A short tutorial on using *PGA* for protein identification based on the database derived from RNA-Seq data

Bo Wen

October 27, 2020

Contents

1 Introduction

The data of mass spectrometry (MS)-based proteomics is generally achieved by peptide identification through comparison of the experimental mass spectra with the theoretical mass spectra that are derived from a reference protein database, however, this strategy could not identify new peptide and protein sequences

that are absent from a reference database. The customized protein databases on the basis of RNA-Seq data was proposed to assist and improve identification of such novel peptides. In addition, the strategy based on searching this database can improve the sensitivity of the peptide identification. The *PGA* package provides functions for construction of customized protein databases based on RNA-Seq data, database searching, post-processing and report generation. This kind of customized protein database includes both the reference database (such as Refseq or ENSEMBL) and the novel peptide sequences form RNA-Seq data. In general, customized protein database includes the following four kind of new peptides (or proteins):1) Single nucleotide variation (SNV) caused peptides; 2) Short insertion and deletion (INDEL) caused peptides; 3) Alternative splicing caused peptides; 4) Novel transcripts codeing peptides. In addition, *PGA* can also be used to create proteomic database based on the transcript sequences from the de novo assembly of RNA-Seq data. This strategy of proteomic database construction is very useful in proteomic study for non-model organism. This document describes how to use the functions included in the R package *PGA*.

2 Construction of customized protein databases based on RNA-Seq data

2.1 Based on the result from analysis of RNA-Seq data with a reference genome

2.1.1 Preparing annotation files

In order to translate the RNA-Seq information to peptide sequences, the users need to download numerous pieces of genome annotation information. There are two functions in *PGA* to prepare these information: `PrepareAnnotationRefseq2` and `PrepareAnnotationEnsembl2`. The methods are similar with functions `PrepareAnnotationRefseq` and `PrepareAnnotationEnsembl` in *customProDB* [1] with several changes. However, the usage of these functions are the same with those in *customProDB*.

2.1.2 Building database from RNA-Seq data

Building a comprehensive customized protein databases based on RNA-Seq data by using *PGA*, the users usually need to provide three files:

1. a VCF format file which contains SNV or INDEL information;
2. a BED format file which contains splice junctions information;
3. a GTF format file which contains novel transcripts information.

The above files provide almost all of the events which generate potential novel peptides from RNA-Seq data.

```
vcffile <- system.file("extdata/input", "PGA.vcf",package="PGA")
bedfile <- system.file("extdata/input", "junctions.bed",package="PGA")
gtffile <- system.file("extdata/input", "transcripts.gtf",package="PGA")
annotation <- system.file("extdata", "annotation",package="PGA")
outfile_path<-"db/"
outfile_name<-"test"
library(BSgenome.Hsapiens.UCSC.hg19)
dbfile <- dbCreator(gtfFile=gtffile,vcfFile=vcffile,bedFile=bedfile,
                    annotation_path=annotation,outfile_name=outfile_name,
                    genome=Hsapiens,outdir=outfile_path)
```

For each kind of event mentioned above, two files are generated. One is a FASTA format file and the other is a file with a .tab suffix. The latter contains the detailed information about novel peptides . Except these files, a combined FASTA format file is generated. This is the final customized protein database which will be used for database searching. If the parameter **"make_decoy"** in `dbCreator` function is set **"TRUE"** (This is the default value for parameter **"make_decoy"**), this file will contain the decoy sequences.

2.2 Based on the result from de novo assembly of RNA-Seq data without a reference genome

The proteomic studies typically depend on the availability of sequenced genomes. For proteomic study of nonmodel organism, it's usually limited by the absence of protein sequence data. Currently in this case, protein sequence databases can be generated from the analysis of RNA-Seq data by a genome-independent (de novo) strategy. At present, several software can be used to perform de novo assembly of RNA-seq data, such as Trinity [2], Oases [3] and SOAPdenovo-Trans [4]. There is a Nature protocol about how to use **Trinity** to perform de novo assembly of RNA-seq data [5]. In general, the de novo assembly sofware output a FASTA format file (click this link "Output of Trinity Assembly" to see the output of **Trinity**) which contains the transcript sequences.

In *PGA*, the function `createProDB4DenovoRNASeq` can be used to create a proteomic database based on the FASTA format file which is derived from the de novo assembly sofware, such as **Trinity**. It does not need any annotation files refered in above section.

```
transcript_seq_file <- system.file("extdata/input", "Trinity.fasta",
                                   package="PGA")
outdb <- createProDB4DenovoRNASeq(infa=transcript_seq_file,
                                  outfile_name = "denovo")

## .......  Proteomic database construction !  ............

## Write protein sequences to file:  denovo_txFinder.fasta

cat(outdb,"\n")

## denovo_txFinder.fasta
```

3 MS/MS data searching

After the customized protein database constructed, *rTANDEM* package [6] is adopted to search the database against tandem mass spectra to detect peptides. *rTANDEM* package interfaces with the popular used open source search engine *X!Tandem* [7] algorithm in R.

```
msfile <- system.file("extdata/input", "pga.mgf",package="PGA")
idfile <- runTandem(spectra = msfile, fasta = dbfile, outdir = "./", cpu = 6,
                    enzyme = "[KR]|[X]", varmod = "15.994915@M",itol = 0.05,
                    fixmod = "57.021464@C", tol = 10, tolu = "ppm",
                    itolu = "Daltons", miss = 2, maxCharge = 8, ti = FALSE)

## 2020-10-27 23:25:22
## Loading spectra
##  (mgf). loaded.
## Spectra matching criteria = 169
## Starting threads ...... started.
## Computing models:
##  t
##  sequences modelled = 0 ks
## Model refinement:
## Merging results:
##  from 23456
##
```

```
## Creating report:
##  initial calculations  ..... done.
##  sorting  ..... done.
##  finding repeats ..... done.
##  evaluating results ..... done.
##  calculating expectations ..... done.
##  writing results ..... done.
##
## Valid models = 169
## Unique models = 151
## Estimated false positives = 0 +/- 1
```

The results are written in xml format to the directory specified and will be loaded for further processing.

Alternatively, Alternatively, search result with dat format from MASCOT [8] or mzIdentML [9] format from MS-GF+ [10], MyriMatch [11], OMSSA [12] (converting OMSSA result to mzIdentML by mzidLibrary [13]) and IPeak [14], was also accepted by *PGA*.

4 Post-processing

After the MS/MS data searching, the function `parserGear` can be used to parse the search result. It calculates the q-value for each peptide spectrum matches (PSMs) and then utilizes the Occam's razor approach [15] to deal with degenerated wild peptides by finding a minimum subset of proteins that covered all of the identified wild peptides.

```
parserGear(file = idfile, db = dbfile, decoyPrefix="#REV#",xmx=1,thread=8,
           outdir = "parser_outdir")
```

It exports some tab-delimited files containing the peptide identification result and protein identification result. The annotated spectra for the identified novel peptides which pass the threshold are exported.

This function also accepts the "raw" MASCOT [8] result (dat format) or mzIdentML [9] format file from MS-GF+ [10], MyriMatch [11], OMSSA [12] (converting OMSSA result to mzIdentML by mzidLibrary [13]) and IPeak [14]. For instance,

```
dat_file <- system.file("extdata/input", "mascot.dat",package="PGA")
parserGear(file = dat_file, db = dbfile, decoyPrefix="#REV#",xmx=1,thread=8,
           outdir = "parser_outdir_mascot")
```

Unfortunately,we don't offer the wrapper function for Mascot search under current conditions. So you have to launch the independent identification by Mascot. For how to export dat format file from MASCOT search, please click this link "Export search results" to see the instruction.

If a user wants to take mzIdentML file as input, a java parser from the github must be downloaded and replaces the java parser in PGA package with the same name. Then use the function `parserGear` as below:

```
## The following code works only after the java parser has been updated.
vcffile <- system.file("extdata/input", "PGA.vcf",package="PGA")
bedfile <- system.file("extdata/input", "junctions.bed",package="PGA")
gtffile <- system.file("extdata/input", "transcripts.gtf",package="PGA")
annotation <- system.file("extdata", "annotation",package="PGA")
outfile_path<-"db/"
outfile_name<-"test"
library(BSgenome.Hsapiens.UCSC.hg19)
dbfile <- dbCreator(gtfFile=gtffile,vcfFile=vcffile,bedFile=bedfile,
                    annotation_path=annotation,outfile_name=outfile_name,
                    genome=Hsapiens,outdir=outfile_path)

msfile <- system.file("extdata/input", "pga.mgf",package="PGA")

## MS-GF+ (mzIdentML) as the peptide identification software
mzid <- system.file("extdata/input", "msgfplus.mzid",package="PGA")
parserGear(file = mzid, db = dbfile, msfile = msfile,
           decoyPrefix="#REV#",xmx=1,thread=8,
           outdir = "parser_outdir")
```

5 HTML-based report generation

The results are then summarised and compiled into an interactive HTML report.

```
reportGear(parser_dir = "parser_outdir", tab_dir = outfile_path,
           report_dir = "report")

## create the main page...
```

After the analysis has completed, the file 'index.html' in the output directory can be opened in a web browser to access report generated. In general, this report will show the identification result for four kinds of novel peptides, such as SNV-caused peptides, INDEL-caused peptides, alternative splicing caused peptides and novel transcripts codeing peptides.

If the RefSeq annotation is used in the RNA-Seq data analysis, the report will show the gene name for each protein. However, if the Ensembl annotation is used in the RNA-Seq data analysis, a user can use the function `addGeneName4Ensembl` to add the gene name information into the report as below:

```
## Don't run. It only works if you have generated a
## report with using Ensembl annotation.
mart <- biomaRt::useMart("ENSEMBL_MART_ENSEMBL",
        dataset="hsapiens_gene_ensembl",
        host="grch37.ensembl.org",
        path="/biomart/martservice",
        archive=FALSE)

addGeneName4Ensembl(mart=mart,report="report")
```

6 Integrated function `easyRun`

The function `easyRun` automates the data analysis process. It will process the dataset in the following way:

1. Customized protein database construction
2. MS/MS searching
3. Post-processing
4. HTML-based report generation

This function can be called as following:

```
vcffile <- system.file("extdata/input", "PGA.vcf",package="PGA")
bedfile <- system.file("extdata/input", "junctions.bed",package="PGA")
gtffile <- system.file("extdata/input", "transcripts.gtf",package="PGA")
annotation <- system.file("extdata", "annotation",package="PGA")
library(BSgenome.Hsapiens.UCSC.hg19)
msfile <- system.file("extdata/input", "pga.mgf",package="PGA")
easyRun(gtfFile=gtffile,vcfFile=vcffile,bedFile=bedfile,spectra=msfile,
        annotation_path=annotation,genome=Hsapiens,cpu = 6,
```

```
        enzyme = "[KR]|[X]", varmod = "15.994915@M",itol = 0.05,
        fixmod = "57.021464@C", tol = 10, tolu = "ppm", itolu = "Daltons",
        miss = 2, maxCharge = 8, ti = FALSE,xmx=1)
```

```
## Stage 1. Customized protein database construction.
## Stage 2. MS/MS searching.
## 2020-10-27 23:29:27
## Loading spectra
##  (mgf). loaded.
## Spectra matching criteria = 169
## Starting threads ...... started.
## Computing models:
##  t
##  sequences modelled = 0 ks
## Model refinement:
## Merging results:
##  from 23456
##
## Creating report:
##  initial calculations  ..... done.
##  sorting  ..... done.
##  finding repeats ..... done.
##  evaluating results ..... done.
##  calculating expectations ..... done.
##  writing results ..... done.
##
## Valid models = 169
## Unique models = 151
## Estimated false positives = 0 +/- 1
##
##
## Stage 3. Post-processing.
## Stage 4. HTML-based report generation.
## create the main page...
```

After the analysis has completed, the file '`index.html`' in the output directory can be opened in a web browser to access report generated.

7 FAQ

7.1 How to export dat format file for MASCOT search?

MASCOT is a widely used commercial software for protein identification based on MS/MS data. For how to export dat format file from MASCOT search, please click this link "Export search results" to see the instruction.

7.2 How to convert OMSSA result file to mzIdentML file?

OMSSA is a free software for protein identification based on MS/MS data. The user can use the SearchGUI [16] to do the OMSSA search. The result file (.omx) of OMSSA can be converted to mzIdentML by mzidLibrary [13]. Please see the user's manual of mzidLibrary for how to do this.

7.3 How to do the MS/MS searching in parallel (multi-threading)?

In *PGA*, the MS/MS searching can be performed in parallel (multi-threading) if the user use the default search software **X!Tandem**. The parameter **"cpu"** in function `runTandem` is used to control the number of CPU used for MS/MS searching.

7.4 How does PGA work in respective to RNA-Seq data generated by different NGS technologies?

As *PGA* takes .bed, .vcf, .gtf and FASTA format files as input for construction of customized proteomic database, it should be compatible with the RNA-Seq data produced by different NGS technologies (eg Illumina, IonTorrent, Pacific Bioscience, ...) as these file formats are standard formats used in NGS data analysis.

7.5 What system requirements are recommended for PGA?

Intel Core processor (8 CPU), 48 GB RAM, 500 GB disk and 64-bit Windows 7.

Session information

All software and respective versions used to produce this document are listed below.

- R version 4.0.3 (2020-10-10), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Running under: `Windows Server 2012 R2 x64 (build 9600)`
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.52.0, BSgenome 1.58.0, BSgenome.Hsapiens.UCSC.hg19 1.4.3, Biobase 2.50.0, BiocGenerics 0.36.0, Biostrings 2.58.0, GenomeInfoDb 1.26.0, GenomicFeatures 1.42.0, GenomicRanges 1.42.0, IRanges 2.24.0, PGA 1.20.0, Rcpp 1.0.5, S4Vectors 0.28.0, XML 3.99-0.5, XVector 0.30.0, data.table 1.13.2, rTANDEM 1.30.0, rtracklayer 1.50.0
- Loaded via a namespace (and not attached): AhoCorasickTrie 0.1.2, BiocFileCache 1.14.0, BiocManager 1.30.10, BiocParallel 1.24.0, BiocStyle 2.18.0, DBI 1.1.0, DelayedArray 0.16.0, GenomeInfoDbData 1.2.4, GenomicAlignments 1.26.0, MASS 7.3-53, Matrix 1.2-18, MatrixGenerics 1.2.0, Nozzle.R1 1.1-1, R6 2.4.1, RColorBrewer 1.1-2, RCurl 1.98-1.2, RSQLite 2.2.1, Rsamtools 2.6.0, SummarizedExperiment 1.20.0, VariantAnnotation 1.36.0, ade4 1.7-15, askpass 1.1, assertthat 0.2.1, biomaRt 2.46.0, bit 4.0.4, bit64 4.0.5, bitops 1.0-6, blob 1.2.1, codetools 0.2-16, colorspace 1.4-1, compiler 4.0.3, crayon 1.3.4, curl 4.3, customProDB 1.30.0, dbplyr 1.4.4, digest 0.6.27, dplyr 1.0.2, ellipsis 0.3.1, evaluate 0.14, farver 2.0.3, generics 0.0.2, ggplot2 3.3.2, glue 1.4.2, grid 4.0.3,

gtable 0.3.0, highr 0.8, hms 0.5.3, htmltools 0.5.0, httr 1.4.2, knitr 1.30, labeling 0.4.2, lattice 0.20-41, lifecycle 0.2.0, magrittr 1.5, matrixStats 0.57.0, memoise 1.1.0, munsell 0.5.0, openssl 1.4.3, pheatmap 1.0.12, pillar 1.4.6, pkgconfig 2.0.3, plyr 1.8.6, prettyunits 1.1.1, processx 3.4.4, progress 1.2.2, ps 1.4.0, purrr 0.3.4, rappdirs 0.3.1, readr 1.4.0, rlang 0.4.8, rmarkdown 2.5, scales 1.1.1, seqinr 4.2-4, stringi 1.5.3, stringr 1.4.0, tibble 3.0.4, tidyselect 1.1.0, tools 4.0.3, vctrs 0.3.4, xfun 0.18, xml2 1.3.2, yaml 2.2.1, zlibbioc 1.36.0

References

[1] Xiaojing Wang and Bing Zhang. customprodb: an r package to generate customized protein databases from rna-seq data for proteomics search. *Bioinformatics*, 29(24):3235–3237, 2013.

[2] Manfred G Grabherr, Brian J Haas, Moran Yassour, Joshua Z Levin, Dawn A Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, et al. Full-length transcriptome assembly from rna-seq data without a reference genome. *Nature biotechnology*, 29(7):644–652, 2011.

[3] Marcel H Schulz, Daniel R Zerbino, Martin Vingron, and Ewan Birney. Oases: robust de novo rna-seq assembly across the dynamic range of expression levels. *Bioinformatics*, 28(8):1086–1092, 2012.

[4] Yinlong Xie, Gengxiong Wu, Jingbo Tang, Ruibang Luo, Jordan Patterson, Shanlin Liu, Weihua Huang, Guangzhu He, Shengchang Gu, Shengkang Li, et al. Soapdenovo-trans: de novo transcriptome assembly with short rna-seq reads. *Bioinformatics*, 30(12):1660–1666, 2014.

[5] Brian J Haas, Alexie Papanicolaou, Moran Yassour, Manfred Grabherr, Philip D Blood, Joshua Bowden, Matthew Brian Couger, David Eccles, Bo Li, Matthias Lieber, et al. De novo transcript sequence reconstruction from rna-seq using the trinity platform for reference generation and analysis. *Nature protocols*, 8(8):1494–1512, 2013.

[6] Frederic Fournier, Charles Joly Beauparlant, Rene Paradis, and Arnaud Droit. *rTANDEM: Encapsulates X!Tandem in R.*, 2013. R package version 1.2.0.

[7] R Craig and R C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–7, Jun 2004. doi:10.1093/bioinformatics/bth092.

[8] John S Cottrell and U London. Probability-based protein identification by searching sequence databases using mass spectrometry data. *electrophoresis*, 20(18):3551–3567, 1999.

[9] Andrew R Jones, Martin Eisenacher, Gerhard Mayer, Oliver Kohlbacher, Jennifer Siepen, Simon J Hubbard, Julian N Selley, Brian C Searle, James Shofstahl, Sean L Seymour, et al. The mzidentml data standard for mass spectrometry-based proteomics results. *Molecular & Cellular Proteomics*, 11(7):M111–014381, 2012.

[10] Sangtae Kim and Pavel A Pevzner. Ms-gf+ makes progress towards a universal database search tool for proteomics. *Nature communications*, 5, 2014.

[11] David L Tabb, Christopher G Fernando, and Matthew C Chambers. Myrimatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis. *Journal of proteome research*, 6(2):654–661, 2007.

[12] Lewis Y Geer, Sanford P Markey, Jeffrey A Kowalak, Lukas Wagner, Ming Xu, Dawn M Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H Bryant. Open mass spectrometry search algorithm. *Journal of proteome research*, 3(5):958–964, 2004.

[13] Fawaz Ghali, Ritesh Krishna, Pieter Lukasse, Salvador Martínez-Bartolomé, Florian Reisinger, Henning Hermjakob, Juan Antonio Vizcaíno, and Andrew R Jones. Tools (viewer, library and validator) that facilitate use of the peptide and protein identification standard format, termed mzidentml. *Molecular & Cellular Proteomics*, 12(11):3026–3035, 2013.

[14] Bo Wen, Chaoqin Du, Guilin Li, Fawaz Ghali, Andrew R Jones, Lukas Käll, Shaohang Xu, Ruo Zhou, Zhe Ren, Qiang Feng, et al. Ipeak: An open source tool to combine results from multiple ms/ms search engines. *Proteomics*, 2015.

[15] A. I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Anal Chem*, 75(17):4646–58, 2003.

[16] Marc Vaudel, Harald Barsnes, Frode S Berven, Albert Sickmann, and Lennart Martens. Searchgui: An open-source graphical user interface for simultaneous omssa and x! tandem searches. *Proteomics*, 11(5):996–999, 2011.