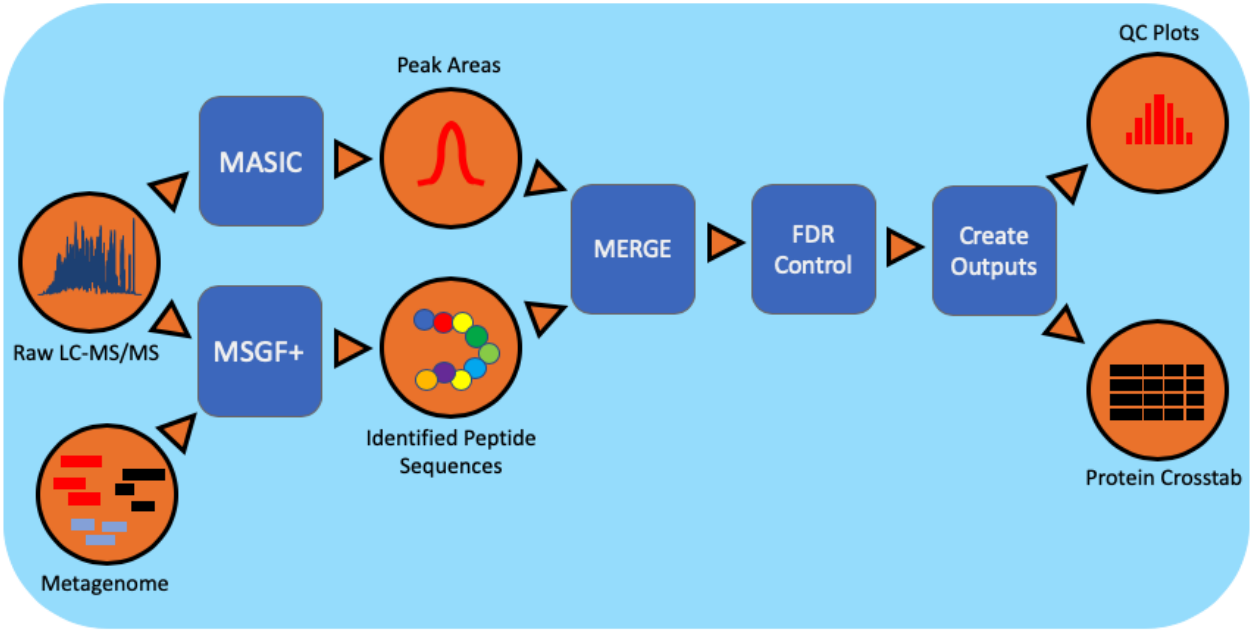
Meta-Proteomics Workflow

Release 1.0

Anubhav

Aug 06, 2020

WORKFLOW DESCRIPTION:


QC Plots
Peak Areas
MASIC
MERGE
FDR Control
Create Outputs
Raw LC-MS/MS
MSGF+
Identified Peptide Sequences
Protein Crosstab
Metagenome

CHAPTER

ONE

ABOUT

Meta-proteomics workflow/pipeline is an end-to-end data processing and analyzing pipeline for studying proteomes i.e studying protein identification and characterization using MS/MS data.

We identify the active organisms/species in a metagenome corresponding to a wet-lab sample obtained from JGI after gene sequencing.

Then the researchers at PNNL culture these samples and make it appropriate to study it as a protein sample. This protein sample may have a single protein or a complex mixture of proteins.

Later, this sample is passed through a mass spectrometry instrument to obtain a proprietary data format .RAW file. This file contains MS/MS spectrum i.e mass analysis(mass-to-charge (m/z) ratios) for each peptide sequences identified in the sample.

Additionally, we need sequenced metagenomes for each datasets. Currently, The metagenomes are obtained from NERSC(JGI) in the of FASTA(*.faa*) files.

This workflow kicks in after .raw and .faa are available.

1.1 Table of Content

1.1.1 Overview of components:

Processing

Third party packages

Packages	Version	Description
`MS-GF-Plus`	v20190628	performs peptide identification by scoring MS/MS spectra against peptides derived from a protein sequence database (FASTA files). Individual peptide sequences are identified(mzML file), then the set of peptide sequences is used to infer which proteins may have been present. It reads an open data format for MS/MS identification i.e mzML files and searches again a protein database(FASTA) to outputs a .mzId file which constitutes a set of scored PSMs along with E-value estimates(i.e computes E-values of PSMs and estimates FDRs)
Mzid-To-Tsv-Converter	v1.3.3	to converts MS-GF+ output (.mzid) into the tsv format (.tsv).
Peptide-HitResultsProcessor	v1.5.7130	converts tsv format (.tsv) to syn.txt file used for downstream analyses.
pwiz-bin-windows	x86_64-vc141-release-3_0_20149_b73158966	
MASIC	v3.0.7235	extracts intensity information for the identified peptides. It accurately measures peptide abundances and elution times in an LC-MS/MS analysis. It reads .Raw files from Thermo Fisher mass spectrometers and generates .SIC files i.e “selected ion chromatograms” (SICs) for each species chosen for MS/MS fragmentation.
sqlite-netFx-full-source	1.0.111.0	
msConvert		A command-line tool converting to/from various mass spectrometry data formats including multiple proprietary formats. It converts.RAW to mzML. Converting to mzML which is an open data format makes easy for academic scientists to directly manipulate MS/MS spectrum data. Open formats enable improved data sharing by allowing the data to be read by a variety of software tools without licensing restrictions
Conda	(3-clause BSD)	

Data flow diagram:

Input:

- *.raw*, *metagenome*, *parameter files : MSGFplus & MASIC*, *contaminant_file*

Output:

- *.mzml*, *mzid*, *tsv*

Credits:

1. MoTrPAC.
2. **MS-GF+: Universal Database Search Tool for Mass Spectrometry.** Sangtae Kim, Pavel A. Pevzner, Nat Commun. 2014 Oct 31;5:5277. doi: 10.1038/ncomms6277.

   This software is Copyright © 2012, 2013 The Regents of the University of California. All Rights Reserved. Permission to copy, modify, and distribute this software and its documentation for educational, research and non-profit purposes, without fee, and without a written agreement is hereby granted, provided that the above copyright notice, this paragraph and the following three paragraphs appear in all copies.
3. **MASIC+: (MS/MS Automated Selected Ion Chromatogram generator) a software program for fast quantitation and flexi**
   Matthew E. Monroe and Shaw, Jason L and Daly, Don S and Adkins, Joshua N and Smith, Richard D Licensed under the 2-Clause BSD License; you may not use this file except in compliance with the License. You may obtain a copy of the License at https://opensource.org/licenses/BSD-2-Clause

Moving on

Now it is time to move on to *Aggregating analysis results*.

Aggregating analysis results

Overview of software:

1. Analyzing component:
   1. Reads in syn.txt files & calculated the best scoring peptides for each scan.
   2. Even though, MSGF+ estimates False discovery rates(FDRs) in some datasets MSGFplus tool when dealing with SPLIT FASTAs(multiple FASTA for the same sample) doesn't actually account all of them due to which the QValue and PepQValue value aren't based on the entire FASTA file for that dataset. Therefore, we're recomputing QValue and PepQValue to improve the FDR.
   3. merges the outputs from MSGF+ and MASIC, and applies to filter to control the false discovery rate. The output is a crosstab format table with rows containing protein sequence information, and columns with relative abundance measurements for proteins identified in each sample analyzed.

Merging Algorithm:

```
- Given an User-input:
    - Generate
       - start_file.xlsx
       - job_info_query.xlsx
    - For each dataset:
        **Algorithm for merging MSGF+ and MASIC analysis job results**
        - Combine multiple `MSGFplusJob results`_:
            1. Merge "*_msgfplus_syn.txt" files based on scan_id.
            2. keep the best scoring peptide(minimum MSGFDB_SpecEValue candidate) for each scan.
            3. Generate `consolidate_syn`( big peptide file).
            4. Generate,`recomupted_consolidate_syn`
              [improving the "False Discovery Rate" algorithm on `Consolidate_syn`.](###Algorithm-to-Recompute-the-qvalues)
            5. Obtain protein information from below files & merge it with `recomupted_consolidate_syn`.
                    `*_msgfplus_syn_SeqToProteinMap.txt`(protein file) and
                    `*_msgfplus_syn_ResultToSeqMap.txt` {mapper file}
            6. Generate `MSGFjobs_Merged`.
        - Combine `MSGFjobs_Merged` with MASIC results
            1. Merge on the basis of `Scan_id` in `_SICstats.txt`.
            2. Generate `MSGFjobs_MASIC_resultant`.
```

Recompute the qvalues Algorithm:

```
 In some datasets MSGFplus tool dealing with SPLIT FASTAs due to which the `QValue` and `PepQValue` value aren't
 based on the entire FASTA file for that dataset. Therefore, we're recomputing these 2 columns after we get
 `Consolidate_syn` (by merging all the MSGFplusjobs ran on multiple FASTAs)
  1) Using the Consolidate_syn object
  3) For each scan, select the peptide with the lowest Spec EValue
      - If there is more than one peptide with a tied spec evalue, select both of them
  4) Examine the proteins for the selected peptide (or peptides) in the given scan
       - If all of the proteins start with XXX_, this is a Decoy (aka reverse) PSM
       - Otherwise, if any of the proteins does not start with XXX, this is a forward PSM
  5) Remove the prefix and suffix letters from the peptide
      - For example, given peptide K.SPVGKS*PPSTGSTYGSSQKEESAASGGAAYTKR.Y, remove K. and .Y to get SPVGKS*PPSTGSTYGSSQKEESAASGGAAYTKR
      - Call this the base peptide
  6) Look for that base peptide in a new in-memory table called the UniquePeptide table
      - If the table does not have the peptide, add it, storing BasePeptide, SpecEValue, and IsDecoy=True or false
      - If the table does have the peptide, and if the SpecEValue for the base peptide in this scan is lower, update the update the entry for the peptide to have the lower SpecEValue
  7) Once all scans are processed, your new in-memory table will have one row per base peptide
      - If you had encountered these two peptides, you would have two rows; that's OK (and appropriate, since the * symbol is in different locations)
```

(continues on next page)

(continued from previous page)

```
         SPVGKS*PPSTGSTYGSSQKEESAASGGAAYTKR
      SPVGKSPPSTGSTYGSS*QKEESAASGGAAYTKR
   8) Sort this unique peptide table on SpecEValue, then IsDecoy, then peptide (thus, for entries with the same SpecEValue, the IsDecoy=false ones should be first)
   9) Step through that table and count the number of Forward hits and the number of Decoy hits, up to the current row
   10) Compute DecoyCount / ForwardCount; this is the PepQValue
   11) When the DecoyCount value increments by one, assign the DecoyCount / ForwardCount value for the previous row to all previous rows with the same DecoyCount value
   12) Once all rows have been processed, go back to the original table and assign an updated PepQValue by linking the two tables on Peptide (linking on Base Peptide)
   13) Report both the original PepQValue and your re-computed PepQValue in the output table so that we can assure that they're similar.
```

Data flow diagram:

Input:

Output:

- start_file.csv

Dataset_ID	MSGFPlusJob	MSGFplus_loc	NewestMasicJob	MASIC_loc

- job_info_query.csv

Job	Dataset	Experiment	OrganismDBName	ProteinCollectionList	ParameterFileName

- consolidate_syn.csv

Result-ID	Scan	Frag-Method	Spec-Index	Charge	Precursor-MZ	DelM	DelM_PPM	MH	Peptide	Protein	NTT	De-Novo-Score	MS-GF-Score	MSGF-DB_SpecEValue	Rank_SpecEValue	MSGF_QValue	MSGF_PepQValue	Decoy	Cleavage_State	Seq_ID	Dataset_Num

- recomputed_consolidate_syn.csv

Result-ID	Scan	Frag-Method	Spec-Index	Charge	Precursor-MZ	DelM	DelM_PPM	MH	Peptide	Protein	NTT	De-Novo-Score	MS-GF-Score	MSGF-DB_SpecEValue	Rank_SpecEValue	MSGF_QValue	MSGF_PepQValue	Decoy	Cleavage_State	Seq_ID	Dataset_Num	Dataset	re-computed-PepQ-Value

- MSGFjobs_Merged.csv

Result-ID	Scan	Frag-Method	Spec-Index	Charge	Precursor-MZ	DelM	DelM_PPM	MH	Peptide	Protein	NTT	De-Novo-Score	MS-GF-Score	MSGF-DB_SpecEValue	Rank_SpecEValue	MSGF_QValue	MSGF_PepQValue	Decoy	Cleavage_State	Seq_ID	Dataset_Num	Dataset	Unique	Cleavage_State	Seq_ID	Pro-tein_Num	Pro-tein_Expectation_Value	Intensity_Value	(I

- *MSGFjobs_MASIC_resultant.csv* and *resultants_df.csv*

ResultID	Scan	FragMethod	SpecIndex	Charge	PrecursorMZ	DelM	DelM_PPM	MH	Peptide	Protein	NTT	DeNovoScore	MSGFScore	MSGFDB_SpecEValue	Rank_MSGFDB_SpecEValue	EValue	QValue	PepQValue	IsotopeError	[illegible]	[illegible]	[illegible]	Cleavage_State	Terminus_State	[illegible]	Protein_Expectation_Value_Log(e)	Protein_Intensity_Log(I)	Dataset	ParentIonIndex	MZ	SurveyScanNumber	OptimalPeakApexScanNumber	PeakApexOverrideParentIonIndex	CustomSICPeak	PeakScanStart	PeakScanEnd	PeakScanMaxIntensity	PeakMaxIntensity	PeakSignalToNoiseRatio	FWHMInScans	PeakArea	ParentIonIntensity	PeakBaselineNoiseLevel	[illegible]

Moving on

Now it is time to move on to *Report generation*.

Report generation

Concepts

Sphinx Philosophy

- **Create a crossTab/ pivot table:**
  - Combine *MSGFjobs_MASIC_resultant* of all datasets.
- **Generate Metrics/ Visualisations**
  - Using *crossTab*
- **Report generation.** *TODO*

Input:

Output:

```
.
├── Data/
├── FDR_table.csv
├── Plots/
├── dataset_job_map.csv
├── peak_area_crosstab_by_dataset_id.csv
├── protein_peptide_map.csv
├── specID_table.csv
└── spectra_count_crosstab_by_dataset_id.csv
```

1.1.2 Overview datasets:

FICUS Datasets:

1. Hess
2. Stegen

1.1.3 Project: nmdc-proteomics-workflow

Package: src

Package : analysis

Module : ficusAnalysis

Module : internalAnalysis

class src.analysis.internalAnalysis.**downStreamAnalysis**(*parent_folder*)
Bases: object

findproteinname()
Get Protein type :param s: :return:

cleansequence()
clean peptide sequence is the sequence without prefix and postfix but with oxidation :param s: :return:

process_data()

Returns

src.analysis.internalAnalysis.**parameter_optimization**(*dataset_ID*)

Parameters **dataset_ID** –

Returns

Package : data_access

Package : data_prepare

Module : parseFasta

Package : via_DMS

Module : DMSDatabase

class src.data_access.via_DMS.DMSDatabase.**DMSDatabase**(*config*)
Bases: object

Database connection class

open_connection()
Connection to DMS MS sqlserver i.e DMS5 or DMS_Data_Package

run_query(*query*)
Execute SQL query.

Module : FileOperations

class src.data_access.via_DMS.FileOperations.**FileOperations**(*analysis_jobs=None*, *parent_folder=None*, *job_info=None*)

Bases: object

Grab locations of the MSGF+ & MASIC analysis tools using analysis_jobs object

create_dir(*folder*)

Parameters **folder** –

Returns

write_to_disk(*url: str*)

Parameters **url** – Job's file path on DMS.

Returns

check_url(*url*)

Parameters **url** –

Returns

download_over_http()
Given a url, copy files from DMS to disk! :return:

parse_fileserverpath_to_web_url(*file_server_path*)
Converts Windows FileSever path to webURL. :param file_server_path: windows server file path. :return:

download_msgf_jobs(*df*)

Parameters **df** –

Returns

download_masic_jobs(*df*)

Parameters **df** –

Returns

download_raw_files(*df*, *path_or_url*)

download_fasta_param_files()

Returns

use_df(*df*)
Called for each dataset in the dataFrame! :param df: reference to analysis_jobs object. :return:

get_files(***kw*)

Module : Input

class src.data_access.via_DMS.Input.**Input**
Bases: object

Handle & validate User input

other_input(*InputType*, *UserInput*)
changes input string to list of numbers.

user_input()

Returns

Module : Query

class src.data_access.via_DMS.Query.**Query**
Bases: object

SQL queries to access data from DMS

DATASET_MSFG = " SELECT A.Dataset_ID,\n A.MSGFPlusJob,\n B.MasicJob\n FROM (SELECT Da

MSGF_loc = 'SELECT JobNum As MSGFPlusJob, [Data Folder Link] As MSGFplus_loc\n FROM V_

DATASET_MASIC = " SELECT Dataset_ID, Max(Job) As NewestMasicJob\n FROM V_Analysis_Job_

MASIC_loc = 'SELECT JobNum AS NewestMasicJob , [Results Folder Path] As MASIC_loc\n FR

DATASET = " SELECT Dataset_ID, Job As MSGFPlusJob, [Results Folder Path] As MSGFplus_l

MSGF = 'SELECT Dataset_ID, Job As MSGFPlusJob,[Results Folder Path] As MSGFplus_loc \n

JOB_INFO = 'SELECT Job, Dataset, Experiment, OrganismDBName, ProteinCollectionList, Pa

Module : QueryBuilder

class src.data_access.via_DMS.QueryBuilder.**QueryBuilder**(*user_input=None*, *storage=None*, *project_name=None*)
Bases: object

1. Build MS-SQl Queries.
2. Execute them
3. **create a dataFrame that holds all Information** Dataset_ID | MSGFPlusJob | Data Folder Link | Newest-MasicJob | Results Folder Path |

save_to_disk(*data*, *data_path*, *msgf_job_list*, *id*)

Parameters

- **data** –
- **data_path** –
- **msgf_job_list** –
- **id** –

Returns

start_with_datapackage_id(*id*)
Given a ID —-Find out the Dataset_ID , MSGFPlusJob ———Using MSGFPlusJob, findout "Data Folder Link" ———Using Dataset_ID, findout NewestMasicJob ————Using NewestMasicJob findout "Results Folder Path" Merge results to create "analysis_jobs".

Parameters **id** – datapackage_id

Returns

start_with_dataset_ids(*id_list*)
Given set of dataset-IDs —-findout MSGFPlusJob, "Results Folder Path" ———Using Dataset_ID, findout NewestMasicJob ————Using NewestMasicJob findout "Results Folder Path" Merge results to create "analysis_jobs".

Parameters **id_list** – set of dataset-IDs

Returns

start_with_job_nums(*id_list*)
Given set of MSGFJobs —-Find the Dataset_ID, & "Results Folder Path" ———Using Dataset_ID, findout MASIC ————Using MASIC, findout "Results Folder Path" Merge results to create "analysis_jobs".

Parameters **id_list** – set of JobNums

Returns

execute()
Design queries here & set it

Module : secure

class src.data_access.via_DMS.secure.**Config**
Bases: object

db_user = None

db_password = None

db_server = None

db_name = None

Package : processing

Module : DatasetsMerger

class src.processing.DatasetsMerger.**DatasetsMerger**(*folder=None*, *combineDatasets=None*)
Bases: *src.processing.MSGFplusMerger.MSGFplusMerger*

1. **Run for UserInput:** a datapackage or a set of datasets or a set of MSGFJobNums
2. create a crossTab object

merge_all_jobs_in_UserInput()

1. Run for each dataset.
2. Merge all MSGFjobs_MASIC_resultant objects.

Returns

Module : MASICmerger

class src.processing.MASICmerger.**MASICmerger**(*folder*)
Bases: *src.processing.MSGFplusMerger.MSGFplusMerger*

Run for each dataset

merge_msgfplus_msaic(**kw*)

Module : MSGFplusMerger

class src.processing.MSGFplusMerger.**MSGFplusMerger**(*dataset_loc=None*)
Bases: object

Merge all MSFGjobs per dataset.

1. Runs for each dataset.
2. Collate "*msgfplus_syn.txt" & –> consolidate_syn object
3. Recompute the QValue and PepQValue –>recomupted_consolidate_syn object
4. Look for protein information into

   *msgfplus_syn_SeqToProteinMap.txt : protein Info. *msgfplus_syn_ResultToSeqMap.txt : Mapper –> MSGFjobs_Merged object

write_to_disk(*df*, *folder*, *file*)

Parameters

- **df** –
- **folder** –
- **file** –

Returns

fill_holes()

Returns

tackle_Unique_Seq_ID_holes_(*df*)

Parameters **df** –

Returns

get_protein_info(**kw*)

improve_FDR()
Recompute QValue` and PepQValue 1. Use consolidate_syn_DF

keep_best_scoring_peptide(**kw*)

stack_files(*grouped_files*, *file_pattern*)

Parameters

- **grouped_files** –
- **file_pattern** –

Returns

group_files(*folder*)

Parameters **folder** –

Returns

consolidate_syn_files()

1. For all jobs Read in(Stack): "*msgfplus_syn.txt" in _syn_DF with added JobNum & dataset column

Note: _syn_DF have duplicate rows for each Scan with MSGFDB_SpecEValue.

Returns

Module : MetProWorkflowApp

```
usage: MetProWorkflowApp.py [-h] [-M MODE] [-It {1,2,3}] [-S STORAGE]
                            [-P [PROJECTNAME]] [-I INPUT]
                            [-C [COMBINEDATASETS]] [-Sa {internal,ficus,both}]
```

Named Arguments

-M, --Mode **Different Modes to run the workflow?** Developer : Automatically generates files at each step! User : Generates CrossTab/Metric only!

-It, --InputType Possible choices: 1, 2, 3

Type of input 1 : a datapackage ID 2 : a list of dataset IDs 3 : a list of MSGFjobs Nums

-S, --Storage Path to store data & results of the pipeline.

-P, --ProjectName FICUS Study name Eg. hess/stegen/blanchard etc.

-I, --Input A valid input InputType: 1, An Integer InputType: 2, A comma-seperated list of Integers InputType: 3, A comma-seperated list of Integers

-C, --CombineDatasets Combine all dataset's MSGF & MASIC jobs to single file for generating crossTabs.

Default: False

-Sa, --SelectAnalysis Possible choices: internal, ficus, both

internal : Run internal analysis ficus : Run ficus analysis both : Run both analysis

Module : Workflow

class src.Workflow.**Workflow**(*mode=None*, *InputType=None*, *path_to_data=None*, *project_name=None*, *UserInput=None*, *CombineDatasets=None*, *SelectAnalysis=None*)

Bases: object

Automate the Meta-proteomics workflow

run_Analysis(*on_file*, *analysis_type*)

Run desired analysis on a file. :meta public: :param analysis_type: :return:

start_downStreamAnalysis(*result_path*)

Decides to run analysis on combined results vs single dataset. :meta public: :param result_path: :return:

start_merging(*folder*)
Start merging MSGF and MASIC jobs :meta public: :param folder: :return:

download_data_from_DMS(*user_obj*)
build & execute query to dowload data from DMS :meta public: :param user_obj: input from shell script. :return: path to !!

start_workflow()
Runs the workflow in 3 stages 1. Download relevant datasets from specified source. 2. Aggregation of analysis tools{MSGF+, MASIC} results: to extract useful data from datasets. 3. Generation of experimental report. :meta public: :return:

Package : utility

Module : gen_meta_data

Module : utils

utility.utils.**timeit**(*method*)
Calculate and logs runtime of a function. :param method: :return: string: HH:MM:SS

utility.utils.**str2bool**(*v*)
chnages userInput to a yes/no :param v: string :return: bool

utility.utils.**current_local_datetime**()

Returns current local date and time

utility.utils.**current_UTC_datetime**()

Returns current UTC date and time.

Please provide feedback Anubhav <anubhav@pnnl.gov>

1.2 Indices and tables

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

U

INDEX

J

K

M

O

P

Q

R

S

T

U

W