# RNA-seq Analysis Package for SOLiD

## Contents

# 1 Introduction

RNA-seq Analysis package for SOLiD can process the RNA-seq analysis for SOLiD data. It is developed by two languages: perl and java.

The RNA-seq analysis pipeline for SOLiD contains 6 steps:

1. Filter low quality reads
2. Mapping
3. Annotation
4. DEGseq analysis
5. GO analysis
6. KEGG analysis

The tool uses PBS to schedule the job queues, so users should install PBS on your server first.

# 2 Installation

## 2.1 Hardware Requirement

- CPU: 64 bit Intel or AMD CPU
- RAM: 2 GB to 4 GB per CPU

## 2.2 Software Requirement

- Operation system: 64-bit Linux
- JDK: version 1.6
- perl: version 5.8.5 or above
- Python: version 2.3.4 or above
- R: version 2.1.0
- blastall：version 2.2.18
- Corona_Lite: Corona_Lite_Plus_4.2.1
- Perl models:
  - SVG model
  - GD model
  - GO model
- DEGseq package
- PBS

## 2.3 Install

Unzip the rna_solid_pipeline.tar.gz package into the desired location:

```
$ tar xzvf rna_solid_pipeline.tar.gz
```

# 3 RNA-seq Analysis Pipeline

The pipeline contains 6 steps:

1. Filter low quality reads
2. Mapping
3. Annotation
4. DEGseq analysis
5. GO analysis
6. KEGG analysis

## 3.1 Filter Low Quality Reads

In this step, reads with a lower quality than the given value will be filtered. This step is optional, so users can choose whether to do it.

## 3.2 Mapping

We use the Corona_Lite_Plus_4.2.1 program to map raw reads to the references selected by users.

## 3.3 Annotation

After mapping, we do annotation for the mapped reads.

## 3.4 DEGseq Analysis

DEGseq is a free R package for identifying differentially expressed genes (DEGs) from RNA-Seq data. Users could identify DEGs between two samples in their projects, after the corresponding annotation program has finished. To process this analysis, users should provide an annotation file with the same format as the output annotate file in the annotation step. This step is optional, so users could choose whether to do it.

## 3.5 GO Analysis

GO (Gene ontology) provides controlled vocabularies for the description of the molecular function, biological process and cellular component of gene products, which is one of the daily tools for downstream gene functional annotation. This step is optional, so users could choose whether to do it.

## 3.6 KEGG Analysis

KEGG PATHWAY is a collection of manually drawn pathway maps representing our knowledge on the molecular interaction and reaction networks and also on the structure relationships. This step is optional, so users could choose whether to do it.

# 4 Usage of the Package

The package consists of three parts:

- perl directory
- rna_solid_analysis.jar
- config.properties

## 4.1 Perl directory

We use perl scripts to process the analysis. The scripts are located in the perl directory.

## 4.2 rna_solid_analysis.jar

We use java programs to do three things:

- Encapsulate the invoking of the perl scripts so users can easily perform the process by running the java program.
- Do some counting and draw pictures after each step is finished.
- Generate html files which demonstrate the results after the pipeline is finished.

All the java programs are packaged in this executable jar file.

## 4.3 config.properties

This is the configuration file which has listed all the parameters needed by the program, so users should fulfill this file before running the program.

Each of the steps has a number of parameters. Table 1 to Table 7 shows the parameters.

Table 1: General parameters

| Parameter | Description | Example |
|---|---|---|
| RAW_DATA_DIRECTORY | The absolute path of directory of the raw data. | /testdata/rawdata/ |
| READS_FILE_NAME | The name of the raw reads file. | /testdata/rawdata/sample_ib.csfasta |
| QUALITY_FILE_NAME | The name of the quality file for the reads. | /testdata/rawdata/sample_ib_QV.qual |
| RESULT_DIRECTORY | The absolute path of directory which the result will be stored. | /result/ |
| WORK_QUEUE | PBS jobs running queue name. | ibque |
| TRACE_QUEUE | Tracking queue name. | ibque |

Table 2: Quality filter parameters

| Parameter | Description | Example |
|---|---|---|
| DO_QUALITY_FILTER | Whether or not filter low quality reads. | yes/no |
| FILTER_QUALITY | Reads with median QV below this value will be removed from the data set. | 8 |

Table 3: Mapping parameters

| Parameter | Description | Example |
|---|---|---|
| MATCH_CMAP_FILE | The cmap file identifies the files comprising the reference genome. It is a single tab delimited text file where each line of the file represents a single chromosome or juction. The Columns are:<br>1. Chromosome ID (must be a sequential integer).<br>2. Chromosome Name (Strings ok).<br>3. Path to the corresponding fasta file representing this chromosome. | /testdata/Cmap/match_cmap |
| TAG_LENGTH | The reads length. | 50,45 |
| MISMATCH_NUMBER | The max number of mismatching bases in math. | 5,5 |
| HITS_NUMBER | Maximum number of hits. Default 10. | 10 |

Table 4: Annotation parameters

| Parameter | Description | Example |
|---|---|---|
| CMAP_GUIDE_FILE | The guide file identifies the files comprising the reference genome and the representing junction files. It is a single tab delimited text file where each line of the file represents a single chromosome. The Columns are:<br>1. Chromosome ID index (must be a sequential integer, the same as the first column of the cmap file).<br>2. Chromosome Name (Strings ok, the chromosome ID the reference sequence represents.). | /testdata/Cmap/guide |

| | | |
|---|---|---|
| | 3. Chromosome Name (String ok, the same as the second column of the cmap file). | |
| GENELIST_CHROM_FILE | The gene list file, which contains the following columns:<br>1. GeneID<br>2. Chromosome<br>3. Gene strands<br>4. Gene start position (1 base)<br>5. Gene end position (1 base)<br>6. Exon number<br>7. Exon start (semicolon delimited)<br>8. Exon end (semicolon delimited) | /testdata/GeneListChrom |
| GENELIST_JUNCTION_FILE | The junction list file, which contains the following columns:<br>1. Gene ID<br>2. Chromosome<br>3. Gene strands<br>4. Junction sequence start<br>5. Junction sequence end | /testdata/GeneListJun |
| GENE_CLASS_BY_RNA_FILE | The gene class file, which contains the following columns:<br>1. Gene ID<br>2. Chromosome<br>3. RNA class | /testdata/GeneClassByRNA |
| GENE_FUNCTION_FILE | The gene function file, which contains the following columns:<br>1. Gene ID<br>2. Chromosome<br>3. Gene function | /testdata/GeneFunction |

Table 5: DEGseq parameters

| Parameter | Description | Example |
|---|---|---|
| IS_DO_DEGSEQ | Whether do DEGseq | yes/no |
| MARK_1 | Sample 1 name (the processed sample ) | sample1 |
| MARK_2 | Sample 2 name (the sample you want to be compared with the processed sample) | sample2 |
| COMPARED_ANNOTATE_FILE | Absolute path of annotate files for both samples, the file should have the same | |

| | | |
|---|---|---|
| | format as the output annotate file in the annotation step. which contains 14 columns:<br>1. ENSEMBL ID, ENSEMBL Gene ID;<br>2. Chromosome, Chromosome number<br>3. Strand, Gene Strand<br>4. GeneLen, Gene Length<br>5. ExonLen, Exon Length<br>6. GReads, Stat. of reads number that mapped to gene sequecnes in chromosome<br>7. JReads, Stat. of reads number that mapped to gene sequecnes in junction sequences<br>8. Exon_Intron, Stat. of reads number that mapped to exon_intron boundary in chromosome<br>9. ExonReads, Stat. of reads number that just mapped to exon sequences which include exon reads in chromosome and junction sequences<br>10. IntronReads, Stat. reads number that just mapped to intron sequences<br>11. RPKMValue, calculated by formula: ((1e+9)*ExonReads)/(ExonLen*Total UniqueMappedReads)<br>12. G_rev, Stat. of reads number that mapped to gene reverse strand in chromosome<br>13. J_rev, Stat. of reads number that mapped to gene reverse strand in junction sequences<br>14. Function, Functional information | |

Table 6：GO parameters

| Parameter | Description | Example |
|---|---|---|
| GO_LEVEL | Go function level | 2 |
| GO_OPTION | Which function will be discard. | proc \|func \|comp |
| GO_NUMBER_FILE | This file contains go number, one gene | /testdata/go/Gonum |

| | | |
|---|---|---|
| | one row, the first column is gene ID, the others are Go numbers delimited by tab. | ber |
| CELLULAR_COMPONENT_ONTOLOGY_FILE | Gene ontology file, http://www.geneontology.org/ontology/ | /testdata/go/component.ontology |
| MOLECULAR_FUNCTION_ONTOLOGY_FILE | Gene ontology file, http://www.geneontology.org/ontology/ | /testdata/go/function.ontology |
| BIOLOGICAL_PROCESS_ONTOLOGY_FILE | Gene ontology file, http://www.geneontology.org/ontology/ | /testdata/go/process.ontology |

Table 7: KEGG parameters

| Parameter | Description | Example |
|---|---|---|
| GENE_ENSEMBLE_LIST_FILE | Gene relevance file ftp://ftp.genome.jp/pub/kegg/genes/organisms/ | /testdata/kegg/genes/organisms/hsa/hsa_ensembl-hsa.list |
| GENE_KO_LIST_FILE | Gene relevance file ftp://ftp.genome.jp/pub/kegg/genes/organisms/ | /testdata/kegg/genes/organisms/hsa/hsa_ko.list |
| GENE_ENZYME_LIST_FILE | Gene relevance file ftp://ftp.genome.jp/pub/kegg/genes/organisms/ | /testdata/kegg/genes/organisms/hsa/hsa_enzyme.list |
| PATHWAY_FILE_DIRECORY | KO map directory ftp://ftp.genome.jp/pub/kegg/pathway/ko/ | /testdata/kegg/pathway /ko/ |

## 4.4 How To Run

1. Install the required softwares and configure the environment.

2. Switch to the directory where the package is installed.

3. Fill the configuration file (config.properties).

4. Run the command:

**java -jar rna_solid_analysis.jar config.properties**

## 4.5 Running Time

The running time depends on the environment of the service.

| Raw data size | Tag length | Reference size | Time |
|---|---|---|---|
| 14.4 GB | 35 bp | 5.6 GB | 25 hours |

## 4.6 Output Files

All the output files are located in the result directory users set.

### 4.6.1 HTML Files

/html/result_index.html

/html/mapping_result.html

/html/annotation_result.html

/html/degseq_result.html

### 4.6.2 Quality Distribution Picture

/filterpic/F3_1.png

/filterpic/F3_2.png

/_pic.html

### 4.6.3 Quality Filter

/filter/*_filter.csfasta: A csfasta file that has filtered the low quality reads of the raw csfasta data.

### 4.6.4 Mapping

/mapping/map_summary: Mapping result statistics

/mapping/map_detail: Detail statistics of mapping result

/mapping/map.png: PNG file demonstrates the result statistics

/mapping/matching_ .tag_length.miss_match_number/: The Corona Lite mapping result in each cycle mapping

### 4.6.5 Annotation

/annotation/annotate: The annotation result file

/annotation/statistic: The annotation result statistics file

/annotation/annopie.png: The pie graph of annotation

/annotation/annobar1.png: The bar graph of annotation genes classified by reads number

/annotation/annobar2.png: The bar graph of annotation genes classified by rpkm value

### 4.6.6 DEGseq Analysis

/degseq/output_score.txt

/degseq/output.html

/degseq/output/result.png

/degseq/output/Sample1_hist.png

/degseq/output/Sample1_Sample2_compare.png

/degseq/output/Sample2_hist.png

/degseq/output/SampleS_box.png

### 4.6.7 GO Analysis

/go/go.svg

### 4.6.8 KEGG Analysis

/kegg/pathwayPlot/

# 5 Test Data

| | |
|---|---|
| **Raw Data** | /testdata/rawdata/sample_ib.csfasta |
| | /testdata/rawdata/sample_ib_QV.qual |
| **Mapping** | /testdata/Cmap/match_cmap |
| **Annotation** | /testdata/Cmap/guide |
| | /testdata/GeneListChrom |
| | /testdata/GeneListJun |
| | /testdata/GeneClassByRna |
| | /testdata/GeneFunction |
| **DEGseq** | /testdata/annotate |
| **GO** | /testdata/GoNumber |
| | /testdata/go/component.ontology |
| | /testdata/go/function.ontology |
| | /testdata/go/process.ontology |
| **KEGG** | /testdata/kegg/genes/organisms/hsa/hsa_ensembl-hsa.list |
| | /testdata/kegg/genes/organisms/hsa/hsa_ko.list |
| | /testdata/kegg/genes/organisms/hsa/hsa_enzyme.list |
| | /testdata/kegg/pathway/ko/ |

# 6 Contact and Support

RNA-seq Analysis package for SOLiD is developed and maintained by Beijing Institute of Genomics(BIG), Chinese Academy of Sciences. If you have feedback or questions, please feel free to contact us at rnatap@big.ac.cn.