GAPE User Manual

(build 20180130)

1. Program description

Version 1.1

GAPE is a one-stop proteogenomic informatics software that provides a multifaceted and standard workflow against eukaryotes in proteogenomic data-analysis cycle for genome refinement and global identification of PTM events. This software allows concurrent querying of proteomic and genomic databases to refining the genome and proteome annotations comprehensively. This includes MS data and database construction, database searches, FDR calculations, statistical result integration, validation of annotated genes, identification of previously unidentified genes, protein level identification of alternative spliced variants and SAAV, biological interpretation, and global PTM discovery. The software doesn't need any installation and enables plug-and-play use by packaging all functions installed and configured into a application framework. With a single command, GAPE automates all analysis steps and provides the user with the results of each step. Additionally, GAPE enables search results from other existing database search algorithms as additional input so that users utilize expediently the potential of proteogenomics in the discovery process.

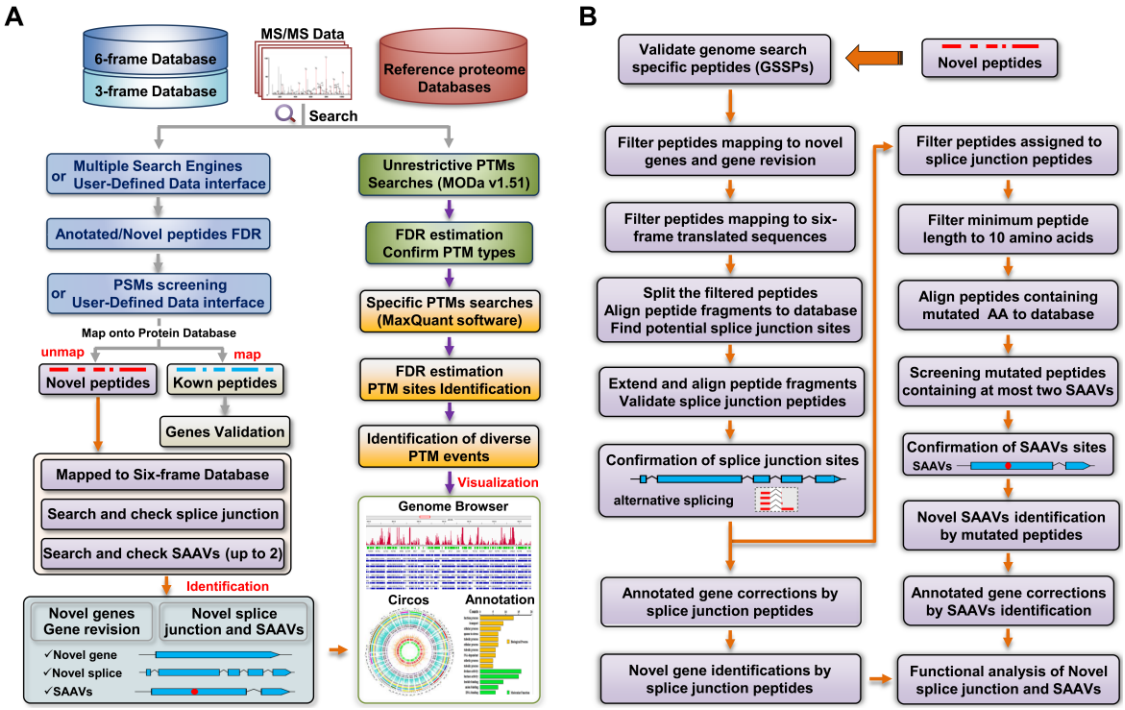

Fig. 1 The workflow of GAPE software. A. A schematic of the GAPE software workflow. B. Detailed pipeline in GAPE software used for identifications of alternative splicing and SAPs.

2. Software requirements

The current release of GAPE runs on Windows and linux 64-bit operating system.

Required softwares on Windows:

(1) JAVA 1.6 or higher

(2) Perl 5.10 or higher

(3) Parallel-Forkmanager module of Perl language (parallel processing fork manager).

Copy the "Parallel" folder under the unzipped GAPE directory to "site\lib" folder in the native Perl installation directory (e.g. C:\Perl64\site\lib).

(4) .net framework and MSFileReader

To perform MaxQuant database search the .net framework 4.5.x or higher and the MSFileReader 3.0 or higher have to be installed (http://www.coxdocs.org/doku.php?id=maxquant:common:download_and_installation)

(5) cygwin1.dll

Copy the "cygwin1.dll" file under the unzipped GAPE directory to the "C:\Windows\SysWOW64" directory.

Required softwares on linux:

(1) JAVA 1.6 or higher

(2) Perl 5.10 or higher

(3)Parallel-Forkmanager module of Perl language(parallel processing fork manager).

Execute the "yum install perl-Parallel-ForkManager" command to install Parallel-Forkmanager module.

(3) mono

The mono runtime environment version 3.2.1 or higher have to be installed. Some linux distributions already come with an installed mono. To check which version you have installed please run "mono -V" on a terminal. (http://www.mono-project.com/download/#download-lin-centos)

(5) libstdc++.so.5

Execute the "yum install libstdc++.so.5" command to install libstdc++.so.5 package.

3. Hardware requirements

The processor and memory requirements of GAPE depends on the complexity of actual analysis (e.g. geneme and MS/MS data size). A desktop PC with a quad core processor and eight GB of memory is sufficient for the entire proteogenomic analysis of a unicellular eukaryotes (e.g. Phaeodactylum tricornutum). If you need to analyze the multi-cellular eukaryote, the computer cluster is the best choice due to the larger geneme size (e.g. a cluster node that supports 40 threads and 250 GB of memory can run efficiently the complete zebrafish proteogenomic analysis using GAPE). While GAPE supports the multi-threading calculations in order to improve the calculation efficiency, it will consume more memory (e.g. SAAVs and splice junction peptides search steps).

4. Installation of GAPE

In order to install GAPE please download and extract necessary files:

(1) Download the GAPE_window64.zip (or GAPE_Linux64.tar.gz) archive from https://sourceforge.net/projects/gapeproteogenomic/

(2) Unpack the GAPE .zip file

GAPE is now ready for use!

5. Preparing Input Files

For proteogenomic analysis, mass spectrometry data must first be converted to the supported MS/MS input formats of mascot generic format (MGF). A popular option for converting from vendor file inputs and between various input formats is Proteowizard (proteowizard.sourceforge.net). The genome, RNA, EST and protein database must be supplied in FASTA format, and annotation file must be supplied in GFF3 format. To perform GO functional

annotation of genes in GAPE, you must have downloaded Uniref50 database (ftp://ftp.uniprot.org/pub/databases/uniprot/uniref/uniref50/uniref50.fasta.gz) and GO sequnces file (http://archive.geneontology.org/latest-lite/go_weekly-seqdb.fasta.gz).

In addition, the PSMs or GSSP result files need to be supplied as input when users use the –A or -G argument options. The PSMs result file format required by GAPE is a tab delimited file with at least four columns:

Column	Column header	Description
1	Protein ID	Protein number, Six-frame translation number or Three-frame translation number, ... Note: Decoy proteins ID prefixed with the "REV_".
2	Peptide	Peptide sequence
3	E-value	database level E-value (expected number of peptides in a random database having equal or better scores than the PSM score) - the lower the better
4	Spectrum ID	Number of peptide-spectrum matches (PSMs) for the given peptide
5	Sample	Name of sample or experiment
6	.....	
7	.....	

The GSSPs file format required by GAPE is a tab delimited file with two columns:

Column	Column header	Description
1	Protein ID	Protein number, Six-frame translation number or Three-frame translation number, ...
2	Peptide	Peptide sequence

6. Usage of GAPE

GAPE can be used from the command line with a text parameters file called Conf.txt. For running a one-stop proteogenomic analysis service including all analysis steps, the syntax of the command line call of GAPE is defined as:
java -jar GAPE_fat_V1.1.jar -c conf.txt

GAPE also provides two user-defined input interfaces to accepts the identified PSMs and GSSPs result files from the other existing database search engines. If you have obtained GSSPs result files, you can write the folder path containing GSSPs result files and customized database files path into the configuration file and run GAPE using the -c argument option. At this way, this software starts to run from the step that peptides map onto protein database(Fig. 1) and performs all subsequent analysis steps. The syntax of the command line call is defined as:
java -jar GAPE_fat_V1.1.jar -G custom_conf.txt

In another way, you can supply PSMs result files searched from other existing database search algorithms to GAPE as input and run this tool with the -A argument option. Similarly, the folder path containing PSMs result files and the related database files path need to be supplied to the configuration file. The syntax of the command line call is defined as:
java -jar GAPE_fat_V1.1.jar -A custom_conf.txt

7. Configuring GAPE

Extract the GAPE.jar into your working directory along with the sample configuration file. GAPE is configured using a text parameters file called conf.txt. You can edit the parameters file for a particular analysis, and the parameters file is passed as the first argument to GAPE. Parameter names are given left of the equal sign and parameter values are given to the right (e.g. threads_num=6). **File name does not contain any blank space or dot. Path separator is "\" on windows OS, but is "/" on Linux OS.**

General Parameters:

(1) User Name

Your user name

Default: test1

(2) Workdir

Path to the output directory

(3) Protein Database

Path to the protein database file(FASTA format).

(4) Genome Database

Path to the genome database file(FASTA format).

(5) Inputpath

Path to the directory containing MS/MS data files(MGF format).

(6) GFF_database

Path to the genome annotation file(GFF format).

(7) RNA_Database

Path to the directory containing the RNA database files(FASTA format). If user does not provide the database files, the parameter is set to null.

(8) EST_Database

Path to the directory containing the EST database files(FASTA format). If user does not provide the database files, the parameter is set to null.

(9) Mutiomics Database

Path to the database built by user-defined way (FASTA format). The database is used to search for identifying PSMs or GSSPs, and related to the PSMs or GSSPs result files. If user does not provide the database, the parameter is set to null.

(10) PSMs_file

Path to the directory containing the PSMs or GSSPs result files. If user does not provide the files, the parameter is set to null.

(11) Index_blast

Path to the index file of the inclusion relation between the ORF sequence from the muti-omics database and annotated protein sequence. If user does not provide the index file, the parameter is set to null.

(12) Index_Genom_RNA

Path to the index file of the inclusion relation between genomic DNA sequence and transcriptome RNA sequence. If user does not provide the index file, the parameter is set to null.

(13) threads_num

Number of CPU threads to use, should be set to the number of logical processors.

Default: 4

Search Parameters:

(14) precursor ion mass tolerance

Precursor mass tolerance

Default: 10

(15) precursor ion mass units

Precursor mass tolerance units(ppm or Daltons)

Default: ppm

(16) product ion mass tolerance

Fragment mass tolerance

Default: 0.05

(17) product ion mass units

Fragment mass tolerance units(ppm or Daltons)

Default: Daltons

(18) FragmentationMethodID

0: as written in the spectrum or CID if no info, 1: CID, 2: ETD, 3: HCD, 4: Merge spectra from the same precursor

Default: 1

(19) InstrumentID

0: Low-res LCQ/LTQ (Default for CID and ETD), 1: High-res LTQ (Default for HCD), 2: TOF, 3: Q-Exactive

Default: 1

(20) enzyme

Enzyme identifier(1=Trypsin, 2=LysC)

Default: 1

(21) allowed_missed_cleavage

The maximum number of considered missed cleavages. Valid values range from 0 to 3.

Default: 2

(22) modifications

Sets variable modifications (Lists of modifications are separated by commas, e.g. modifications=1,2,3). 1=M,oxidation on methionine;2=NQ,deamidation of N and Q;3=Ac_Nterm,acetylation of protein n-term

Default: 1

(23) peptide_SAAV_num

The maximum search number of SAAV on a peptide. Valid values range from 1 to 2.

(24) min_orf_length

The minimum ORF length.

Annotation Parameters:

(25) blast_db

Path to the Uniref50 database file.

(26) blast_GOdb_path

Path to the GO sequnces file.

MaxQuant search Parameters:

Each PTM search parameters are written to MaxQuant parameter section which separated by "#MSAmanda start#" and #Maxquant end#. GAPE implements PTMs automated search by integrating Maxquant search engine, so you can customize PTM search parameters by appending MaxQuant parameter section to search special PTM type.

Configuraiton Sample:

#Maxquant start#

num_threads=1

mainSearchTol=8

Enzyme=Trypsin

variable_mod_first=Oxidation (M);Deamidation (NQ)

variable_mod_second=Farnesyl (C)

fixedModifications=Carbamidomethyl (C)

maxMissedCleavages=2

minPepLen=6

#Maxquant end#

#Maxquant start#

……

……

#Maxquant end#

(27) open_Maxquant

Whether or not to search Maxquant(open=true, close=false).

Default: true

(28) Maxquant_inputpath

Path to the directory containing .raw data files(RAW format).

(29) num_threads

Number of concurrent threads to be executed together.

Default: 2

(30) mainSearchTol

The peptide mass mass tolerance in ppm during the main search.

Default: 6

(31)Enzyme

The search enzyme is specified by this parameter.

(e.g. Trypsin,Trypsin/P,Lys-C,GluC,GluC/Trypsin. More enzyme names refer to http://www.coxdocs.org/doku.php?id=maxquant:start)

Default: Trypsin

(32) variable_mod_first

Dynamic modifications (e.g. oxidation of methionine, deamidation of Asn/Gln or acetylation of protein N terminu). Multiple modifications are separated by semicolon.

(33) variable_mod_second

Dynamic modifications (e.g. acetylation of Lys, phosphorylation of Ser/Thr/Tyr or succinylation of Lys). The parameter can only be set to a single dynamic modification.

(34) fixedModifications

Specify a fixed modification.

Default: Carbamidomethyl (C)

(35) maxMissedCleavages

The number of missed cleavages that are maximally to tolerated in the in-silico digestion of the protein sequences.

Default: 2

(36) minPepLen

The minimal peptide length.

Default: 6

8. Output explanation

The analysis results by GAPE are stored in the "Result" folder of GAPE output directory.

Annotated proteins output files:

(1) annotated_PSMs.csv

CSV format file containing peptide-spectrum matches of annotated proteins.

(2) annotated_UniquePeptides.csv

CSV format file of annotated proteins with at least two unique peptides.

(3) annotated_SharedPeptides.csv

CSV format file of annotated proteins with at least two shared peptides.

Novel gene and revision of annotated genes output files:

(4) Novel_Genes.csv

CSV format file of novel Genes(previously unidentified protein-coding regions) identified with at least two unique GSSPs(genome search specific peptide).

(5) Corrections_Genes.csv

CSV format file of revision of annotated genes identified with at least two unique GSSPs(genome search specific peptide).

(6) Novel_Genes.gff, Corrections_Genes.gff

GFF format file of novel Genes and revision of annotated genes.

(7) singlePep_novel_Proteins.csv

CSV format file of novel Genes(previously unidentified protein-coding regions) identified with only a single unique GSSPs(genome search specific peptide).

(8) singlePep_corrections_Proteins.csv

CSV format file of revision of annotated genes identified with only a single unique GSSPs(genome search specific peptide).

(9) singlePep_novel_Proteins.gff, singlePep_corrections_Proteins.gff
GFF format file of novel Genes and revision of annotated genes with only a single unique GSSPs.

Alternative splicing protein coding genes output files:

(10) Novel_SpliceGenes.csv
CSV format file of novel alternative splicing protein coding genes identified with at least two unique GSSPs.

(11) Corrections_SpliceGenes.csv
CSV format file of revised alternative splicing protein coding genes identified with at least two unique GSSPs.

(12) SplicePeptides_GenomePosition.gff
GFF format file of identified splice junction peptides (note: a splice junction peptide per two lines).

(13) Splice_Peptides.csv
CSV format file of identified splice junction peptides.

(14) singlePep_Splice_novelProteins.csv
CSV format file of novel alternative splicing protein coding genes identified with only a single unique GSSPs.

(15) singlePep_Splice_correctionsProteins.csv
CSV format file of revised alternative splicing protein coding genes identified with only a single unique GSSPs.

Single amino acid variants output files:

(16) Novel_MutationsGenes.csv
CSV format file of novel single amino acid variants identified with at least two unique GSSPs.

(17) Corrections_MutationsGenes.csv
CSV format file of revised single amino acid variants identified with at least two unique GSSPs.

(18) AnnotatedGenes_SAAVs.csv
CSV format file of single amino acid variants from annotated proteins using genome search specific peptide.

(19) SAAVsPeptides_GenomePosition.gff
GFF format file of identified SAAVs peptides.

(20) SAAVs_Peptides.csv
CSV format file of identified SAAVs peptides.

(21) singlePep_SAAV_novelProteins.csv
CSV format file of novel single amino acid variants identified with only a single GSSPs.

(22) singlePep_SAAV_correctionsProteins.csv
CSV format file of revised single amino acid variants identified with only a single GSSPs.

Modifications output files identified by Moda:

(23) moda_modifiedProteins_GOannotations.csv
CSV format file of PTMs(post-translational modifications) result identified by Moda search algorithm.

Modifications output files identified by Maxquant:

(24) Identification results of Maxquant
The .txt text files prefixed by modification name of all PTMs results identified by MaxQuant search algorithm are stored in Maxquant_Result directory.