QuickIsoSeq for isoform quantification in large-scale RNA sequencing

Shanrong Zhao

Pfizer Worldwide Research and Development, Cambridge, MA 02139

January 2020

Outline

1. Overview of *QuickIsoSeq*

QuickIsoSeq is built upon multiple best-in-the-class open source tools and generate many useful QC metrices to support practical RNA-seq data analysis, in addition to isoform quantification. It is particularly designed to be run in a high-performance computing cluster (HPC) environment for large-scale RNA-seq projects. The goal of *QuickIsoSeq* is to streamline the process of isoform quantification and improve efficiency and reproducibility in RNA-seq data analysis.

1.1 Computational algorithms for isoform quantification

A number of packages have been developed to quantify expression at the transcript level including RSEM [1], eXpress [2], TIGAR2 [3], and Cufflinks [4]. Most Recently, ultra-fast alignment-free methods, such as Sailfish [5], Salmon [6] and Kallisto [7] have been developed by exploiting the idea that precise alignments are not required to assign reads to their origins. After a comprehensive evaluation of those seven packages for isoform quantification [8], Salmon, Kallisto and RSEM are the three recommended tools considering both the accuracy and computational resources needed, and thus incorporated into our *QuickIsoSeq* pipeline.

1.2 The architecture of *QuickIsoSeq*

The flowchart of *QuickIsoSeq* is shown in **Figure 1.** The inputs to the pipeline are raw sequence reads and reference genome/transcriptome. Step #1 performs RNA-seq read mapping, SNP calling and isoform quantification. This step is computationally intensive and processes each sample completely independently of each other. Therefore, all samples can be processed in parallel in an HPC environment. Step #2 merges the analysis results from the individual samples and generates different kinds of QC metrics. All QC metrices are available in tab delimited text files, as well as in plots generated by ggplot2.

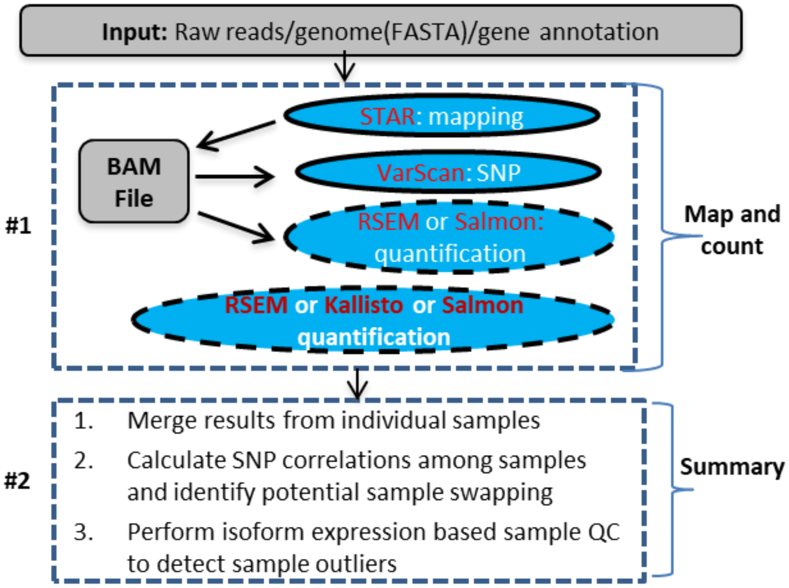
Input: Raw reads/genome(FASTA)/gene annotation
#1
BAM File
STAR: mapping
VarScan: SNP
RSEM or Salmon: quantification
RSEM or Kallisto or Salmon quantification
Map and count
#2
1. Merge results from individual samples
2. Calculate SNP correlations among samples and identify potential sample swapping
3. Perform isoform expression based sample QC to detect sample outliers
Summary

Figure 2. Overview of the *QuickIsoSeq* pipeline. Step #1 is computationally intensive, and processes individual samples independently. Step #2 integrates RNA-seq data analysis results from the individual samples in Step #1 and generates comprehensive QC metrices.

2. Deploy *QuickIsoSeq* in your own computational environment

2.1. Install third-party open source tools

After you download and unpack *QuickIsoSeq* package from GitHub, set environment variable **QuickIsoSeq** to the folder where the *QuickIsoSeq* package is installed. All required third-party tools are listed in the table below. You can install those tools manually or run ***install-tools.sh*** that automates the installation of all tools. It takes <10min to run ***install-tools.sh***. Additionally, R version 3.2 or higher is required in your local computational environment. The ggplot2 and reshape2 R libraries should be installed.

Tool	**Function**	**Source**
STAR	Read alignment/mapping	https://github.com/alexdobin/STAR
FeatureCounts	Count reads	http://subread.sourceforge.net/
VarScan	Variant calling	https://github.com/dkoboldt/varscan
Samtools	Manipulate read alignments	https://github.com/samtools/samtools
RSEM	Isoform quantification	https://github.com/deweylab/RSEM/
Salmon	Isoform quantification	https://github.com/COMBINE-lab/salmon
Kallisto	Isoform quantification	https://github.com/pachterlab/kallisto
Bowtie	Read alignment	https://sourceforge.net/projects/bowtie-bio/
gffread	Extract transcript sequences	http://ccb.jhu.edu/software/stringtie/dl/

By default, ***install-tools.sh*** will install all tools under **$*QuickIsoSeq*/Tools** folder and generate a file **tools_path.txt** to record the locations of all the installed packages. You can append all the contents in **tools_path.txt** into your **run.config** later. The contents in this file look as follows.

```
APPLICATION_ROOT=${QuickIsoSeq}/Tools
STAR=$APPLICATION_ROOT/STAR_2.7.3a/bin/Linux_x86_64_static
FEATURECOUNTS=$APPLICATION_ROOT/subread-2.0.0/bin
VARSCAN_JAR=$APPLICATION_ROOT/VarScan.v2.4.0.jar
SAMTOOLS=$APPLICATION_ROOT/samtools-1.9/bin
RSEM=$APPLICATION_ROOT/RSEM-1.3.1
SALMON=$APPLICATION_ROOT/salmon-1.1.0/bin
KALLISTO=$APPLICATION_ROOT/kallisto
GFFREAD=$APPLICATION_ROOT/gffread-0.11.4.Linux_x86_64
BOWTIE=$APPLICATION_ROOT/bowtie-1.2.3-linux-x86_64
export
PATH=$SAMTOOLS:$STAR:$FEATURECOUNTS:$RSEM:$SALMON:$KALLISTO:$GFFR
EAD:$BOWTIE:$PATH
```

2.2. Prepare index files for different species

A reference genome in FASTA format and an isoform annotation file in GTF format are needed to create index files for isoform quantification. To simply this step, an example script ***create_indexes.GRCh38_Genecode30.sh*** is provided, which creates all the index files for human genome GRCh38 and Gencode Release version 30. It takes about 1hr to run this script. You can use this script to create indexes for other species and annotations as well, and all you need to change are input files (i.e. the reference genome and gene annotation files). The main functions of this script are summarized as follows.

1. Download a genome fasta file, unzip and rename it to **genome.fa**
2. Download a gene annotation in GTF format, unzip and rename it to **gene.gtf**
3. Extract sequences for all transcript in gene.gtf, and generates **transcript.fa**
4. Parse gene.gtf to get the corresponding annotations **gene.annot** and **transcript.annot**
5. Create genome index files for STAR, RSEM, Salmon, and Kallisto, respectively
6. Create bowtie indexes for rRNA and hemoglobin transcripts, respectively
7. Define the genomic region **CHR_REGION** for SNP calls. In the human genome, chr6 is recommended where the MHC region is located. For mouse, it is chr17.

After the above steps are completed, you are ready for analyzing your own dataset using *QuickIsoSeq*. The bash script ***create_indexes.GRCh38_Genecode30.sh*** outputs the locations for all indexed files into the file **${INDEX_ROOT}/ indexes_path.txt**. Its contents are something like below, and you can append them into your **run.config** later.

```
SAMPLE_SPECIES=human
INDEX_ROOT=${QuickIsoSeq}/Indexes/GRCh38_Genecode30
GENOME_FASTA=$INDEX_ROOT/genome.fa
GTF_FILE=$INDEX_ROOT/gene.gtf
TRANSCRIPT_FASTA=$INDEX_ROOT/transcript.fa
TRANSCRIPT_ANNOTATION=$INDEX_ROOT/transcript.annot
GENE_ANNOTATION=$INDEX_ROOT/gene.annot
CHR_REGION=chr6:1-170805979
STAR_INDEX=${INDEX_ROOT}/STAR_100
RSEM_INDEX=${INDEX_ROOT}/rsem/rsem
```

```
SALMON_INDEX=${INDEX_ROOT}/salmon
KALLISTO_INDEX=${INDEX_ROOT}/kallisto/index
rRNA_BWT_INDEX=${INDEX_ROOT}/bowtie_rRNA/rRNA
hgRNA_BWT_INDEX=${INDEX_ROOT}/bowtie_hgRNA/hgRNA
```

3. Run *QuickIsoSeq* to analyze your own RNA-seq

After downloading *QucikIsoSeq*, you will see a directory named "test_run". This is an example project directory. This "test_run" directory contains four files below. You are recommended to copy these files to any empty working folder and then tailor them to your own computational environment and RNA-seq project.

1) allIDs.txt: sample identifiers to be processed
2) sample.annotation.txt: an annotation file for RNA samples
3) run.config: run configuration file
4) master-cmd.sh: command lines for run *QuickIsoSeq*. This is simply for convenience so that you can cut-and-paste the command lines and run them.

3.1. Prepare sample annotation, sample ID file and run.config

Sample annotation is optional, but it is strongly recommended to prepare a meaningful annotation file to capture important meta data for RNA samples. A proper annotation file should be in tab delimited text format. The first and second columns correspond to sample and subject identifiers, respectively. The sample.annotation.txt file in test_run directory has columns as "Run", "subject_id", "histological_type", and "sex". The second column is used for checking potential sample swapping and all RNA samples from a same subject are supposed to have very high SNP concordance, whereas the SNP concordance is lower for a pair of samples from different subjects. Sample ID file contains one unique sample ID per line. There is no column header. The allIDs.txt file in test_run directory lists all 48 samples in this demo project. For an RNA-seq project, only those samples in the sample ID file are processed by *QuickIsoSeq*.

The **run.config** controls how *QuickIsoSeq* to run. It consists of three parts. Part #1 contains RNA-seq sequencing project specific information; Part #2 provides species-specific indexes, reference genome and gene annotation, generated by the step of preparing index files for different species; and Part #3 set the locations of tools and software specific parameters. For a given species, Parts #2 and #3 in **run.config** usually remain the same across different RNA-seq projects, but Part #1 varies from project to project, and you have to set the paramters properly. The most important parameters in Part #1 are briefly described as follows.

- ***FASTQ_DIR***: the directory where the fastq files are located.
- ***FASTQ_SUFFIX***: a fastq file typically ends with fastq.gz, fq.gz, fastq or fq
- ***STRAND***: non-stranded:0; first read forward strand:1; first read reverse strand:2
- ***SEQUENCE_TYPE***: "pair" for pair-ended reads; "single" for single-ended sequencing
- ***SEQUENCE_DEPTH***: set it to "regular" if <80 million reads; otherwise set it to "deep"
- ***ISOFORM_ALGORITHM***: set it to RSEM, KALLISTO, SALMON_ALN, SALMON or ALL

3.2. Process and analyze individual samples

Under your project folder, invoke mapping, quantitation, QC, and SNP calling for each sample by running **run-isoseq.sh**. Because this step is computationally intensive, it is advised to run this command on an HPC cluster. The **run-isoseq.sh** can be run off-shelf if your HPC uses LSF as the job scheduler. Otherwise, for a cluster using a job scheduler other than LSF, **run-isoseq.sh** needs to be twisted or modified. A separate result folder will be created for each sample under the project folder. Below is the example command line.

```
export QuickIsoSeq=<Your QuickIsoSeq Install Folder>
export PATH=$QuickIsoSeq:$PATH
run-isoseq.sh allIDs.txt run.config
```

3.3. Post-processing: merge results from individual samples

As in the previous step, this step also runs under the project directory. We run the merging and summarization step when all jobs to process individual samples are finished. This step not only merges results from individual samples and generates combined results such as counts table, but also generate a variety of QC metrics for stringent sample QC. All QC outputs are available in both plain tab delimited text files and plots. Below are commands to run to generate the summary results and QC metrices.

```
export QuickIsoSeq=<Your QuickIsoSeq Install Folder>
ml RHEL6-apps R/3.2.3
merge-isoseq.sh allIDs.txt run.config &> Results.log
```

The default output folder is **Results/Summary** unless you provide a different output folder name in your command line as the 3rd parameter when running **merge-isoseq.sh**. The main output files are as follows.

- The library size, the summary of reads mapping and counting for individual samples
- The breakdown of sequence reads: rRNA, hemoglobin and other
- SNP concordance among samples to detect potential sample swapping
- Merged counts table from isoform quantification algorithms
- The number of expressed genes or transcripts at different TPM cut-offs
- The top highly expressed genes and the percentage of genes from mitochondria
- The correlations of expression profiles among all samples and potential outliers

Summary: By combing the best open source tool sets, we implemented the *QuickIsoSeq* pipeline, which significantly reduces the efforts involved in primary RNA-seq data analyses. The configuration file contains project, species, and software related parameters, and thus improves the reproducibly in RNA-seq data analyses. RSEM, Salmon and Kallisto have different input requirements and output formats, *QuickIsoSeq* takes care of the difference, and for end users. *QuickIsoSeq* is a unifying workflow for isoform quantification no matter which algorithm is chosen. We have already applied *QuickIsoSeq* to multiple in-house large-scale RNA-seq projects, and its current version is stable and mature for public release and adoption.

References

1. Li, B. and C.N. Dewey, *RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome.* BMC Bioinformatics, 2011. **12**.
2. Roberts, A. and L. Pachter, *Streaming fragment assignment for real-time analysis of sequencing experiments.* Nat Methods, 2013. **10**.
3. Nariai, N., et al., *TIGAR2: sensitive and accurate estimation of transcript isoform expression with longer RNA-Seq reads.* BMC Genomics, 2014. **15**.

4. Trapnell, C., et al., *Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation.* Nat Biotechnol, 2010. **28**.

5. Patro, R., S.M. Mount, and C. Kingsford, *Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms.* Nat Biotechnol, 2014. **32**.

6. Patro, R., et al., *Salmon provides fast and bias-aware quantification of transcript expression.* Nat Methods, 2017. **14**.

7. Bray, N.L., et al., *Near-optimal probabilistic RNA-seq quantification.* Nat Biotechnol, 2016. **34**.

8. Zhang, C., et al., *Evaluation and comparison of computational tools for RNA-seq isoform quantification.* BMC Genomics, 2017. **18**(1): p. 583.