APIR Tutorial

Yiling Chen

4/29/2021

Contents

---

1 Introduction

In this tutorial, we will demonstrate how users could use APIR to combine the PSM-level output from multiple database search algorithms.

You could find out more about APIR from our APIR paper: https://www.biorxiv.org/content/10.1101/2021.09.08.459494v2.external-links.html

Note: For technical problems, please report to Github Issues (https://github.com/yiling0210/APIR/issues). For suggestions and comments on the method, please contact Yiling Elaine Chen (yiling0210@ucla.edu) or Dr. Jingyi Jessica Li (jli@stat.ucla.edu).

---

2 Installation

APIR package is currently deposited on Github. Users could run the following command in R to install APIR:

```
if (!require(devtools)) install.packages("devtools")
library(devtools)
install_github("yiling0210/APIR")
```

---

3 Implementation

3.1 The data preparation

We will use the search results from Byonic (ThermoScientific) and MaxQuant (Andromeda) on a triple negative breast cancer dataset from Fang et al. (2016). In this section, we show how to process the PSM-level search results into the format APIR requires as input.

First, download processed `.xlsx` files from this link and save them to the working directory. Here we use `.xlsx` files for illustration. Users could use other types of format as long as these files could be loaded and processed into an R data.frame object.

Next, load them into R using the following command

```
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.0.2
```

```
byonic_target = read.xlsx(xlsxFile = "Genistein Byonic 100_FDR Target PSMs.xlsx")
byonic_decoy = read.xlsx(xlsxFile = "Genistein Byonic 100_FDR DECOY PSMs.xlsx")
maxquant_target = read.xlsx(xlsxFile = "Genistein MaxQuant 100_FDR Target PSMs.xlsx")
maxquant_decoy = read.xlsx(xlsxFile = "Genistein MaxQuant 100_FDR DECOY PSMs.xlsx")
head(byonic_target[, 1:5])
```

```
##   PSM.Ambiguity                              Annotated.Sequence
## 1   Unambiguous [K].yGPADVEDTTGSGATDSkDDDDIDLFGsDDEEESEEAkR.[L]
## 2   Unambiguous     [K].tDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHk.[T]
## 3   Unambiguous     [K].tDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHk.[T]
## 4   Unambiguous     [K].tDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHk.[T]
## 5   Unambiguous     [K].tDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHk.[T]
## 6   Unambiguous     [K].tDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHk.[T]
##                                   Sequence
## 1 YGPADVEDTTGSGATDSKDDDDIDLFGSDDEEESEEAKR
## 2     TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK
## 3     TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK
## 4     TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK
## 5     TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK
## 6     TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK
##                                                   Modifications #.Proteins
## 1 N-Term(TMT6plex); K18(TMT6plex); S28(Phospho); K38(TMT6plex)          1
## 2                               N-Term(TMT6plex); K35(TMT6plex)          1
## 3                               N-Term(TMT6plex); K35(TMT6plex)          1
## 4                               N-Term(TMT6plex); K35(TMT6plex)          1
## 5                               N-Term(TMT6plex); K35(TMT6plex)          1
## 6                               N-Term(TMT6plex); K35(TMT6plex)          1
```

`byonic_target`, `byonic_decoy`, `maxquant_target`, and `maxquant_decoy` contain the target or decoy PSM-level output from Byonic and Maxquant respectively.

APIR requires the following formatting of these files from users' end.

3.1.1 Mass spectrum files

Users need to make sure the mass spectrum files between search algorithms and between target and decoy searches are consistently formatted. A sanity check would be

```
ifcompleteoverlap = function(x, y) {
    x = unique(x)
    y = unique(y)
    if (all(x %in% y) & all(y %in% x)) {
        return(T)
    } else {
        return(F)
    }
}
ifcompleteoverlap(byonic_target$Spectrum.File,
    byonic_decoy$Spectrum.File)
```

```
## [1] TRUE
```

```
ifcompleteoverlap(byonic_target$Spectrum.File,
    maxquant_target$Spectrum.File)
```

```
## [1] FALSE
```

```
### correct file formatting in MaxQuant
### target and decoy
files = maxquant_target$Raw.file
files = paste0(files, ".raw")
all(files %in% byonic_target$Spectrum.File)
```

```
## [1] TRUE
```

```
maxquant_target$Raw.file = files

files = maxquant_decoy$Raw.file
files = paste0(files, ".raw")
all(files %in% byonic_target$Spectrum.File)
```

```
## [1] TRUE
```

```
maxquant_decoy$Raw.file = files
```

Moreover, within the same search algorithms, the column name of mass spectrum files in the decoy output should match that in the target output. In this case, all four files use `Spectrum.File` as the column name.

3.1.2 Scan numbers

Users need to make sure the column names of the scan numbers are consistent between the target and the decoy output. In this case, both `byonic_target` and `byonic_decoy` use `First.Scan` as the column name, and `maxquant_target` and `maxquant_decoy` use `MS/MS.scan.number` as the column name.

3.1.3 Peptide sequences

Users need to make sure that the peptide sequences are formated in the same fashion as in Byonic. That is, the sequence is coded by capital letters representing amino acid residues; modification information should be included in a separate column. To give concrete examples,

```
head(byonic_target$Sequence)
```

```
## [1] "YGPADVEDTTGSGATDSKDDDDIDLFGSDDEEESEEAKR"
## [2] "TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK"
## [3] "TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK"
## [4] "TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK"
## [5] "TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK"
## [6] "TDNAGDQHGGGGGGGGGAGAAGGGGGGENYDDPHK"
```

Moreover, within the same search algorithms,the column names of the peptide sequences are consistent between the target and the decoy output. In this case, all four files use `Sequence` as the column name.

3.1.4 Scores

Users need to pick a column as the score of PSMs. We recommend use q-values or PEPs. Within the same search algorithms, the column name of the peptide sequences should be consistent between the target and the decoy output. Note that if users want to choose other scores, transformations may be necessary so that a smaller score value indicates a higher probability of a PSM being correct. In this case, we use q-values from Byonic and PEP from MaxQuant:

```
head(byonic_target$`q-Value.2D`)
```

```
## [1] 1.070466e-23 1.342059e-23 6.273472e-23 4.097557e-22 6.179711e-22
## [6] 7.839196e-22
```

```
head(maxquant_target$PEP)
```

```
## [1] 2.4505e-148 2.5284e-132 6.4334e-128 3.6601e-118  9.3753e-97  1.8780e-94
```

3.1.5 Master/leading proteins

Users need to specify the column names of master proteins in the target search. The master proteins should be formated as below:

```
head(byonic_target$Master.Protein.Accessions)
```

```
## [1] "P24534" "P14866" "P14866" "P14866" "P14866" "P14866"
```

```
maxquant_target$`Proteins.(format.fixed)`[grep(";",
    maxquant_target$`Proteins.(format.fixed)`)[1:5]]
```

```
## [1] "P31150; P50395"
## [2] "P42167; P42166"
## [3] "P07910; O60812; B2RXH8; P0DMR1; B7ZW38"
## [4] "P0DMV8; P0DMV9; P34931; P17066; P48741"
## [5] "P08238; Q58FF8"
```

where "P24534" is a Uniprot protein accession number, and should not contain species information. If the master proteins are not unique, they should be separated by ";" with a single space following the semicolon, for example, "P08238; Q58FF8".

3.1.6 Protein positions

Users need to specify the columns that contain the positions of a peptide in a protein. This column should be formatted in the following format:

```
head(byonic_target$Positions.in.Proteins)
```

```
## [1] "P24534 [79-117]" "P14866 [63-97]"  "P14866 [63-97]"  "P14866 [63-97]"
## [5] "P14866 [63-97]"  "P14866 [63-97]"
```

```
byonic_target$Positions.in.Proteins[grep(";",
    byonic_target$Positions.in.Proteins)[1:5]]
```

```
## [1] "P68400 [22-43]; Q8NEV1 [22-43]"     "P46013 [1959-1982]; [2442-2465]"
## [3] "Q9UMR2 [68-82]; Q9NUU7 [67-81]"     "Q3ZCM7 [123-154]; P68371 [123-154]"
## [5] "Q3ZCM7 [123-154]; P68371 [123-154]"
```

which contains the Uniprot protein accession numbers in front of the square bracket and the positions of the sequence inside the square bracket. For example, "P24534 [79-117]" means that this peptide goes from the 79th amino acid residue to the 117th amino acid residue. If a peptide sequence could come from multiple proteins, these proteins and positions are separated by ";" followed with a single space.

3.1.7 Modifications

Users need to specify the column names of master proteins in the target search. The modifications should be formatted in the Proteome Discoverer format:

```
head(byonic_target$Modifications)
```

```
## [1] "N-Term(TMT6plex); K18(TMT6plex); S28(Phospho); K38(TMT6plex)"
## [2] "N-Term(TMT6plex); K35(TMT6plex)"
## [3] "N-Term(TMT6plex); K35(TMT6plex)"
## [4] "N-Term(TMT6plex); K35(TMT6plex)"
## [5] "N-Term(TMT6plex); K35(TMT6plex)"
## [6] "N-Term(TMT6plex); K35(TMT6plex)"
```

where each modification is separated by ";" followed with a single space.

3.1.8 Abundances

If users are interested in combining quantification information, columns of PSM abundances also need to be specified. For example, the abundance information in Byonic are

```
head(byonic_target[, c("Abundance:.126",
    "Abundance:.127", "Abundance:.128", "Abundance:.129",
    "Abundance:.130", "Abundance:.131")])
```

```
##   Abundance:.126 Abundance:.127 Abundance:.128 Abundance:.129 Abundance:.130
## 1          122.9          325.0          166.6          149.4          300.8
## 2          171.1          186.5          243.5          167.6          232.2
## 3          219.6          214.0          217.5          251.2          236.8
## 4          115.9          108.4          109.3          126.6          125.8
## 5          307.7          365.3          476.5          308.6          427.6
## 6          205.1          238.6          288.3          217.0          279.9
##   Abundance:.131
## 1          178.6
## 2          205.6
## 3          204.3
## 4          104.2
## 5          409.1
## 6          266.2
```

3.2 Combining methods

Once we preprocess the PSM-level output from Byonic and MaxQuant, we could input them into APIR using the following code:

```
library(APIR)
data("PhosphoSitePlus")
target_ls = list(byonic = byonic_target,
    maxquant = maxquant_target)
decoy_ls = list(byonic = byonic_decoy, maxquant = maxquant_decoy)
re = apir(saveas = "apir_byonic&maxquant.xlsx",
    target_ls = target_ls, decoy_ls = decoy_ls,
    scoreColTitle = c("q-Value.2D", "PEP"),
    scannumColTitle = c("First.Scan", "MS/MS.scan.number"),
    sequenceColTitle = c("Sequence", "Sequence"),
    fileColTitle = c("Spectrum.File", "Raw.file"),
    modificationColTitle = c("Modifications",
        "Modifications.ProteomeDiscoverer.Format"),
    masterproteinColTitle = c("Master.Protein.Accessions",
        "Leading.razor.protein.(format.fixed)"),
    proteinPositionColTitle = c("Positions.in.Proteins",
        "Positions.in.Proteins"), abundanceColTitle = list(byonic = c("Abundance:.126",
        "Abundance:.127", "Abundance:.128",
        "Abundance:.129", "Abundance:.130",
        "Abundance:.131"), maxquant = c("Reporter.intensity.corrected.1",
        "Reporter.intensity.corrected.2",
        "Reporter.intensity.corrected.3",
        "Reporter.intensity.corrected.4",
        "Reporter.intensity.corrected.5",
        "Reporter.intensity.corrected.6")),
    ifadjust = c(TRUE, TRUE), ifAggregateAbundance = TRUE,
    ifRecommendMasterProtein = TRUE, ifRecommendModification = TRUE,
    staticModification = c("C", "K", "N-Term"),
    phospho_dataset = PhosphoSitePlus, organism = "human",
    FDR = 0.05, ncores = 5)
```

```
## [1] "Performing protein inference and aggregating abundances..."
## [1] "Formatting the final output..."
```

```
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, :
## Number of characters exeed the limit of 32767.
```

4 References

Yi Fang et al. “Quantitative phosphoproteomics reveals genistein as a modulator of cell cycleand DNA damage response pathways in triple-negative breast cancer cells”. In:Internationaljournal of oncology48.3 (2016), pp. 1016–1028.