Tutorial of integrating KGMN results with other in-silico MS/MS workflows

Zhiwei Zhou

2022-06-10

Introduction

Knowledge-guided multi-layer network (KGMN) is a new approach leveraging knowledge-guided multi-layer networks to annotate known and unknown metabolites in untargeted metabolomics data. Although KGMN is an independent software tool, it can further integrate with other workflows to help users discover and validate metabolites. This tutorial aims to provide an easy instruction to integrated KGMN results with 3 common in-silico MS/MS tools (MetFrag, CFM-ID, MS-FINDER).

Here, we mainly focus on providing ways to help users linking KGMN with other tools. It should be note that the parameters need to be adjusted according to their instrument settings and experimental designs. **The detailed usage please refer their own tutorials.**

Tutorials:

- MetFrag: https://ipb-halle.github.io/MetFrag/
- CFM-ID: https://cfmid.wishartlab.com/
- MSFINDER: https://mtbinfo-team.github.io/mtbinfo.github.io/MS-FINDER/tutorial.html

Demo datasets:

- NIST urine set (Positive mode, processed by KGMN): **Download**

References:

If you use these tools, please cite their papers.

- MetFrag: MetFrag relaunched: incorporating strategies beyond in silico fragmentation. *J Cheminform* 8, 3 (**2016**). **DOI**
- CFM-ID: CFM-ID 4.0: More Accurate ESI MS/MS Spectral Prediction and Compound Identification. *Anal Chem.* 93, 34 (**2021**). **DOI**
- MSFINDER: Hydrogen rearrangement rules: computational MS/MS fragmentation and structure elucidation using MS-FINDER software. *Anal Chem.* 88, 16 (**2016**). **DOI**

1. Installation.

This integration of KGMN and in-silico MS/MS tools is mainly performed by R package "MetDNA2InSilicoTool". It can be downloaded as below:

```
# Install required packages
if(!require(devtools)){
install.packages("devtools")
}

if(!require(BiocManager)){
install.packages("BiocManager")
}

# Install CRAN/Bioconductor packages
required_pkgs <- c("dplyr","tidyr","readr","stringr","rcdk")
list_installed <- installed.packages()

new_pkgs <- required_pkgs[!(required_pkgs %in% list_installed[,'Package'])]
if (length(new_pkgs) > 0) {
  BiocManager::install(new_pkgs)
} else {
  cat('Required CRAN/Bioconductor packages installed\n')
}



# Install GitHub packages - call MetFrag
devtools::install_github("schymane/ReSOLUTION")

# Install GitHub packages
devtools::install_github("ZhuMetLab/MetDNA2InSilicoTool")
```

2. MetFrag

MetFrag is a common in-silico MS/MS tool developed by *Dr. Sebastian Wolf* and *Dr. Christoph Ruttkies*. It provides multiple ways to use it, including web server (MetFragWeb), MetFrag commandline tool (MetFragCL) and R package (MetFragR). In this workflow, we mainly use **MetFragCL (version 2.4.5)** to demonstrate the connection between KGMN and MetFrag.

2.1 Download MetFragCL program.

MetFragCL is a Java Archive File. It can be downloaded from GitHub. https://github.com/ipb-halle/MetFragRelaunched/releases/tag/v2.4.8

software > metfrag

Name	Date modified	Type	Size
MetFrag2.4.5-CL.jar	5/21/2019 10:00 PM	Executable Jar File	45,560 KB

Note: The MetFragCL program is depended on **Java**. Please install java and set environment variable first.

2.2 Load required packages, and setting the working directory.

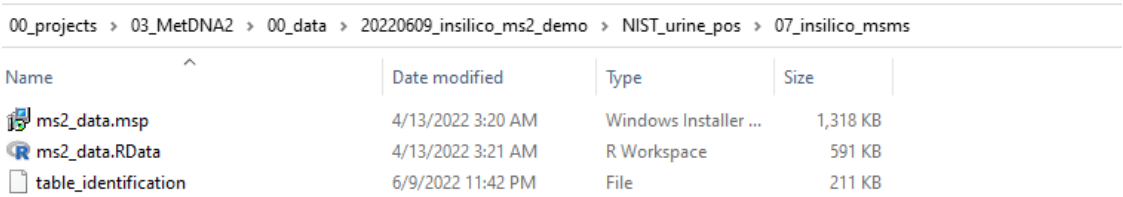
We use MetDNA2InSilicoTool to call MetFragCL. Please set the working directory at 07_insilico_msms, which is localized at KGMN result folder. Then, load some required packages.

```
# set working directory
setwd('G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')

# load packages
library(dplyr)
library(MetDNA2InSilicoTool)

# reformat identification_table
reformatTable1(dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

It looks like as below:



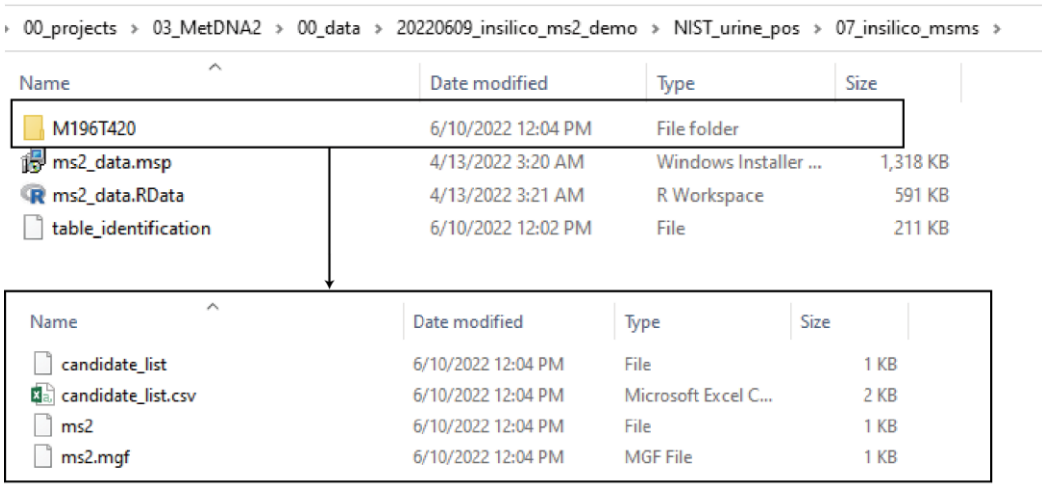
2.3 Generate input files for your interested peak.

In this workflow, users need generate necessary files for different in-silico tools. Here, we use an interesting peak **M196T420** as example (Figure 4c). This peak is annotated as an unknown peak in KGMN, while it has 6 possible metabolite candidates.

First, generate necessary file for M196T420.

```
# generate files for in-silico MS/MS match
# peak 'M196T420' as example
generateFiles4InsilicoMsMs(peak_id = 'M196T420',
                           dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

A folder “M196T420” will be created as blow:



It contains two files, candidate_list and MS/MS file. The **candidate list** is a list of chemical structures for in-silico MS/MS tool validation. The **MS/MS file** is a experimental spectrum of the targeted peak. The MS/MS file can be used for other in-silico tools if needed.

2.4 Run MetFrag.

We provide a R function (runMetFragMatch) to call MetFragCL. Here, the path of MetFragCL should be given. Other parameters can be adjusted. In MetDNA2InSilicoTool package, we only open limited parameters. For advanced users, the parameters can be adjusted according to MetFragCL tutorial.

```
# run MetFrag

# parameters
# peak_id: name of interested peak
# metfrag_path: path of metfrag program
# ppm: relative error of precursor MS1. 25 ppm
# mzabs: absolute error or MS1. 0.01 Da
# frag_ppm: relative error of precursor MS1. 25 ppm

runMetFragMatch(peak_id = 'M196T420',
            dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/',
            metfrag_path = 'F:/software/metfrag/MetFrag2.4.5-CL.jar',
            ppm = 25,
            mzabs = 0.01,
            frag_ppm = 25)
```

2.5 Output of MetFrag.

A folder “01_metfrag” is created in the “M196T420” folder. It contains results of MetFrag. For candidate with different adducts, they are divided into different folders. The rank results localize at the subfolder “results”.

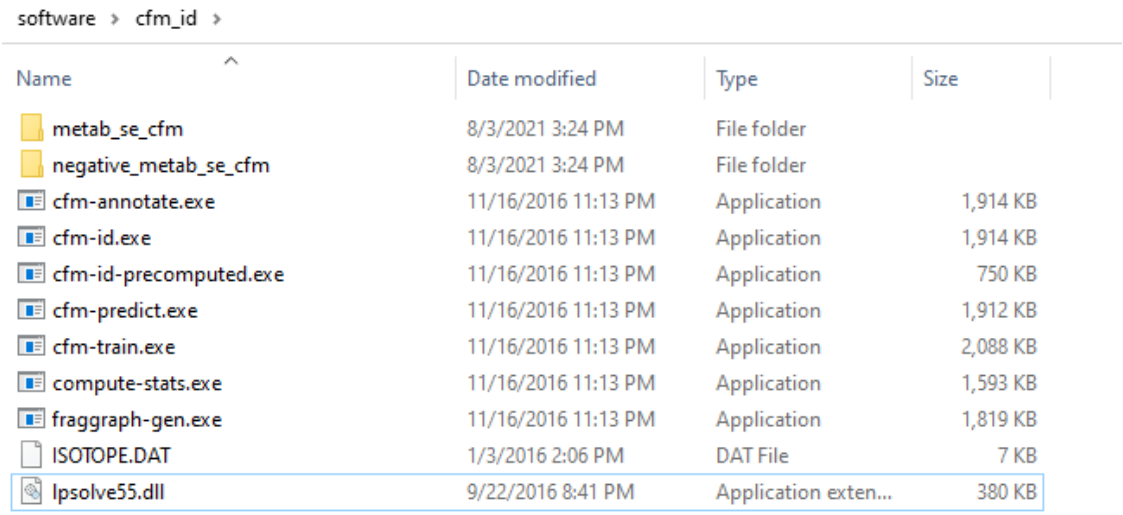
3. CFM-ID

CFM-ID is a machine-learning based MS/MS prediction tool, which developed by *Prof. David S Wishart Lab*. It provides several access ways, including web server and command lines. In this workflow, we mainly use CFM-ID (version 2.4) to demonstrate the connection between KGMN and CFM-ID

.

3.1 Download and Set CFM-ID program.

Here, we utilize CFM-ID (v2.4). The program can be downloaded at here. The new docker image of CFM-ID4 is available at here .



Note:

- The prediction model is required for CFM-ID. Users can train their own model or directly use the pre-trained model. The predicted model can be downloaded at **here**.

3.2 Load required packages, and setting the working directory.

Similar with MetFrag, we use MetDNA2InSilicoTool to call CFM-ID. Please set the working directory at 07_insilico_msms, which is localized at KGMN result folder. Then, load some required packages.

```
# set working directory
setwd('G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')

# load packages
library(dplyr)
library(MetDNA2InSilicoTool)

# reformat identification_table
reformatTable1(dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

3.2 Generate input files for your interested peak.

This step is consistent with MetFrag. We use an interesting peak M196T420 as example.

```
# generate files for in-silico MS/MS match
# peak 'M196T420' as example
generateFiles4InsilicoMsMs(peak_id = 'M196T420',
                           dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

3.3 Run CFM-ID.

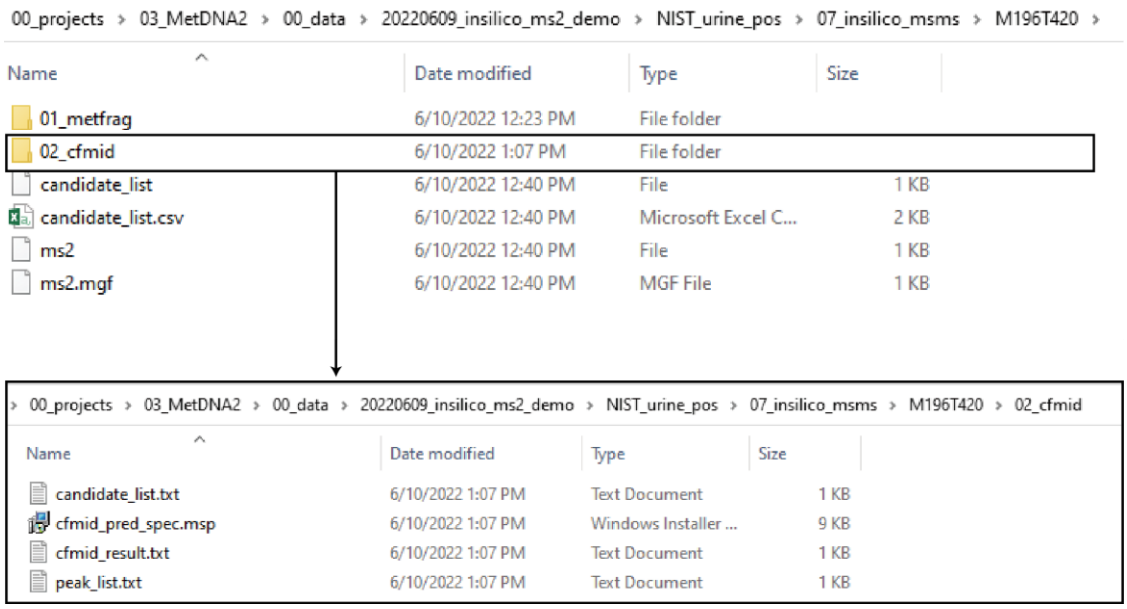
```
# run CFM-ID

# parameters
# cfmid_path: path of cfm-id
# config_file: config file of prediction model. It should be selected according to ionzation polairty. Pos: metab_se_cfm/param_config.txt; Neg: negative_metab_se_cfm/param_config.txt
# param_file: parameter file of prediction model. It should be selected according to ionzation polairty. Pos: metab_se_cfm/param_output0.log; Neg: negative_metab_se_cfm/param_output0.log
# score_type: rank score of CFM-ID. Default: 'jaccard'
# ppm: relative mz tolerance
# mzabs: absolute mz tolerance

runCfmIdMatch(peak_id = 'M196T420',
        dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/',
        cfmid_path = 'F:/software/cfm_id/cfm-id.exe',
        config_file = 'F:/software/cfm_id/metab_se_cfm/param_config.txt',
        param_file = 'F:/software/cfm_id/metab_se_cfm/param_output0.log',
        score_type = 'Jaccard',
        ppm = 25,
        mzabs = 0.01)
```

3.4 Output of CFM-ID.

A folder “02_cfmid” will be created in the “M196T420” folder. It contains results of CFM-ID. The “cfmid_result.txt” is the CFM-ID rank result. The “cfmid_pred_spec.msp” is the predicted MS/MS spectra of candidates.



4. MS-FINDER

MS-FINDER is a rule-based fragmentation tool, which developed by *Prof. Hiroshi Tsugawa* and *Prof. Masanori Arita* Lab. It usually is combined with MS-DIAL. In this tutorial, we mainly used it command tool (version 3.2.4) to evaluate KGMN metabolites.

4.1 Download MS-FINDER program.

We used the MS-FINDER v3.24. The newest version can be downloaded from here.

Note: The instruction of MetDNA2InSilicoTool is only supported and tested in Windows System.

software › MSFINDER › MSFINDER_ver_3.24

Name	Date modified	Type	Size
IKVM.OpenJDK.Text.dll	1/15/2015 3:02 PM	Application exten...	801 KB
IKVM.OpenJDK.Util.dll	1/15/2015 3:02 PM	Application exten...	1,950 KB
IKVM.OpenJDK.XML.API.dll	1/15/2015 3:02 PM	Application exten...	201 KB
IKVM.OpenJDK.XML.Parse.dll	1/15/2015 3:02 PM	Application exten...	2,619 KB
IKVM.Runtime.dll	1/15/2015 3:02 PM	Application exten...	1,016 KB
IKVM.Runtime.JNI.dll	1/15/2015 3:02 PM	Application exten...	76 KB
IsotopeRatioCalculator.dll	6/2/2019 5:13 PM	Application exten...	32 KB
MassLynxRaw.dll	5/10/2018 10:39 AM	Application exten...	738 KB
MassLynxRawSDK.dll	5/10/2018 10:39 AM	Application exten...	24 KB
MassSpectrogram.dll	6/10/2019 5:04 PM	Application exten...	97 KB
MassSpectrogram.dll.config	9/20/2018 11:43 AM	CONFIG File	4 KB
Mathematics.dll	5/5/2016 12:04 PM	Application exten...	24 KB
MessagePack.dll	1/30/2018 3:19 PM	Application exten...	273 KB
MolecularFormulaFinder.dll	6/10/2019 5:02 PM	Application exten...	135 KB
MsdialGcmsProcess.dll	6/10/2019 5:03 PM	Application exten...	156 KB
MsdialLcmsProcess.dll	6/10/2019 5:03 PM	Application exten...	324 KB
MSFINDER.exe	6/10/2019 5:04 PM	Application	1,235 KB
MSFINDER.exe.config	9/20/2018 11:43 AM	CONFIG File	4 KB
MSFINDER.INI	5/28/2019 11:57 AM	Configuration sett...	3 KB
MsfinderCommon.dll	6/10/2019 5:04 PM	Application exten...	54 KB
MsfinderConsoleApp.exe	6/10/2019 5:02 PM	Application	194 KB
MsfinderConsoleApp.exe.config	11/21/2018 5:36 PM	CONFIG File	4 KB

4.2 Load required packages, and setting the working directory.

Repeat procedures in MetFrag and CFIM-ID. Set the working directory at 07_insilico_msms, which is localized at KGMN result folder. Then, load some required packages.

```
# set working directory
setwd('G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')

# load packages
library(dplyr)
library(MetDNA2InSilicoTool)

# reformat identification_table
reformatTable1(dir_path = 'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

4.3 Generate input files for your interested peak.

Consist with **MetFrag** and **CFM-ID**, generate related files for targeted peaks. The example M196T420 is here.

```
# generate files for in-silico MS/MS match
# peak 'M196T420' as example
generateFiles4InsilicoMsMs(peak_id = 'M196T420',
                           dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms/')
```

4.4 Run MS-FINDER

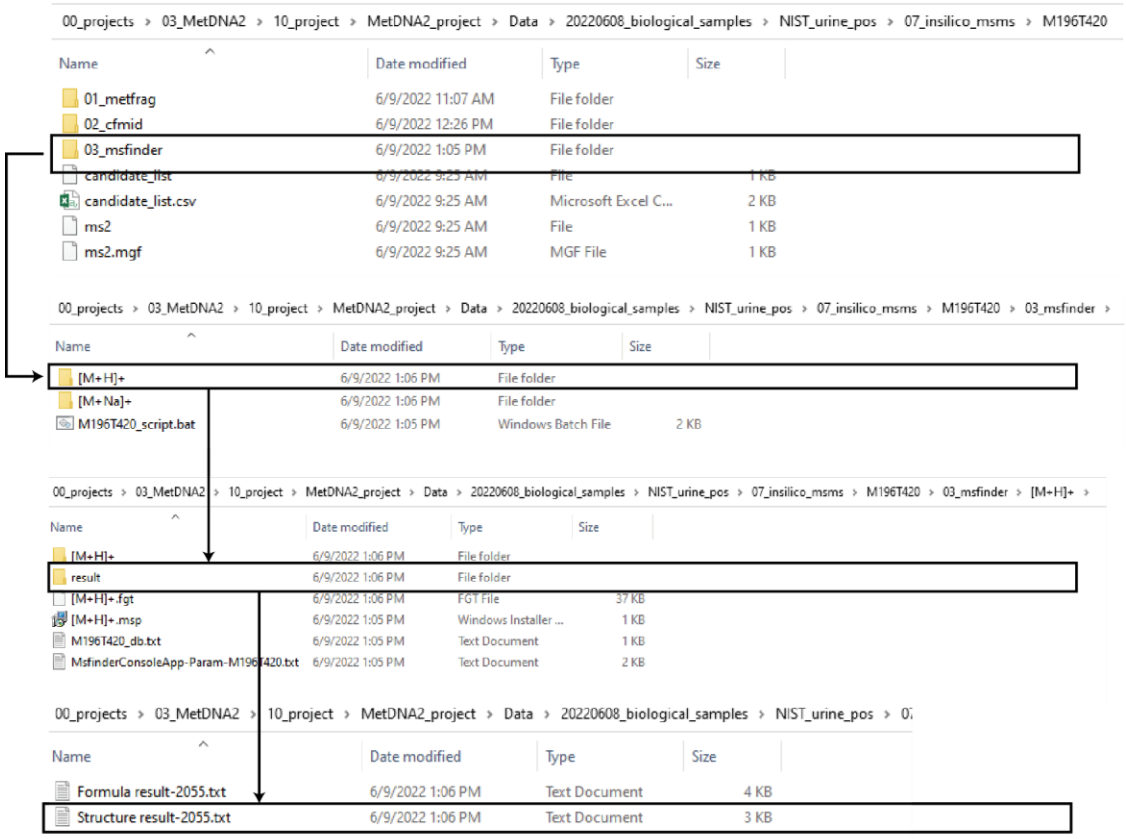
We provided a R function (runMsFinderMatch) to call MS-FINDER. Here, we use the command tool of MS-FINDER (MsfinderConsoleApp.exe). The path of MS-FINDER should be given.

```
# run MS-FINDER

# parameters
#
runMsFinderMatch(peak_id = 'M196T420',
                 dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms',
                 msfinder_path = 'F:/software/MSFINDER/MSFINDER_ver_3.24/MsfinderConsoleApp.exe')
```

4.5 Output of MS-FINDER.

A folder “03_msfinder” will be created in the “M196T420” folder. It contains results of MS-FINDER. The result of MS-FINDER is organized as adduct types. The rank result will be 03_msfinder -> [M+H]+ -> result -> Structure result-2055.txt.



Note:

- The parameter file of MS-FINDER is in '/03_msfinder/[M+H]+/MsfinderConsoleApp-param.txt'. Advanced users can adjust this file, and rerun MS-FINDER.

5. The script for connection KGMN and in-silico MS/MS tools

Here is a script contains above codes to help to connect KGMN and in-silico MS/MS tools quickly.

```
# set working directory
setwd('G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/')

# load packages
library(dplyr)
library(MetDNA2InSilicoTool)

# copy files
copyFiles4InsilicoTool(dir_path = '.')

# set working directory
setwd('G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilic
o_msms/')

# reformat identification_table
reformatTable1(dir_path = '.')


# generate files for in-silico MS/MS match
# peak 'M196T420' as example
generateFiles4InsilicoMsMs(peak_id = 'M196T420')

# run MetFrag
runMetFragMatch(peak_id = 'M196T420',
            metfrag_path = 'F:/software/metfrag/MetFrag2.4.5-CL.jar',
            ppm = 25,
            mzabs = 0.01,
            frag_ppm = 25)


# run CFM-ID
runCfmIdMatch(peak_id = 'M196T420',
          cfmid_path = 'F:/software/cfm_id/cfm-id.exe',
          config_file = 'F:/software/cfm_id/metab_se_cfm/param_config.txt',
          param_file = 'F:/software/cfm_id/metab_se_cfm/param_output0.log',
```

```
          score_type = 'Jaccard',
          ppm = 25,
          mzabs = 0.01)

# run MS-FINDER
# note: the dir_path must be given
runMsFinderMatch(peak_id = 'M196T420',
             dir_path =
'G:/00_projects/03_MetDNA2/00_data/20220609_insilico_ms2_demo/NIST_urine_pos/07_insilico_msms',
             msfinder_path = 'F:/software/MSFINDER/MSFINDER_ver_3.24/MsfinderConsoleApp.exe')
```