cDNA tool kit primers 24](#_Toc381193966)

[10 References 25](#_Toc381193967)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev.** | **Author** | **Comments** |
| 2013-11-10 | 0.1.0 | Y. Li, E. Tseng | Created |
| 2013-11-18 | 0.1.1 | K. Chaturvedi, E. Tseng, J. Drake, A. Klammer and Y. Li | Reviewed & commented |
| 2013-12-19 | 0.1.2 | Y. Li | Updated flow chart |
| 2014-01-06 | 0.1.3 | Y. Li | Changed name from pbtranscript ‘ice’ to ‘cluster’. |
| 2014-02-26 | 0.1.4 | Y. Li | Included RS\_IsoSeq SMRTPortal screen shorts and reflected recent changes. |

# Introduction

## Purpose

Pacific Biosciences (PacBio) Single Molecule, Real Time (SMRT) Technology can generate high-throughput ultra-long reads at single molecule level, which provides a unique opportunity for scientists to study full-length cDNAs from high-quality polyA+ RNAs.

RS\_IsoSeq, a SMRTPortal/SMRTPipe protocol, and pbtranscript, a PacBio Bioinformatics tool, will be provided to our customers for analyzing cDNA reads sequenced by SMRT Technology.

In order to get high quality full-length reads, customers should follow the protocol described in “[Full-length cDNA Sequencing on the PacBio RS](http://www.smrtcommunity.com/servlet/servlet.FileDownload?file=00P7000000EWzydEAD)” to prepare cDNA library using cDNA kit primers, and then sequence the material using SMRT RS. Movies of SMRT Cells will then be produced, from which reads of insert (RoI, which are the insert sequence of single molecules, optionally splitting by barcode) will be extracted in order to generate high quality data for cDNA transcript analysis.

The SMRTPipe/SMRTPortal RS\_IsoSeq protocol provides two major functionalities: ***classify*** and ***cluster***, by invoking pbtranscript. pbtranscript is designed to be invoked both by the SMRTPortal/SMRTPipe RS\_IosSeq protocol, as well as by advanced customers from command line for cDNA analysis. The RS\_IsoSeq protocol has higher priority and needs to be developed first.

1. ***classify***: classification of PacBio reads into chimeric or non-chimeric, full-length (FL) or non-full-length (non-FL) reads.
2. ***cluster***: de novo isoform-level clustering of full-length, non-chimeric reads through iterative isoform clustering and error corrections (ICE). Non-full-length reads are recruited in the final consensus calling phase.

We will provide the RS\_IsoSeq protocol as part of the SMRTPortal protocols, and pbtranscript python package, as part of the PacBio Bioinformatics Software Distribution.

The pbtranscript package is written in python depending on the following code base and tools. The prerequisite tools are listed in [Appendix Table 9.1](#_Prerequisite_tools_and).

* PacBio core libraries – pbcore
* PacBio bioinformatics tools - ConsensusTools, pbdagcon[1], Quiver[2], BLASR[3]
* Third party tools – GMAP[4] and HMMER[5].

The main script of the pbtranscript package is pbtranscript.py.

## Definitions

|  |  |
| --- | --- |
| *Reads of Inserts* | RoI, the insert sequence of single molecules. |
|  |  |

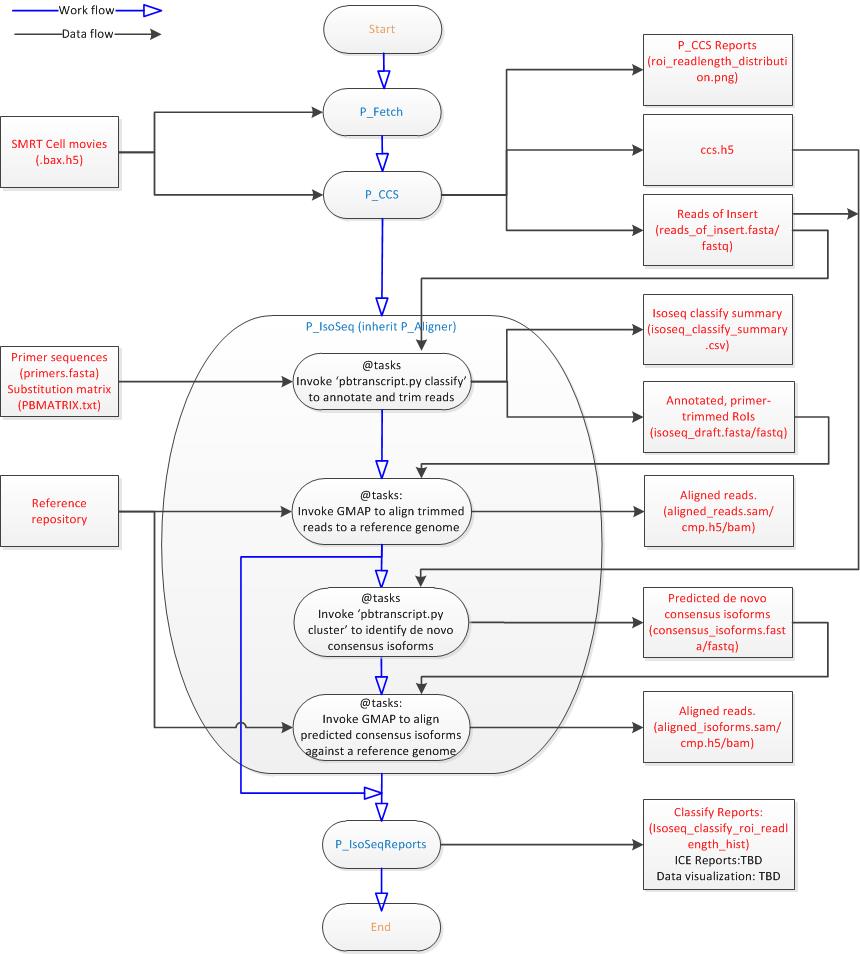
## References

ICE: Iterative Clustering for Error Correction

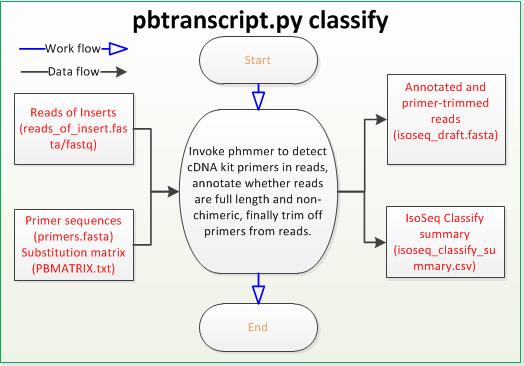
# Software Overview

## Flow Chart Overview

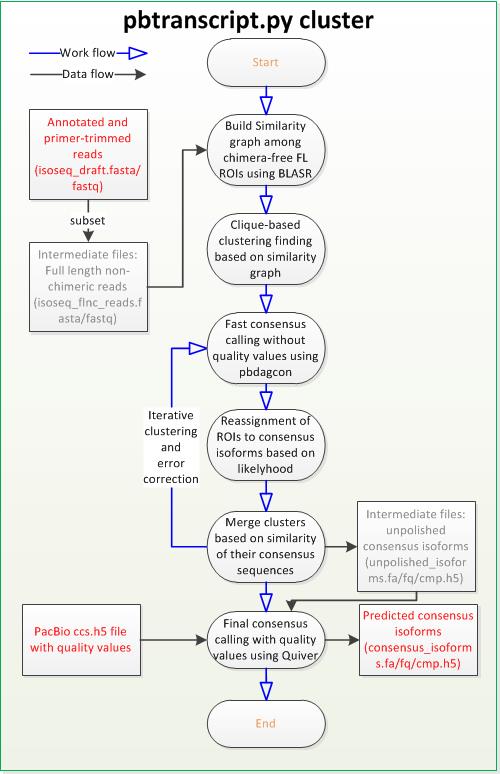
The flow chart of the RS\_IsoSeq protocol is shown in Figure 3.1. Flow charts of pbtranscript classify and cluster are shown in Figure 3.2 and Figure 3.3.



**Figure 3.1 RS\_IsoSeq Flow Chart**

****

**Figure 3.2 ‘pbtranscript.py classify’ Flow Chart**

****

**Figure 3.3 ‘pbtranscript.py cluste’ Flow Chart**

## Product Description

We will describe pbtranscript, the PacBio bioinformatics tool, in section 3.2.1 and RS\_IsoSeq, the SMRTPortal/SMRTPipe protocol in section 3.2.2, respectively.

### pbtranscript.py

pbtranscript.py is a python tool developed to analyze cDNA transcripts data generated by PacBio RS Sequencers. Two functions will be provided: ***classify*** and ***cluster***.

The pbtranscript ***classify*** function takes in PacBio reads, classifies these reads into chimeric or non-chimeric, full-length or non-full-length reads through identifying cDNA kit primers and PolyA+ tails in the reads, then trims cDNA kit primers and PolyA+ tails from reads, and finally generates classified, primer-trimmed reads with updated coordinates in the output file. The classification of PacBio reads will be summarized in a tab-delimited CSV report (e.g., classify\_summary.csv).

* The input file can only be in FASTA format, containing reads of insert of PacBio movies.
* The output file can only be in FASTA or FASTQ format, containing classified, primer-trimmed reads.
* The report is a tab-delimited CSV file with the following fields:
  + read ID, strand, is 5’ primer seen, is 3’ primer seen, is polyA+ seen, 5’ primer position, 3’ primer position, polyA+ position, index of primer detected, and is this read non-chimeric.

A typical use case should look like:

$ pbtranscript.py classify reads.fasta classified\_trimmed\_reads.fa --preport classify\_summary.csv[options]

A typical classified and primer-trimmed read:

>m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0/45/ccs strand=+;fiveseen=1;polyAseen=1;threeseen=1;fiveend=30;polyAend=3919;threeend=3945;primer=1;nonchimera=1

ATAAGACGACGCTATATGATTGGGTCACCTATGGGTGAGACAACATCCCACTTTCTCCTTATCGTATCC ……

A typical classification csv report:

ID strand fiveseen polyAseen threeseen fiveend polyAend threeend primer nonchimera

m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0/43/ccs NA 0 0 0 NA NA NA NA NA

m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0/45/ccs + 1 1 1 30 3919 3945 1 1

The pbtranscript.py ***cluster*** function takes in the classified, primer-trimmed reads from ‘pbtranscript.py ***classify***’***,*** builds unpolished consensus isoforms using the ICE algorithm to iteratively cluster full-length non-chimeric reads, and finally polishes consensus isoforms with recruited non-full-length reads using Quiver. The clustering phase will be summarized in a CSV report (e.g., cluster\_summary.csv).

A typical use case should look like:

|  |
| --- |
| $ pbtranscript.py cluster classified\_trimmed\_reads.fasta movies.ccs.h5 outdir --preport cluster\_summary.csv [options] |

Below is a typical predicted de novo consensus isoform:

\* ‘c1’, ‘52’ and ‘158’ representing the cluster id, the number of reads belonging to the cluster and read length of this consensus isoform.

|  |
| --- |
| >c1/52/158  AGACAGTGCTCACCTGTGACTGGGTGCACATCACCACCTCTGGGCCCCTTCCAGGGGTTGTCCAGCTCAGCTGCTTCGGGGGCTCTTAGCCAGGGGTCCCTTCAAGATGGCCTGGCTGTGGCTGCTCTGTGTACTGCTTCCTACCTTCATGGTGTCTG |

A typical cluster csv report:

\* each line contains a PacBio read and a cluster that the read belongs to

|  |
| --- |
| m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0/41788/31\_3992\_CCS c0  m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0/12547/31\_3964\_CCS c0  … |

### RS\_IsoSeq Protocol

The RS\_IsoSeq protocol provides a general cDNA analysis protocol in SMRTPortal and SMRTPipe. The protocol is composed of the SMRTPipe P\_Fetch, P\_CCS, P\_IsoSeq, and P\_IsoSeqReports modules, as shown in Figure 3.1.

* The P\_Fetch module fetches PacBio BAX.H5 HDF5 files.
* The P\_CCS module generates reads of insert and CCS.H5 files from BAX.H5 movies. It is also called the Filtering module.
* The P\_IsoSeq module
  + Calls ‘pbtranscript.py classify’ to classify reads into chimeric or non-chimeric, full-length or non-full-length reads, trim cDNA kit primers from reads, and summarize the classification in a CSV report.
  + Calls GMAP to map classified primer-trimmed reads to the user-specific reference genome.
  + Calls ‘pbtranscript.py cluster’ to predict de novo consensus isoforms using the ICE algorithm, polish the predicted consensus isoforms using Quiver, and summarize the prediction procedure to a CSV report.
  + Calls GMAP to map polished consensus isoforms to the user-specific reference genome.
* The P\_IsoSeqReports module
  + Plots statistics of non-chimeric, full-length, primer-trimmed reads (e.g., read length distribution).
  + TBD – plots some statistics of clustering consensus isoforms.

The user-specific reference genome is a reference repository directory produced by customers using referenceUploader (a PacBio software for uploading references to server).

Output of the RS\_IsoSeq protocols are listed in Table 3.1.



**Table 3.1 Reports and output files of RS\_IsoSeq**

## Product Functional Capabilities

The RS\_IsoSeq protocol is a SMRTPortal/SMRTPipe protocol which provides a comprehensive cDNA analysis tool kit for SMRTPortal/SMRTPipe users.

The pbtranscript tool is a PacBio bioinformatics tool which can be used by the SMRTPortal/SMRTPipe RS\_IsoSeq protocol, as well as customers who want to analyze cDNA without installing SMRTPipe and SMRTPortal.

Useful applications include discovery of new exons of known transcripts, detection and verification of fusions proteins, detection of transcript SNP, de novo prediction of novel consensus isoforms.

## User Characteristics

The SMRTPipe/SMRTPoral RS\_IsoSeq protocol provides users with convenient tools for cDNA analysis, including identification of full-length non-chimeric transcript reads from PacBio movies, de novo construction and polish of consensus isoforms, and mapping of classified reads as well as predicted consensus isoforms to user-specified reference genomes.

Users will be able to select the RS\_IsoSeq protocol in SMRTPortal/SMRTPipe for cDNA analysis. In addition, advanced users will be able to

* call ‘pbtranscript.py ***classify***’ to classify reads and trim cDNA kit primers from PacBio reads of insert
* call ‘pbtranscript.py subset’ to subset classified reads.
* call ‘pbtranscript.py ***cluster***’ to identify de novo transcript isoforms from full-length non-chimeric reads through iterative clustering and error corrections

using user-specific configurations and parameters.

## User Operations and Practices

Basic use cases of pbtranscript:

|  |
| --- |
| $ pbtranscript.py classify reads\_of\_insert.fasta isoseq\_draft.fasta  $ pbtranscript.py subset isoseq\_draft.fasta flnc.fasta –-FL –nonChimeric  $ pbtranscript.py subset isoseq\_draft.fasta nflnc.fasta –-nonFL --nonChimeric  $ pbtranscript.py cluster reads.fa ccs.fofn outdir |

Advanced use cases:

|  |
| --- |
| $ pbtranscript.py classify reads\_of\_insert.fasta isoseq\_draft.fa –-outDir your\_out\_dir –-nproc=15 –-primerFile=PRIMERFILE –-minScore=11  $ pbtranscript.py cluster flnc.fasta nflnc.fasta consensus\_isoforms.fa –-outDir your\_out\_dir –-useSGE –-maxSGEJobs=40 |

Please refer to section ‘User Interfaces’ for detailed options.

## General Constraints

The PacBio Bioinformatics tools will be installed.

## Assumptions

SMRTPortal and SMRTPipe will be installed if users want to run the RS\_IsoSeq protocol.

pbtranscript will be installed in Ubuntu > 10.02 with or without SGE environment setup, if users wants to do cDNA analysis from command line.

## Prerequisites

Python version >= 2.7 installed.

Python package numpy >= 1.6.0 installed.

Python package h5py >= 1.3.0 installed.

The PacBio Bioinformatics tools core library pbcore > 0.6.3 will be installed.

The PacBio Bioinformatics tools, ConsensusTools, pbdagcon, BLASR, Quiver and pbalign, will be installed.

HMMER 3.0 will be installed.

GMAP version > 2014-01-21 will be installed.

## Software

Appendix Table 9.1 lists all the software that may be used by pbtranscript.

# Functional Requirements

## PRD

The SMRTPortal/SMRTPipe **RS\_IsoSeq** protocol correspond to PRD-22007 and PRD-22010 for SMRTPortal 2.2 release, as shown in Table 4.1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Protocol | Functions | ID | Category | Priority | Release | Requirement Name and Description |
| RS\_IsoSeq | ***classify*** | PRD-22007 | cDNA Analysis | P1 | 2.2.0 | Pre-Q/C for cDNA analysis: incorporate Liz's scripts into SMRTAnalysis/SMRTPortal |
| ***cluster*** | PRD-22010 | Assembly | P2 | 2.2.0 | ICE (cDNA Iterative clustering for error correction): want to get accurate read for each isoform; call one consensus per isoform. |

**Table 4.1 PDR**

## Function Specifications

As shown in Figure 3.1, the RS\_IsoSeq protocol

* First, fetches PacBio reads from movies (bas/bax.h5 files) using SMRTPortal/SMRTPipe P\_Fetch module,
* Then, extracts reads of insert from bas/bax.h5 file using P\_CCS module,
* Next, calls the P\_IsoSeq module in order to:
  + Classify reads using ‘pbtranscript.py ***classify***’, and map the classified reads to the reference genome
  + Predict consensus isoforms using ‘pbtranscript.py cluster’ and map the predicted isoforms to the reference genome
* Finally, calls the P\_IsoSeqReports module to generate reports.

In the following, three key components, including reads classification, consensus isoforms clustering, and reports generation, will be described in section 4.2.1, 4.2.2 and 4.2.3 respectively.

### Reads Classification

The purpose of reads classification phase is to classify cDNA reads into chimeric or non-chimeric, full-length or non-full-length reads.

The input data of SMRTPortal/SMRTPipe P\_IsoSeq module are PacBio reads of insert produced by the P\_CCS module (or ConsensusTools) with parameters:

* minimum 0 passes and
* 75% predicted accuracy

. Each read of insert represents a best-estimated read for a sequencing ZMW. Figure 4.1 shows the schematic for reads of insert.

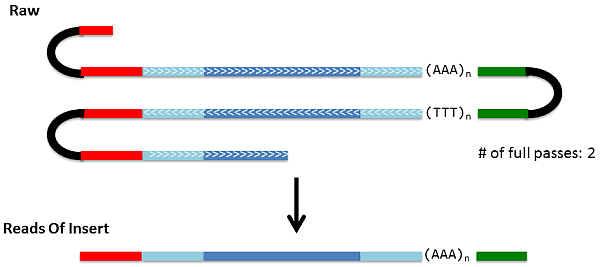


Figure 4.1 Schematic for Reads Of Insert

The P\_IsoSeq module calls ‘pbtranscript.py ***classify***’ in the reads classification phase in an effort to classify reads, detect chimeras and trim primers from reads.

* The phmmerprogram from HMMER is invoked to detect and remove the Clontech 5’ / 3’ primers (5’ – AAGCAGTGGTATCAACGCAGAGTAC – 3’). A read is considered full-length if both primers are detected at the ends with a polyA tail signal of at least 12 consecutive ‘A’s preceding the 3’ end. Based on polyA tail and 3’ primer orientation, primer-trimmed reads are reverse complemented to represent the sense strand. Because the Clontech protocol does not ensure the capture of the 5’ cap, reads are considered 3’-complete but potentially 5’-partial.
* The phmmer program is also used to detect the presence of Clontech primers at least 100 bp away from either end of the sequence, in an effort to remove artificial concatemers that may have formed via ligation of primer-attached inserts.

Note that:

* The phmmer program uses a substitution matrix specific for PacBio reads to compute hit score. The substitution matrix is described in Appendix Table 9.2.
* The primers that will be used in the PacBio cDNA tool kit are listed in Appendix Table 9.3.

When the reads classification phase of P\_IsoSeq is done, classified, primer-trimmed reads will be available at “isoseq\_draft.fasta/fastq”, and summary of the reads classification phase will be reported in “isoseq\_classify\_summary.csv” for subsequent analysis.

If a user-specific reference genome is available, all non-chimeric full-length reads will be mapped against the reference genome using GMAP. The generated alignments will be piped to ‘aligned\_reads.sam’, and then converted to ‘aligned\_reads.bam’ and ‘aligned\_reads.cmp.h5’ using pbsamtools.py and samtoh5.

### Consensus Isoforms Clustering

The purpose of consensus isoform clustering phase is to predict consensus transcript isoforms. The P\_IsoSeq module will call ‘pbtranscript.py cluster’ which implements an iterative isoform clustering algorithm called ICE (Iterative Clustering for Error Correction) developed by H. Tseng.

ICE uses PacBio sequencing QVs for determining whether two reads come from the same isoform. ICE consists of several main modules (Figure 4.2)

* clique-finding based on similarity graph
* fast consensus calling with no QV information using pbdagcon
* reassignment of sequences to different clusters based on likelihood
* final consensus calling with QV information using Quiver

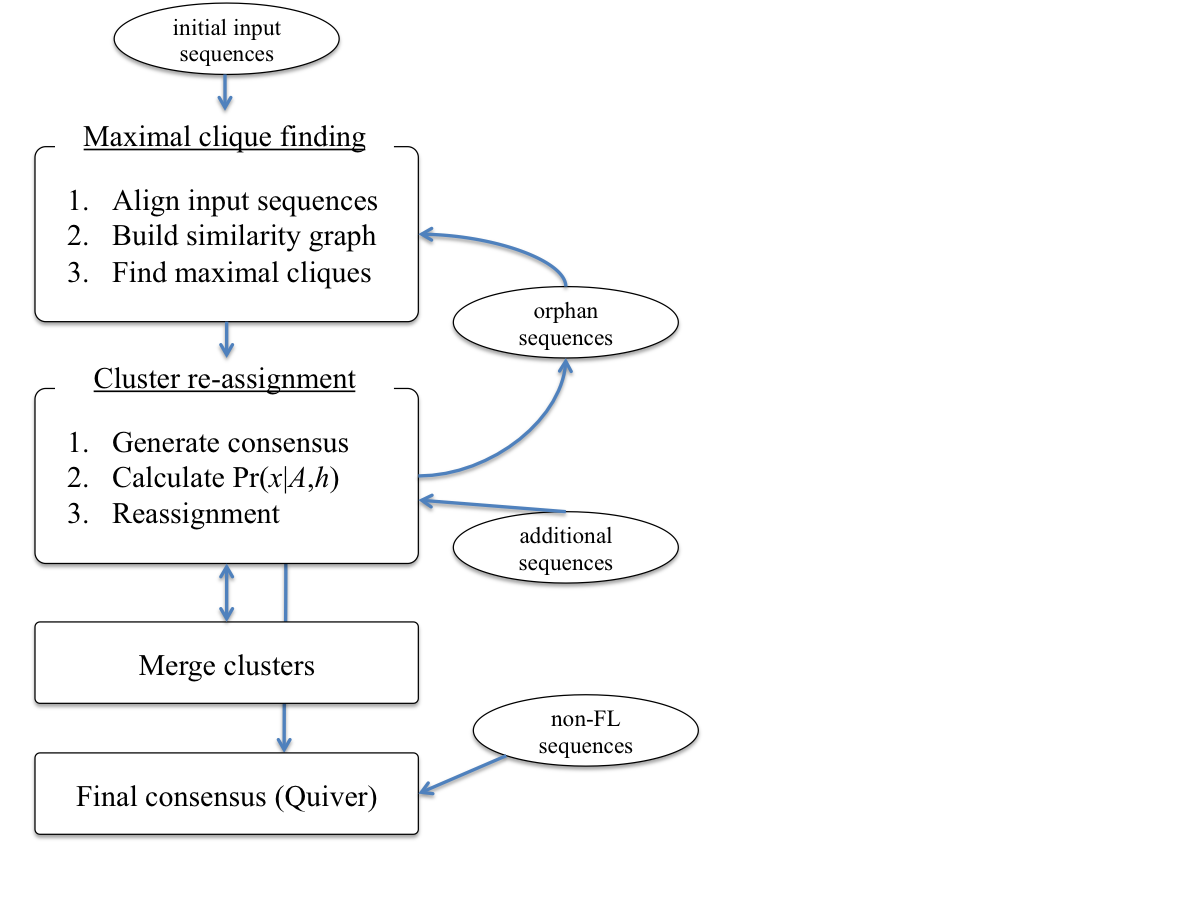


Figure 4.2 Flowchart for the ICE algorithm.

In the initial phase of clustering, the input sequence, which are often only a portion of the entire dataset, are aligned against each other using BLASR to construct a similarity graph where each node represents a read and each connecting edge indicates an “isoform hit”.

With the similarity graph constructed using isoform hits, pbtranscript looks for perfect cliques in the graph, using a maximal clique finding algorithm developed by H. Tseng. The clique finding algorithm non-deterministically finds maximal cliques in a graph, removes the clique nodes from the graph, then repeat the process until the entire graph is partitioned into mutually exclusive cliques (clusters).

Next, an initial consensus on all clusters will be constructed using pbdagcon. With the improved accuracy of the consensus sequences, the likelihood of sequences belonging to the same isoform can be better estimated. Therefore, sequences are ‘reassgined’ to clusters using a method similar to the Gibbs sampling method described in (Zagordi et al. 2010) for detecting HIV quasispecies.

At the end of the reassignments, the “orphan” sequences go through the same similarity graph construction and maximal clique finding process to form new clusters. Any cluster that has membership changes must have run through pbdagcon again.

Next, clusters will be merged according to alignment of their consensus sequences. If the consensus sequences of any two clusters are aligned against each other with highly percentage identity ( similar), they are considered an isoform hit, and will be merged together.

The final stage is to call Quiver consensus calling, non-full-length reads, which were excluded from the iterative clustering process, is recruited to improve consensus accuracy. Non-full-length reads are aligned to all pbdagcon-generated consensus sequences and filtered so that only “isoform hits” (using the same criterion as before but allowing for partial alignment) remain in the final alignment. Quiver uses the raw QVs from all aligned PacBio reads and outputs QVs along with the consensus sequence. Using the consensus QVs, we can filter out low quality consensus sequences that are often junk sequences and artifacts, though we also risk throwing out rare transcripts that have too little coverage.

When the consensus isoform clustering of P\_IsoSeq is done, predicted consensus isoforms will be available at ‘consensus\_isoforms.fasta/fastq’, and summary of the clustering phase will be reported in “cluster\_summary.csv”.

Similarly, all predicted isoforms will be mapped against the reference genome using GMAP. The generated alignments will be piped to ‘aligned\_consensus\_isoforms.sam’, and then converted to ‘aligned\_consensus\_isoforms.bam’ and ‘aligned\_consensus\_isoforms.cmp.h5’ using pbsamtools.py and samtoh5.

### Generate Reports

The P\_IsoSeqReport module is called to generates reports and plots statistics for P\_IsoSeq.

* Read length histogram of full-length, non-chimeric reads will be reported in ‘fulllength\_nonchimeric\_readlength\_hist.png’.
* Statistics of cDNA reads classification including number of reds, percentage of chimeras, number of 5’ primer detected reads, number of 3’ primer detected reads, number of polyA+ detected reads, number of full-length non-chimeric reads and their average length will be reported to ‘isoseq\_classify.json/html’.
* Read length histogram of predicted consensus isoforms will be plotted in ‘consensus\_isoforms\_readlength\_hist.png’. This report may be compared with ‘roi\_readlength\_hist.png’ and ‘fulllength-nonchimeric\_readlength\_hist.png’ for further analysis.
* Statistics of consensus isoforms, including number of clusters and average read length of consensus isoforms will be reported to ‘isoseq\_cluster.json/html’.

## Deployment

The RS\_IsoSeq protocol will be provided within SMRTPortal as a standard customer facing protocol.

The pbtranscript tool and all dependencies will be bundled in the PacBio Bioinformatics tools and installed by default.

## Development Plan

### Deployment Development

A makefile will build and install pbtranscript.

### Configuration

Parameters, exposed to users to change the behavior or performance of the RS\_IsoSeq protocol, are shown in Figure 4.3 and 4.4 and described in the following.

* Filtering 🡪 Minimum Full Passes: the minimum number of full passes to consider a ZMW.
* Filtering 🡪 Minimum Predicted Accuracy: the minimum predicted accuracy to consider a read of insert.
* IsoSeq 🡪 Minimum Sequence Length: the minimum sequence length to analyze a read.
* IsoSeq 🡪 Predict Consesnus Isoforms Using The ICE Algorithm: Whether or not to predict consensus isoforms by invoking the ICE Algorithm.
* IsoSeq 🡪 Estimated cDNA Size: The estimated cDNA length, can be ‘under 1kbp’, ‘between 1kbp to 2kbp’, ‘between 2kbp to 3kbp’, ‘above 3kbp’
* IsoSeq 🡪 Quiver: Call Quiver to polish consensus isoforms. (OFF in beta release)

### Integration

The RS\_IsoSeq protocol will be part of the standard SMRTPortal/SMRTPipe customer facing protocols.

The pbtranscript tool will be integrated in SMRTPipe.

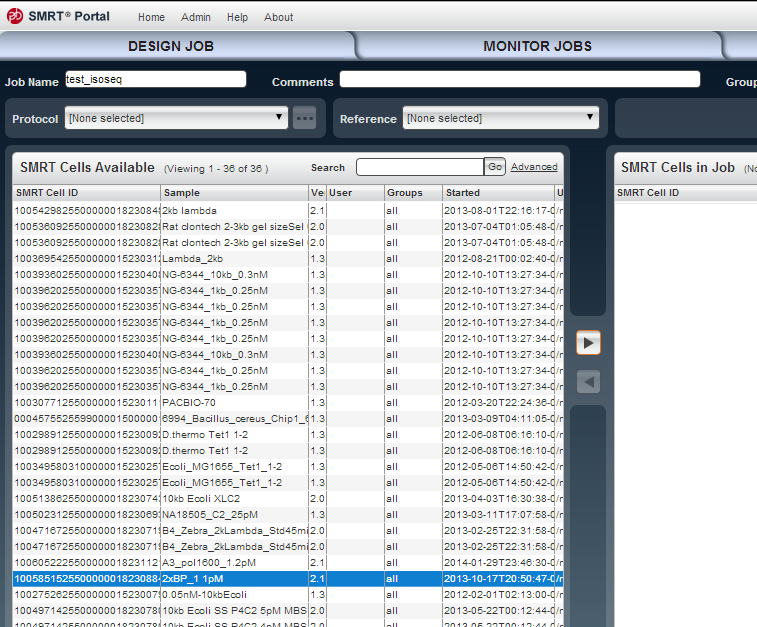
# External Interfaces

## User Interfaces

Users can create and run the RS\_IsoSeq protocol jobs using SMRTPortal or SMRTPipe. Or users can perform their own cDNA analysis manually.

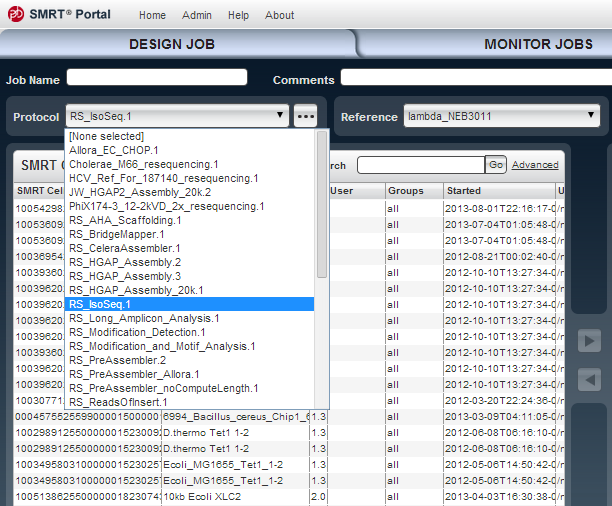
**To start a job of the RS\_IsoSeq protocol using SMRTPortal,**

* Enter Job Name and select SMRTCell movies, as shown in Figure 5.1.



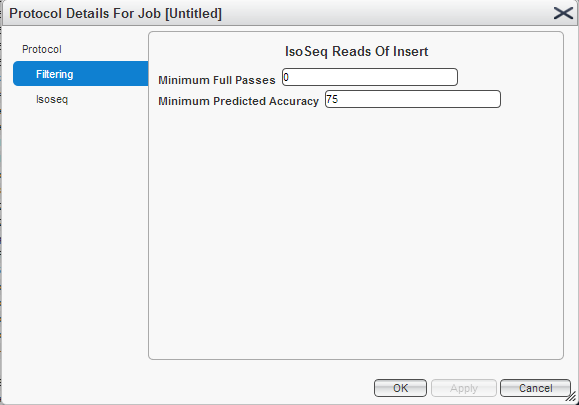
**Figure 5.1 Screenshot of Entering Job Name and Selecting Movies**

* Select RS\_IsoSeq.1 from protocols, as shown in Figure 5.2.

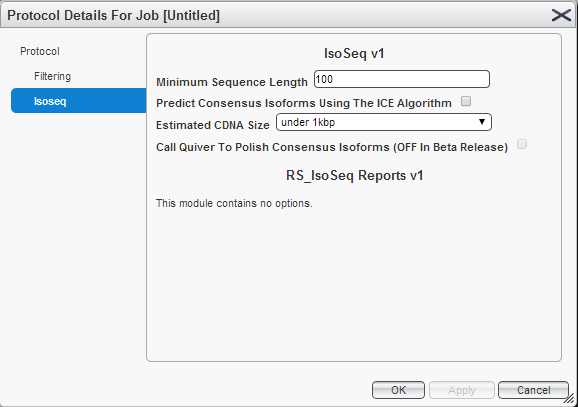


**Figure 5.2 Screenshot of Selecting the RS\_IsoSeq Protocol**

* Click ‘...’ button right next to RS\_IsoSeq.1 in order to edit selected protocol details, as shown in Figure 5.3 and Figure 5.4.
  + Modify Filtering 🡪 Minimum Full Passes and Minimum Predicted Accuracy as needed.
  + Modify IsoSeq 🡪 Minimum Sequence Length as needed.
  + Check IsoSeq 🡪 Predicted Consensus isoforms using the ICE Algorithm in order to invoke ICE.
  + Modify IsoSeq 🡪 Estimated cDNA Size as needed to optimize ICE for a particular job.

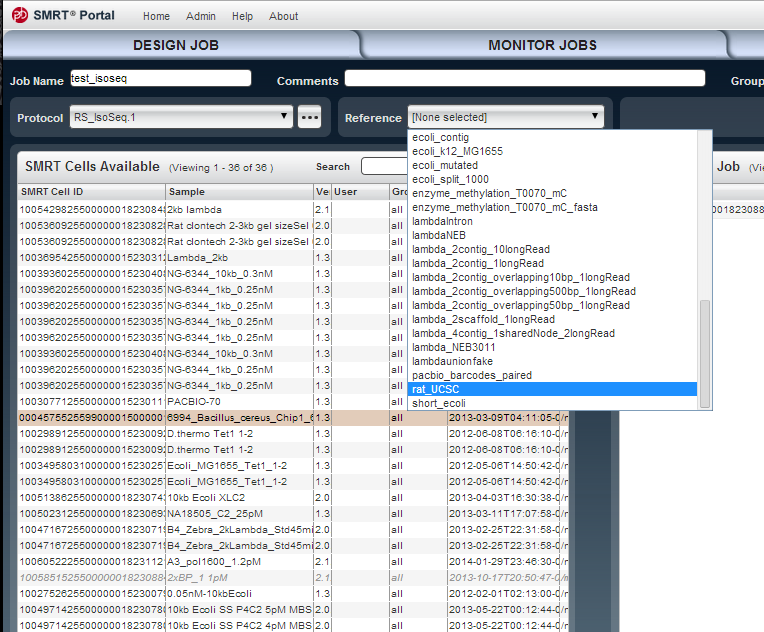


**Figure 5.3 Screenshot of RS\_IsoSeq protocol Filtering Sub-protocol**



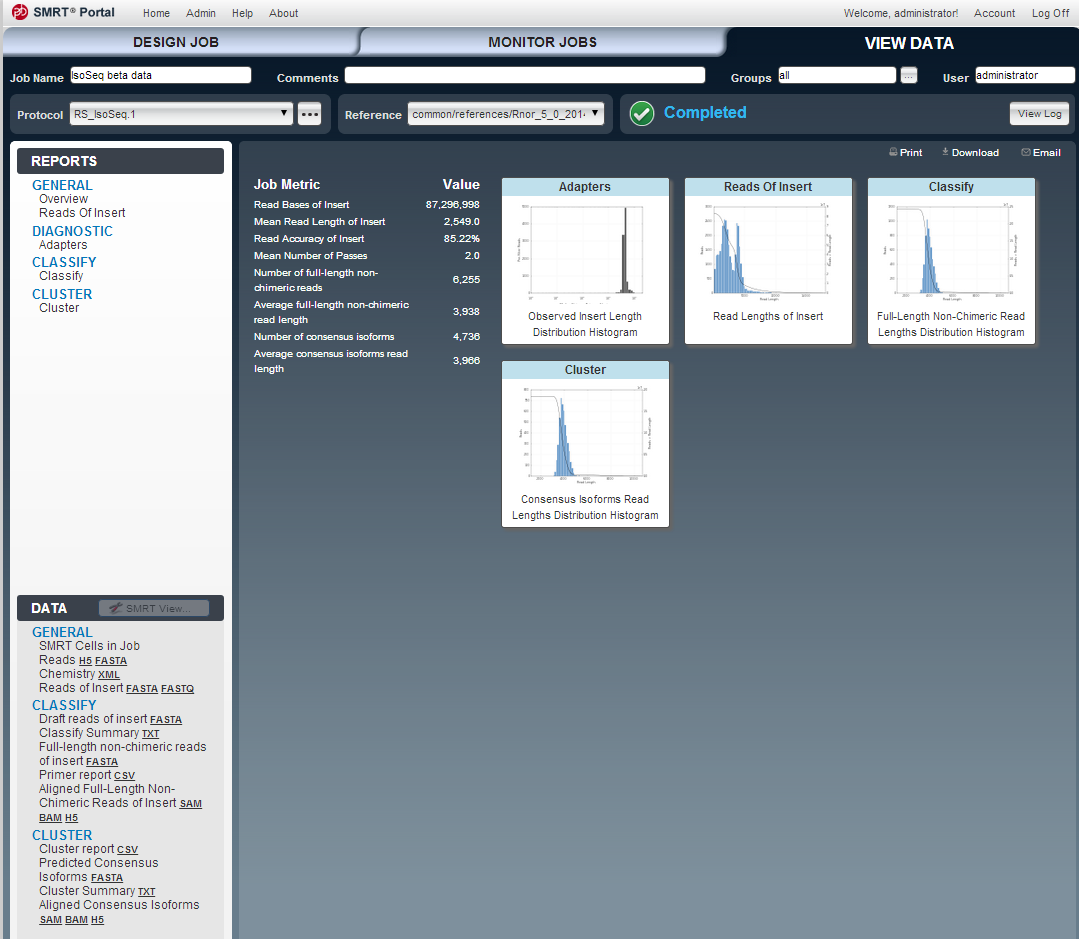
**Figure 5.4 Screenshot of RS\_IsoSeq protocol IoSeq Sub-protocol**

* Select a reference that you want transcripts to align to, as shown in Figure 5.5.



**Figure 5.5 Screenshot of selecting a reference**

* Clique ‘Save’ and ‘start’ to save and start the job.
* When the job is completed, IsoSeq reports will be displayed, as shown in Figure 5.6.



**Figure 5.6 Screenshot of RS\_IsoSeq report**

**To launch a RS\_IsoSeq protocol using SMRTPipe,**

* Specify input.xml, workflow.xml and your\_out\_dir for saving this job.
  + input.xml contains paths to the PacBio movies (i.e., bas/bax.h5 files), and is created by fofnToSmrtpipeInput.py, as shown in Table 5.1.

**Table 5.1 RS\_IsoSeq: input.xml**

|  |
| --- |
| <?xml version="1.0"?>  <pacbioAnalysisInputs><dataReferences><url ref="run:0000000-0000">  <location>/mnt/data3/vol53/2450619/0005/Analysis\_Results/m131018\_081703\_42161\_c100585152550000001823088404281404\_s1\_p0.1.bax.h5</location>  </url></dataReferences></pacbioAnalysisInputs> |

* + workflow.xml configures work flow of the job, as shown in Table 5.2

**Table 5.2 RS\_IsoSeq: workflow.xml**

|  |
| --- |
| <?xml version="1.0"?>  <smrtpipeSettings>  <global>  <param name="version"><value>1</value></param>  <param name="fetch"><value>common/protocols/preprocessing/Fetch.1.xml</value></param>  <param name="state"><value>active</value></param>  <param name="description"><value>(BETA) A general-purpose cDNA analysis workflow. </value></param>  <param name="filtering"><value>common/protocols/consensus/IsoSeq\_ReadsOfInsert.1.xml</value></param>  <param name="isoseq"><value>common/protocols/isoseq/IsoSeq.1.xml</value></param>  <param name="reference"><value>/mnt/secondary-siv/references/rat\_UCSC</value></param>  <param name="name"><value>RS\_IsoSeq</value></param>  </global>  <module name="P\_Fetch">  <param name="moduleName"><value>P\_Fetch</value></param>  <param name="description"><value>Sets up inputs</value></param>  </module>  <module name="P\_CCS">  <param name="description"><value>Generates consensus sequences from single molecules for IsoSeq.</value></param>  <param name="minPredictedAccuracy"><value>75</value></param>  <param name="minFullPasses"><value>0</value></param>  <param name="moduleName"><value>P\_CCS</value></param>  </module>  <module name="P\_IsoSeq">  <param name="moduleName"><value>P\_IsoSeq</value></param>  <param name="minSeqLen"><value>100</value></param>  <param name="cluster"><value>False</value></param>  <param name="quiver"><value>False</value></param>  </module>  <module name="P\_IsoSeqReports">  <param name="moduleName"><value>P\_IsoSeqReports</value></param>  </module>  </smrtpipeSettings> |

* + your\_out\_dir is a user-specific output directory to save all the results and reports described in Table 3.1.
* Invoke smrtpipe.py from command line, as described in Table 5.3.

**Table 5.3 Invoke SMRTPipe from command line**

|  |
| --- |
| $ smrtpipe.py --debug --distribute --params=workflow.xml \  --output=your\_out\_dir xml:input.xml 1>stdout 2>stderr & |

**To conduct cDNA analysis from command line,**

* Classify reads by invoking ‘pbtranscript.py classify’. User interface is shown in Table 5.4, an example is shown in Table 5.5.

**Table 5.4 pbtranscript.py classify: user interface**

|  |
| --- |
| usage: pbtranscript.py classify [-h] [-d OUTDIR] [-p PRIMERFN] [--cpus CPUS]  [--preport PRIMERREPORTFN]  [--summary SUMMARYFN] [--minSeqLen MINSEQLEN]  [--minScore MINSCORE]  readsFN outReadsFN  Classify reads based on whether they are non-chimeric, full-length and have  their 5', 3' and poly A tail seen.  positional arguments:  readsFN Input fasta file (usually filtered\_subreads.fasta or  reads\_of\_insert.fasta)  outReadsFN Output fasta file  optional arguments:  -h, --help show this help message and exit  HMMER options:  -d OUTDIR, --outDir OUTDIR  Directory to store HMMER output (default: output/)  -p PRIMERFN, --primer PRIMERFN  Primer fasta file (default: primers.fa)  --cpus CPUS Number of CPUs to run HMMER (default: 8)  --preport PRIMERREPORTFN  CSV file to output primer info (default:  \*.primer\_info.csv  --summary SUMMARYFN TXT file to output classsify summary (default:  \*.classify\_summary.txt  Chimera detection options:  --minSeqLen MINSEQLEN  Minimum sequence length to output (default: 50)  --minScore MINSCORE Minimum phmmer score for primer hit (default: 10) |

**Table 5.5 pbtranscript.py classify: an example**

|  |
| --- |
| $ pbtranscript.py classify –-outDir outdir \  --preport classify\_report.csv \  --summary classify\_summary.txt \  reads\_of\_insert.fasta isoseq\_draft.fasta |

* Subset classified reads into full-length non-chimeric reads and non-full-length non-chimeric reads by invoking ‘pbtranscript.py subset’. User interface is described in Table 5.6, and two examples are shown in Table 5.7.

**Table 5.7 pbtranscript.py subset: user interface**

|  |
| --- |
| usage: pbtranscript.py subset [-h] [--FL | --nonFL]  [--nonChimeric | --chimeric]  [--printReadLengthOnly]  readsFN outFN  Subset annotated reads in Fasta format.  positional arguments:  readsFN Input fasta file (usually filtered\_subreads.fasta or  reads\_of\_insert.fasta)  outFN Output fasta/txt file  optional arguments:  -h, --help show this help message and exit  --FL Reads to outut must be Full-Length, with 3' primer and  5' primer and polyA tail seen.  --nonFL Reads to output must be Non-Full-Length reads.  --nonChimeric Reads to output must be non-chimeric reads.  --chimeric Reads to output must be non-chimeric reads.  --printReadLengthOnly  Only print read lengths, no read names and sequences. |

**Table 5.7 pbtranscript.py subset: examples**

|  |
| --- |
| $ pbtranscript.py subset –-FL –-nonChimeric isoseq\_draft.fasta \  full\_length\_non\_chimeric.fasta |

* Predict consensus isoforms by invoking ‘pbtranscript.py cluster’. User interface is described in Table 5.8, and an example is shown in Table 5.9.

**Table 5.8 pbtranscript.py cluster: user interface**

|  |
| --- |
| usage: pbtranscript.py cluster [-h] [--fofn FOFNFN] [-d OUTDIR]  [--report REPORTFN] [--summary SUMMARYFN]  [--cDNASize {under1k,between1k2k,between2k3k,above3k}]  [--quiver] [--useSGE] [--maxSGEJobs MAXSGEJOBS]  [--uniqueID UNIQUEID] [--blasrNProc BLASRNPROC]  flncFa nflncFa consensusFa  Discover consensus isoforms based on quality controlled non-chimeric, full  length reads to reference genome.  positional arguments:  flncFa Input full-length non-chimeric reads in fasta format.  nflncFa Input non-full-length non-chimeric reads in fasta  format.  consensusFa Output consensus isoforms in fasta file.  optional arguments:  -h, --help show this help message and exit  --fofn FOFNFN A FOFN file ccs.h5 files, which contain quality values  reads in flncFa and nflncFa, e.g.,  reads\_of\_insert.fofn.  -d OUTDIR, --outDir OUTDIR  Directory to store temporary and output cluster  files.(default: output/)  --report REPORTFN CSV file to output cluster info (default:  \*.primer\_info.csv  --summary SUMMARYFN TXT file to output cluster summary (default:  \*.cluster\_summary.txt  ICE options:  --cDNASize {under1k,between1k2k,between2k3k,above3k}  Estimated cDNA size.  --quiver Call quiver to polish consensus isoforms using non-  full-length non-chimeric reads of insert. WARNING: not  enabled yet.  SGE environment options:  --useSGE The maximum number of jobs that will be submitted to  SGE concurrently.  --maxSGEJobs MAXSGEJOBS  The maximum number of jobs that will be submitted to  SGE concurrently.  --uniqueID UNIQUEID Unique ID for submitting SGE jobs.  --blasrNProc BLASRNPROC  Number of cores for each BLASR job. |

**Table 5.9 pbtranscript.py cluster: an example**

|  |
| --- |
| $ pbtranscript.py cluster -–fofn reads\_of\_insert.fofn \  --outDir your\_out\_dir \  --report cluster\_report.csv \  --summary cluster\_summary.txt \  full\_length\_non\_chimeric.fasta \  non\_full\_length\_non\_chimeric.fasta \  consensus\_isoforms.fasta |

### Scaling

IsoSeq classify scales linearly with the number of bases in input fasta (e.g. reads\_of\_insert.fasta).

## Hardware Interfaces

This is a part of PacBio Bioinformatics tools, it is pure software.

## Software Interfaces

Please refer to section 5.1.

## Communications Interfaces

NA.

# Performance Requirements

The RS\_IsoSeq protocol will run within SMRTPortal and SMRTPipe.   
The ‘pbtranscript.py’ python package will run under Linux with or without SGE environment.

# Attributes

## Security

NA.

## Reliability, Availability, Maintainability

PacBio will ensure reliability.

The software is part of the PacBio Bioinformatics tools, which is available in PacBio software package and also on github.

The Secondary Analysis Group is responsible for bugs. Questions can be sent to

[**devnet@pacificbiosciences.com**](mailto:devnet@pacificbiosciences.com).

## Configuration and Compatibility

### Configuration

Default configurations of protocols and software used in the RS\_IsoSeq protocol are listed in Table 7.1.

|  |  |  |
| --- | --- | --- |
| Name | Default parameters | Descriptions |
| Filtering | --minFullPasses 0  --minPredictedAccuracy 75 | Minimum full pass is 0.  Minimum predicted accuracy is 75%. |
| IsoSeq Classify | --minSeqLen 100  --minScore 10 | Minimum read lengths is 100.  Minimum phmmer score to detect a primer is 10. |
| IsoSeq Cluster | --cDNASize ‘under1k’ | Estimated cDNA size is under 1k. |

**Table 7.1 Software default configurations**

No configuration required after the standard installation.

### Compatibility

Data from 2.1 release and later will be supported. Data from pre-2.1 may be supported.

## Installation

The RS\_IsoSeq protocol will installed with SMRTPortal 2.2 and SMRTPipe 2.2.

The pbtranscript python package will be installed as part of PacBio Bioinformatics tools.

## Usability

The RS\_IoSeq protocol will be used by PacBio customers and external developers through SMRTPortal and SMRTPipe.

# Additional Requirements

## User Documentation

User documentations are in p4 depot:

//depot/software/bioinformatics/tools/pbtranscript/.

## Other Requirements

NA.

# Appendix

## Prerequisite tools and libraries

|  |  |
| --- | --- |
| Tool Name | Description |
| pbcore | The Pacbio python core library, providing IO and utilities for PacBio specific data. |
| ConsensusTools | A PacBio bioinformatics tool which will be used by the Filtering Sub-Protocol in order to generate reads of inserts from movies of SMRT cells. |
| pbdagcon | A PacBio bioinformatics tool which is a directed acyclic graph based consensus calling algorithm developed for error correcting PacBio genomic sequences. A modified version, called icedagecon will be used by IsoSeq ICE to quickly generate consensus isoforms using only full-length reads during iterations. |
| Quiver | A PacBio bioinformatics tool which generates highly accurate consensus and variant callers for PacBio reads using quality values. It will be called in the final stage of IsoSeq ICE for final consensus calling with both full-length and non-full-length reads. |
| HMMER | A biosequence analysis toolkit using profile hidden Markov Models. Phmmer will be called to detect cDNA kit primers from transcripts. |
| GMAP | A genomic mapping and alignment program for cDNA transcripts, mRNA, EST and fragmented sequences. GMAP will be called to map non-chimeric FL reads to reference genomes specified by users. |
| BLASR | An aligner designed for fast mapping PacBio reads to references. BLASR will be called to align classified reads against themselves as well as predicted consensus isoforms. |

**Appendix Table 9.1 Prerequisite tools and libraries**

## Substitution matrix used in phmmer

phmmer uses a substitution matrix specific for PacBio reads to calculate scores. The default substitution matrix is saved in PBMATRIX.txt, as shown in Appendix Table 9.2.

# Matrix made by matblas from blosum62.iij

# \* column uses minimum score

# BLOSUM Clustered Scoring Matrix in 1/2 Bit Units

# Blocks Database = /data/blocks\_5.0/blocks.dat

# Cluster Percentage: >= 62

# Entropy = 0.6979, Expected = -0.5209

A R N D C Q E G H I L K M F P S T W Y V B Z X \*

A 4 -1 -2 -2 -2 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1 1 -2 -3 -2 0 -2 -1 0 -4

R -1 5 0 -2 -3 1 0 -2 0 -3 -2 2 -1 -3 -2 -1 -1 -3 -2 -3 -1 0 -1 -4

N -2 0 6 1 -3 0 0 0 1 -3 -3 0 -2 -3 -2 1 0 -4 -2 -3 3 0 -1 -4

D -2 -2 1 6 -3 0 2 -1 -1 -3 -4 -1 -3 -3 -1 0 -1 -4 -3 -3 4 1 -1 -4

C -2 -3 -3 -3 4 -3 -4 -2 -3 -1 -1 -3 -1 -2 -3 -1 -2 -2 -2 -1 -3 -3 -2 -4

Q -1 1 0 0 -3 5 2 -2 0 -3 -2 1 0 -3 -1 0 -1 -2 -1 -2 0 3 -1 -4

E -1 0 0 2 -4 2 5 -2 0 -3 -3 1 -2 -3 -1 0 -1 -3 -2 -2 1 4 -1 -4

G -2 -2 0 -1 -2 -2 -2 4 -2 -4 -4 -2 -3 -3 -2 0 -2 -2 -3 -3 -1 -2 -1 -4

H -2 0 1 -1 -3 0 0 -2 8 -3 -3 -1 -2 -1 -2 -1 -2 -2 2 -3 0 0 -1 -4

I -1 -3 -3 -3 -1 -3 -3 -4 -3 4 2 -3 1 0 -3 -2 -1 -3 -1 3 -3 -3 -1 -4

L -1 -2 -3 -4 -1 -2 -3 -4 -3 2 4 -2 2 0 -3 -2 -1 -2 -1 1 -4 -3 -1 -4

K -1 2 0 -1 -3 1 1 -2 -1 -3 -2 5 -1 -3 -1 0 -1 -3 -2 -2 0 1 -1 -4

M -1 -1 -2 -3 -1 0 -2 -3 -2 1 2 -1 5 0 -2 -1 -1 -1 -1 1 -3 -1 -1 -4

F -2 -3 -3 -3 -2 -3 -3 -3 -1 0 0 -3 0 6 -4 -2 -2 1 3 -1 -3 -3 -1 -4

P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4 7 -1 -1 -4 -3 -2 -2 -1 -2 -4

S 1 -1 1 0 -1 0 0 0 -1 -2 -2 0 -1 -2 -1 4 1 -3 -2 -2 0 0 0 -4

T -2 -1 0 -1 -2 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1 1 4 -2 -2 0 -1 -1 0 -4

W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1 1 -4 -3 -2 11 2 -3 -4 -3 -2 -4

Y -2 -2 -2 -3 -2 -1 -2 -3 2 -1 -1 -2 -1 3 -3 -2 -2 2 7 -1 -3 -2 -1 -4

V 0 -3 -3 -3 -1 -2 -2 -3 -3 3 1 -2 1 -1 -2 -2 0 -3 -1 4 -3 -2 -1 -4

B -2 -1 3 4 -3 0 1 -1 0 -3 -4 0 -3 -3 -2 0 -1 -4 -3 -3 4 1 -1 -4

Z -1 0 0 1 -3 3 4 -2 0 -3 -3 1 -1 -3 -1 0 -1 -3 -2 -2 1 4 -1 -4

X 0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2 0 0 -2 -1 -1 -1 -1 -1 -4

\* -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 1

**Appendix Table 9.2 Substitution Matrix for PacBio reads**

## cDNA tool kit primers

primers.fa should contain primers that will be used in the cDNA library preparation. Two default sets of primers for PacBio cDNA tool kits are in shown Appendix Table 9.3.

>F0

TCGTCGGGGACAACTTTGTACAAAAAAGTTGGATGGGG

>R0

AAAACCCAACTTTCTTGTACAAAGTTGTCCCC

>F1

AAGCAGTGGTATCAACGCAGAGTACATGGGG

>R1

GTACTCTGCGTTGATACCACTGCTT

**Appendix Table 9.3 cDNA tool kit primers**

# References

[1] Chin, Chen-Shan, David H Alexander, Patrick Marks, Aaron A Klammer, James Drake, Cheryl Heiner, Alicia Clum, et al. 2013. “Nonhybrid, Finished Microbial Genome Assemblies from Long-read SMRT Sequencing Data.” *Nature Methods* 10 (6) (May 5): 563–569. doi:10.1038/nmeth.2474.

pbdagcon: <https://github.com/PacificBiosciences/pbdagcon>

[2] Chin, CS, Alexander, DH, Marks, P, Klammer, AA, Drake, J, Heiner, C, Clum, A, Copeland, A, Huddleston, J, Eichler, EE, Turner, SW, Korlach, J **(2013)**. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat. Methods, 10, 6:563-9.

Quiver: <https://github.com/PacificBiosciences/GenomicConsensus>

[3] Chaisson, Mark J, and Glenn Tesler. 2012. “Mapping Single Molecule Sequencing Reads Using Basic Local Alignment with Successive Refinement (BLASR): Application and Theory.” *BMC Bioinformatics* 13 (1): 238. doi:10.1186/1471-2105-13-238.

BLASR: <https://github.com/PacificBiosciences/blasr>

[4] Thomas D. Wu and Serban Nacu. Fast and SNP-tolerant detection of complex variants and splicing in short reads. Bioinformatics 2010 26: 873-881

GMAP: <http://research-pub.gene.com/gmap/>

[5] The theory behind profile HMMs: R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, [*Biological sequence analysis: probabilistic models of proteins and nucleic acids*](http://selab.janelia.org/cupbook.html), Cambridge University Press, 1998.

HMMER: <https://hmmer.janelia.org/>