User Manual of PASS

Peng Wu, wupeng1@ihcams.ac.cn

Institute of hematology, CAMS, China

2019-4-28

1. Introduction

PASS (a Proteomics Alternative Splicing Screening pipeline) aims to identify alternative splicing (AS) events from mass spectrometry (MS)-based proteomics data. AS in transcriptomics level has been widely detected since RNA-seq technology was developed and applied, however, AS identification in proteomics scale has been few investigated. To adequately and accurately detecting AS from proteomics MS data, we have developed this PASS pipeline. Protein identification in proteomics still largely depends on the sequence database, and the proteogenomics method provides more available protein sequences for MS identification. Hence, PASS was designed to integrate the RNA-Seq data processing into MS database searching for more comprehensive AS identification. Of course, if RNA-Seq data is unavailable, PASS also supports the conventional MS searching against the known sequences in public databases (e.g. UniProt, ENSEMBL and NCBI).

PASS includes five modules. **1. processRNASEQ**. The reads from RNA-Seq are aligned to the reference genome and known genes and the resulted alignments are used to transcripts reconstruction, which could build the known and novel transcripts. **2. getORF**. We use all the reconstructed transcripts to extract the corresponding transcript sequences, which are further *in silico* converted into protein sequences using 6-frame translation. To reduce sequence redundancy, the longest ORF for each transcript is used in the following. **3. searchMS**. PASS prefers to apply the MSGF+ software[1] to carry out MS/MS database search. The high-confidence peptide-spectral-matches (PSMs) are filtered by the MSGFscore and FDR. **4. generateSAM**. The identified peptides are aligned to the reference genome, producing the alignment file with SAM format. **5. screenAS**. The minorly modified script sourced from MATS software[2] perform the AS detection and quantification, including the common seven spicing types.

PASS supports MS data from multiple proteases digestion (Trypsin, Lys-C, Lys-N, Glu-C, etc.), fragmentation types (collision-induced dissociation, higher energy collision dissociation, electron-transfer dissociation, etc.) and even distinct post-translational modification enrichments (phosphorylation, ubiquitination, etc.). PASS prefers to MS data with deeper coverage and high-precision spectra to carry out AS identification.

This user manual provides a step-by-step guide to AS identification based on MS data.

2. Installation

- Install PASS
  - unzip PASS-master.zip
  - chmod a+x *
  - export PATH=/PASS_install_path/:$PATH
- Install dependencies
  - Tophat2

        - wget tophat2-*.tar.gz
        - tar -zxvf tophat2-*.tar.gz
        - export PATH=/topaht2_install_path/:$PATH
    - Cufflinks
        - wget cufflinks-*.tar.gz
        - tar -zxvf cufflinks-*.tar.gz
        - export PATH=/cufflinks_install_path/:$PATH
    - MSGF+
        - wget MSGFPlus_v20190228.zip
        - unzip MSGFPlus_v20190228.zip

3. Transcript reconstruction

- **processRNASEQ**: Align RNA-Seq reads to the reference genome and reconstruct transcripts. *PASS* uses the classic RNA-Seq data analysis pipeline with *Tophat2* and *Cufflinks* tools[3]. We need to prepare the reference genome and its *bowtie2* index files. The known gene annotation file (.gtf) can help the alignment. Both paired-end and single-end sequencing reads are supported.
- **Usage**: `processRNASEQ [options] -g <genome> -f <.gtf> -r <.fastq>`

```
-g    Genome bowtie2 index name.
-f    Gene annotation file, .gtf format.
-r    File names for sequencing reads, .fastq format.
      - Compressed files (.fastq.gz) are also supported.
      - Paired-end files separated by commas.
-t    Path to tophat, eg. /home/user/bin/tophat
      - By default, we try to search tophat in system PATH.
-c    Path to cufflinks, eg. /home/user/bin/cufflinks
      - By default, we try to search cufflinks in system PATH.
-p    Number of used threads. [Default: 12]
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `processRNASEQ -g path_genomeandbowtie2index/genome.test -f exampleData/genes.test.gtf -r exampleData/Sample_R1.fastq.gz,exampleData/Sample_R2.fastq.gz`
- **Note**: In the above example, the genome and bowtie2 index could be generated as following:
    - cp mouse.mm10.chr12.fa genome.test.fa
    - bowtie2-build --threads 4 genome.test.fa genome.test
- **Output**:
    - transcripts.gtf

4. Protein sequences translation

- **getORF**: Firstly, *getORF* extracts transcript sequences by inputting the reconstructed transcript annotation .gtf file and the reference genome. Secondly, *getORF* applys the 6-frame translation method to produce all the potential amino acid sequences. Thirdly, the reconstructed transcripts by *Cufflinks* have no protein-coding sequence annotation information. So, *getORF* adds the CDS annotation to the transcripts by the translated ORFs. Lastly, *getORF* chooses the longest ORF as the protein product for each transcript for the following MS searching to reduce the sequence redundancy.
- **Usage**: `getORF [options] -f <.gtf> -g <genome.fa>`

```
-f    File name of gene annotation, .gtf format.
      - Recommend cufflinks to generate this file.
-g    Reference genome file name, fasta format.
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `getORF -f exampleData/transcripts.gtf -g exampleData/genome.test.fa`
- **Output**:
  - transcripts.longestorf.gtf
  - transcript.longestorf.fa
  - protein.longestorf.fa

5. MS/MS database searching

- **searchMS**: *PASS* performs peptide identification using the database search tool *MSGF+*, which is a sensitive and universal tool and works well on Linux/Unix system. In addition to *MSGF+*, users could apply other search tools, including *MaxQuant*, *MASCOT* and *SEQUEST*, as long as they can generate high-confidence peptide-spectrum matches. *MSGF+* supports multiple MS/MS file formats and requires that spectral should be centroided. Low-resolution MS is not encouraged to proteomics alternative splicing detection. The default modification parameters include fixed Carbamidomethyl C and variable Oxidation M and Acetylation Protein N-term. The input protein sequence file can be generated by *getORF* or downloaded the public sequence database.
- **Usage**: `searchMS [options] -s <MSGF_path> -m <example.mzML> -f <protein.fa>`

```
-s    Path to MSGFPlus.jar. eg. ~/software/MSGF.
-m    MS/MS file.
      - Support file formats including .mzML, .mzXML, .mgf, .ms2, .pkl and _dta.txt
      - Spectral should be centroided.
-f    Protein sequences
-p    Number of used threads. [Default: 12]
-t    Modification file name.
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `searchMS -s ~/software/MSGF -m exampleData/example.mzML -f exampleData/protein.longestorf.fa`
- **Output**:
  - PSM.tab

- File format:

```
No.|Column
-|-
1|spectrum
2|spectrumNativeID
3|assumed_charge
4|hit_rank
5|peptide
6|num_missed_cleavages
7|mvh
8|modification
9|NTT
```

6. Peptides alignment

- **generateSAM**: Due to the fact of that mose alternative splicing detection tools are based on alignment files from RNA-Seq, *PASS* would convert peptide-spectrum matches into an alignment .SAM file, which can clearly record the genomic location for each peptide. The input files include peptide-spectrum matches, transcript annotation, transcript sequences and protein sequences. The last three are generated by *getORF*. If users don't process RNA-Seq data and only identify known splicing events, users can download the corresponding files from the GENCODE database. Users can choose one representative spectral for one peptide used to generate alignment file if the study is only interested in whether the splicing occurs or not.
- **Usage**: `generateSAM [options] -m <PSM> -f <.gtf> -t <transcript.fa> -p <protein.fa>`

```
-m    Peptide spectral matches.
-f    File name of gene annotation, .gtf format.
-t    File name of transcript sequences, .fa format.
-p    File name of protein sequences, .fa format.
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `generateSAM -m exampleData/PSM.tab -f exampleData/transcripts.longestorf.gtf -t exampleData/transcript.longestorf.fa -p exampleData/protein.longestorf.fa`
- **Output**:
  - PSM.sam

7. Alternative splicing identification

- **screenAS**: *PASS* can detect seven common alternative spicing types, including skipped exons (SE), retained introns (RI), multually exclusive exons (MXE), alternative 5' or 3' splicing sites (A5SS, A3SS) and alternative first or last exons (AFE, ALE). The output file *summary.txt* concludes number of each splicing type. Only if both inclusion and exclusion isoforms are detected, the alternative splicing event will be successfully identified.
- **Usage**: `screenAS [options] -s <PSM.sam> -g <genes.gtf>`

```
-s    Sam format file generated by proteome identification.
-g    Gene annotation file, .gtf format.
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `screenAS -s exampleData/PSM.sam -g exampleData/transcripts.longestorf.gtf`
- **Output**:
  - summary.txt
  - PASS.SE.txt
  - PASS.RI.txt
  - PASS.MXE.txt
  - PASS.A5SS.txt
  - PASS.A3SS.txt
  - PASS.AFE.txt
  - PASS.ALE.txt

8. PASS - all-in-one command

- **PASS**: If users start the pipeline from the first step *processRNASEQ*, a command *PASS*, including the above 5 steps, could be performed in one line command.
- **Usage**: `PASS [options] -g <genome> -f <genes.gtf> -r <reads.fastq> -s <MSGFPlus.jar> -m <example.mzML>`

```
-g    Genome bowtie2 index name.
-f    Gene annotation file, .gtf format.
-r    File names for sequencing reads, .fastq format.
      - Compressed files (.fastq.gz) are also supported.
      - Paired-end files separated by commas.
-t    Path to tophat, eg. /home/user/bin/tophat
      - By default, we try to search tophat in system PATH.
-c    Path to cufflinks, eg. /home/user/bin/cufflinks
      - By default, we try to search cufflinks in system PATH.
-p    Number of used threads. [Default: 12]
-s    Path to MSGFPlus.jar. eg. ~/software/MSGF.
-m    MS/MS file.
      - Support file formats including .mzML, .mzXML, .mgf, .ms2, .pkl and _dta.txt
      - Spectra should be centroided.
-d    Modification file name.
-o    Output folder. [Default: ./PASS_out]
-h    Help message.
```

- **Example**: `PASS -g path_genomeandbowtie2index/genome.test -f exampleData/genes.test.gtf -r exampleData/Sample_R1.fastq.gz,exampleData/Sample_R2.fastq.gz -s ~/software/MSGF -m exampleData/example.mzML -p 4`
- **Output**:
  - summary.txt
  - PASS.SE.txt

- PASS.RI.txt
- PASS.MXE.txt
- PASS.A5SS.txt
- PASS.A3SS.txt
- PASS.AFE.txt
- PASS.ALE.txt

References

1. S. Kim, P. A. Pevzner, Nature communications 2014, 5, 5277.
2. S. Shen, J. W. Park, J. Huang, K. A. Dittmar, Z. X. Lu, Q. Zhou, R. P. Carstens, Y. Xing, Nucleic acids research 2012, 40, e61.
3. C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, L. Pachter, Nature biotechnology 2010, 28, 511.