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

Bo Wen

October 14, 2015

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. 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 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*.

2.2 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.

3 MS/MS data searching

After the customized protein database constructed, *rTANDEM* package [2] 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* [3] 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)
```

```
## 2015-10-14 00:56:21
## 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.

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 [4] 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 result file as input(dat format). For instance,

```
dat_file<-"mascot_raw.dat"
parserGear(file = dat_file, db = dbfile, decoyPrefix="#REV#",xmx=1,thread=8,
           outdir = "parser_outdir")
```

Unfortunately,we don't offer the wrapper function for Mascot search under current conditions. So you have to launch the independent identification by Mascot.

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 kind of novel peptides, such as SNV-caused peptides, INDEL-caused peptides, alternative splicing caused peptides and novel transcripts codeing peptides.

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.
## 2015-10-14 00:56:46
```

```
## 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.

Session information

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

- R version 3.2.2 (2015-08-14), `x86_64-apple-darwin13.4.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.32.0, BSgenome 1.38.0, BSgenome.Hsapiens.UCSC.hg19 1.4.0, Biobase 2.30.0, BiocGenerics 0.16.0, Biostrings 2.38.0, GenomeInfoDb 1.6.0, GenomicFeatures 1.22.0, GenomicRanges 1.22.0, IRanges 2.4.0, PGA 1.0.0, Rcpp 0.12.1, S4Vectors 0.8.0, XML 3.98-1.3, XVector 0.10.0, data.table 1.9.6, rTANDEM 1.10.0, rtracklayer 1.30.0

- Loaded via a namespace (and not attached): BiocParallel 1.4.0, BiocStyle 1.8.0, DBI 0.3.1, GenomicAlignments 1.6.0, MASS 7.3-44, Nozzle.R1 1.1-1, RColorBrewer 1.1-2, RCurl 1.95-4.7, RSQLite 1.0.0, Rsamtools 1.22.0, SummarizedExperiment 1.0.0, VariantAnnotation 1.16.0, biomaRt 2.26.0, bitops 1.0-6, chron 2.3-47, colorspace 1.2-6, customProDB 1.10.0, digest 0.6.8, evaluate 0.8, formatR 1.2.1, futile.logger 1.4.1, futile.options 1.0.0, ggplot2 1.0.1, grid 3.2.2, gtable 0.1.2, highr 0.5.1, knitr 1.11, labeling 0.3, lambda.r 1.1.7, magrittr 1.5, munsell 0.4.2, pheatmap 1.0.7, plyr 1.8.3, proto 0.3-10, reshape2 1.4.1, scales 0.3.0, stringi 0.5-5, stringr 1.0.0, tools 3.2.2, zlibbioc 1.16.0

References

[1] Xiaojing Wang and Bing Zhang. customprodb: an r package to generate customized protein databases from rna-seq data. *Bioinformatics*, 29:3235–3237, 2013. URL: http://bioinformatics.oxfordjournals.org/content/29/24/3235, doi:10.1093/bioinformatics/btt543.

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

[3] 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.

[4] 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.