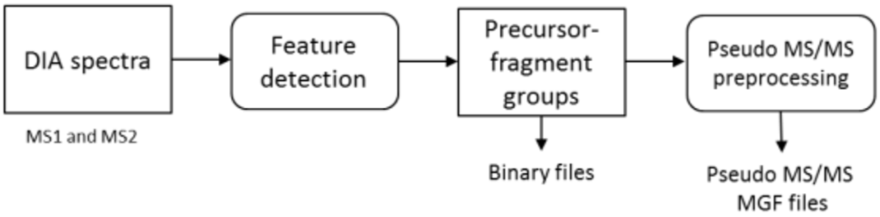
Tutorial 3: Library Generation with DIA-UMPIRE

In this tutorial, we will use the java-based tool DIA-Umpire to perform untargeted analysis of DIA data. We will take two DIA files of the LFQbench dataset (see tutorial overview) to perform DIA-Umpire signal-extraction (Step1) and database searching for spectral library and query parameter generation (Step2). This library will then be used in the next tutorial for targeted re-extraction and quantification with OpenSWATH.

1. Overview

The input for DIA-Umpire are converted and centroided mzXML files containing measured DIA spectra. The signal extraction algorithm detects all possible precursor and fragment ion features in the provided MS1 and MS2 data and groups them based on the correlation of their elution profiles. Thus, the module can generate **"pseudo-MS2 spectra"** that look like spectra generated from measurements in DDA mode.



The pseudo-MS2 spectra generated from the DIA-Umpire signal extraction module can be used for a regular database search to identify the peptides and proteins present in the measured DIA data.

Due to limited sensitivity and quantitative accuracy in the analysis of a single DIA file with DIA-Umpire signal extraction and database searching only, an additional quantification step is strongly recommended. For this, you can either use the DIA-Umpire quantification module directly or use multiple DIA runs to generate a spectral library, like you have done in the last tutorial, for targeted re-extraction and quantification with other tools.

In this tutorial we will use the results from the database search of the pseudo-MS2 spectra to generate a spectral library, which can be used for targeted re-extraction by OpenSWATH tomorrow.

2. DIA-Umpire workflow

2.1. DIA-Umpire signal extraction

In this first part of the tutorial we will use the signal extraction module of DIA-Umpire to extract all possible precursor and fragment ion features based on the correlation of their elution profiles. The end result will be the generation of a collection of pseudo-MS2 spectra in the form of mgf files, which we will then convert to mzXML files ready for database search.

We will run DIA-Umpire on only two exemplary DIA file containing mixture spectra from human, E.coli and yeast:

C:\DIA_Course\Data\DIA_data\TTOF\collinsb_I180316_001_SW-A.mzXML
C:\DIA_Course\Data\DIA_data\TTOF\collinsb_I180316_002_SW-B.mzXML

Or

```
C:\DIA_Course\Data\DIA_data\QE\collinsb_X1803_171.mzXML
C:\DIA_Course\Data\DIA_data\QE\collinsb_X1803_172.mzXML
```

Before starting the signal extraction module of DIA-Umpire one needs to revise the parameters file and tune it according to the experimental and machine setup. DIA-Umpire provides template parameter files for different instrument types. **We will adjust the parameter file for ABSciex data: *diaumpire_se_ABSciex_params.txt* or for QE data: *diaumpire_se_Thermo_params.txt***

- In RStudio open the respective file from the C: drive folder on your VM and move to the “Tutorial3_DIAUmpire”
- Here you find the default parameter files of DIA-Umpire in the parameter_files folder
- We will do our changes again in the RStudio environment, so please open the respective parameter template file.
- Review the following parameters and make sure you understand them. We will only mention the most important parameters here, the additional parameters not mentioned here usually do not need adjustments.

#Fragment intensity adjustment

⇒ **BoostComplementaryIon** For each detected complementary y/b ion pair (two peaks with masses that sum up to the precursor peptide mass), it sets the intensities of both complementary ions to the same (largest of the two intensities) value. When building spectral libraries using DIA-Umpire extracted pseudo-MS/MS spectra, such boosting is not recommended since low intensity fragments can be selected for inclusion in the library (at the expense of excluding better fragments) only because their intensities got artificially boosted. Turning this boosting option off decreases the number of IDs only slightly. In several representative AB Sciex 6600 and Thermo Fusion datasets tested, with boosting, the number of IDs (at 1% FDR) dropped by ~ 3% at the protein level and ~ 6% at the peptide ion level. Set this to '`false`' if you plan to build spectral libraries using DIA-Umpire results for subsequent targeted re-extraction using OpenSWATH or Skyline.

Caution! Change BoostComplementaryIon to false because we build a spectral library from the DIA-Umpire results!

#Signal extraction

⇒ **SE.MS1PPM and SE.MS2PPM:** These parameters establish the mass error (in ppm) to be used for MS1 and MS2 signal extraction. **For Orbitrap data the suggested values are 10-15 ppm (MS1) and 15-25 ppm (MS2), whereas for TOF instruments the suggested values are 30-40 ppm.** You don’t have to adjust this value for the tutorial.

⇒ **SE.Resolution:** This defines the resolution used during data acquisition and it is crucial for data generated in profile mode. In that case, recommended values are **30000-60000 (Thermo Fusion or QE HF) and 17000 (Sciex)**, but these values may need to be adjusted depending on the actual resolution settings of the instrument during data acquisition.

⇒ **SE.SN and SE.MS2SN** These parameters establish the minimal signal-to-noise ratio to perform signal extraction and they do not matter much as long as the values are low enough**. For Orbitrap data the recommended value is 1.1 whereas for Sciex TOF data the recommended value is 1.5**.

⇒ **SE.EstimateBG:** This parameter is used to automatically set a background noise. **For Sciex data this must be set to ‘`true`’** as Sciex data typically contain a lot of low intensity noise that needs to be removed before feature detection. Setting this parameter to ‘`true`’ will result in automatic detection and removal of the background noise signal. **In contrast, for Thermo data, no filtering is typically needed and this parameter should be set to '`false`'**.

- ⇒ **SE.MinMSIntensity** and **SE.MinMSMSIntensity:** If SE.EstimateBG = true (see above), these parameters are not used. If SE.EstimateBG = false, then one can apply these user-defined minimum intensity filters to remove the background before signal extraction. As mentioned above, for Thermo data no filtering is typically needed, and these parameters can be set to a small value, e.g. 1.
- ⇒ **SE.NoMissedScan:** Determines how many consecutive missing scans are allowed. **For Thermo data, set this to 2, whereas for Sciex, set this parameter to 1**.
- ⇒ **SE.StartCharge** and **SE.EndCharge:** These parameters define the charge state range for precursor ion detection in MS1 scans. Normally is set from +1 to +5.
- ⇒ **SE.MS2StartCharge and SE.MS2EndCharge:** These parameters define the charge state range for precursor ion detection in MS2 scans. Normally is set from +2 to +5.
- ⇒ **SE.MinFrag**: Minimum number of fragments needed for peak extraction. This parameter is normally set at 5.
- ⇒ **SE.StartRT** and **SE.EndRT**: These parameters define the range of the chromatogram at which the signal extraction will take place (units in minutes). You can leave the default values that consider the entire chromatographic RT dimension.
- ⇒ **SE.MinMZ**: Defines the minimal m/z value of the chromatogram at which the signal extraction ends. You can keep the 200 m/z that is the default.
- ⇒ **SE.MinPrecursorMass** and **SE.MaxPrecursorMass**: These parameters define the range of the precursor mass for signal extraction. Normally it ranges from 400 to 5000.

#Isolation window setting

- ⇒ **WindowType:** This parameter defines the isolation window schema and currently it supports the following window type: SWATH (fixed window size), V_SWATH (variable SWATH window), MSX, MSE, pSMART. Set this to V_SWATH
- ⇒ **WindowSize:** It defined the size for fixed windows. **Only valid for Sciex data and not required for Thermo data**.
- ⇒ **Variable SWATH window setting:** You can define the window isolation schema using the start m/z, end m/z, separated by Tab. ***This is what we will do!***

Caution! We need to specify the variable SWATH windows. For this you need to go to the Data folder and open **swath64_noheader.txt or dia_QE19_w_header.txt** in Notepad++.

- Copy the entire file content.
- Paste in between:
  ==window setting begin
  399.5 408.2
  407.2 415.8
  414.8 422.7
  …
  ==window setting end
- Save the parameter file as: *diaumpire_se_ABSciex_params_DIAcourse.txt or diaumpire_se_Thermo_params_DIAcourse.txt* respectively.

Now we are ready to run DIA-Umpire from the command line. ☺

- Open the bash script "`run_DIAUmpire_LibraryGeneration.sh`" from the folder Tutorial3_DIAUmpire.
- Like in Tutorial 1 we will first change the working directory. Therefore run the ommand on line 2:

```
cd /c/DIA_Course/Tutorial3_DIAUmpire/
```

- DIA-Umpire writes all output files to the directory of the input sample. Since we want to keep your data directory clean, we will copy our DIA file of interest to the folder for today's tutorial. The fist command for copying is already written in your script. Copy the second command as well (line 9) in order to copy the second file, too. Change the file names and directory if you are using the QE data. Run those commands:

```
cp /c/DIA_Course/Data/DIA_data/TTOF/collinsb_I180316_001_SW-A.mzXML \
/c/DIA_Course/Tutorial3_DIAUmpire/

cp /c/DIA_Course/Data/DIA_data/TTOF/collinsb_I180316_002_SW-B.mzXML \
/c/DIA_Course/Tutorial3_DIAUmpire/
```

  - **Note!** `cp` stands for copy and it moves the file specified in the first argument to the directory specified in the second argument.

- Now run the signal extraction module of DIA-Umpire using the following command (Again: **Remember to change the file names for the data and the parameter file if you use QE data**):

```
java -jar -Xmx8G DIA_Umpire_SE.jar \
collinsb_I180316_001_SW-A.mzXML \
./parameter_files/diaumpire_se_ABSciex_params_DIACourse.txt \
&>> Tutorial3_log.txt

java -jar -Xmx8G DIA_Umpire_SE.jar \
collinsb_I180316_002_SW-B.mzXML \
./parameter_files/diaumpire_se_ABSciex_params_DIACourse.txt \
&>> Tutorial3_log.txt
```

  - **Note!** java –jar is the standard command to run a java tool from the command line. The option –Xmx8G specifies that 8 GB of ram should be used for the job. This might have to be adjusted depending on the files to be analyzed with DIA-Umpire. Generally, all paths need to be adjusted to the file system you are using.

You will see that your directory is being filled with different output files from DIA-Umpire. For today's tutorial, we will only focus on the final output mgf files that contain all the pseudo-MS2 spectra classified in different quality levels according to how the precursor ion was detected: Q1, Q2 and Q3.

- **Q1 file** contains MS2 spectra corresponding to peak groups with more than two isotopic peaks detected in MS1 spectra
- **Q2 file** contains MS2 spectra corresponding to peak groups with only two isotopic peaks detected in MS1 spectra
- **Q3 file** contains MS2 spectra corresponding to peak groups for which the precursor information was derived from the detected unfragmented precursor in MS2 spectra

  - **Note!** Each file corresponds to a different "quality level" of precursor ions. These spectra are written to separate files because they must be searched separately against a protein sequence database as a consequence of differences in FDR estimates for these varying quality data

In order to prepare the DIA-Umpire output for the database searching and library generation step, we now convert all mgf files to mzXML files using MSConvert.

- Run MSConvert using the following command:

```
msconvert --mzXML *.mgf \
&>> Tutorial3_log.txt
```

- **Note!** The option --mzXML means that you convert to mzXML. The asterisk (*.mgf) means that any file that ends with the extension mgf will be converted.

2.2. Database search & spectral library generation

In this second part of the tutorial we will generate a spectral library from the pseudo-MS2 spectra we generated in step 1 with DIA-Umpire. We will generally follow exactly the same protocol as in Tutorial 1 for library generation from DDA files. Like this you can repeat and further extend on what you have already learned. We don't explain all parameters in detail again in this tutorial, for references please look into Tutorial 1 in case you need a recap.

- First, we perform a database search for the three mzXML files by using the comet parameter file we generated in Tutorial 1:

```
comet -P/c/DIA_Course/Tutorial1_Library/parameter_files/comet.params.high-
high_TTOF \
collinsb_*Q*.mzXML \
&>> Tutorial3_log.txt
```

- **Note!** We run all comet searches in one command by using the asterisk (`collinsb_*Q*.mzXML`). This means that all files (Q1, Q2 and Q3 from both inputs) are processed after each other by comet.

- If you remember from the first tutorial, the next step is the scoring of PSMs with PeptideProphet:

```
xinteract -dreverse_ \
-OARPd \
-Ninteract_Q1.comet.pep.xml \
collinsb_*Q1*.pep.xml \
&>> Tutorial3_log.txt

xinteract -dreverse_ \
-OARPd \
-Ninteract_Q2.comet.pep.xml \
collinsb_*Q2*.pep.xml \
&>> Tutorial3_log.txt

xinteract -dreverse_ \
-OARPd \
-Ninteract_Q3.comet.pep.xml \
collinsb_*Q3*.pep.xml \
&>> Tutorial3_log.txt
```

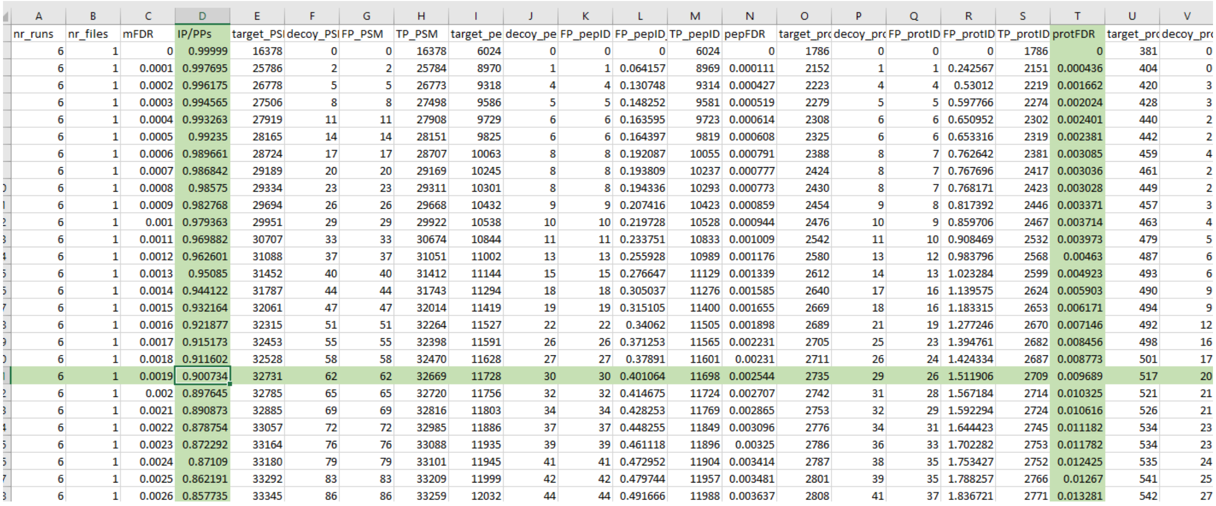
- Followed by combination and rescoring of all PSMs within iProphet:

```
InterProphetParser DECOY=reverse_ \
interact*.pep.xml \
iProphet.pep.xml \
&>> Tutorial3_log.txt
```

- Next, we perform a MAYU analysis to subsequently select an iProphet probability threshold that corresponds to a 1% global protein FDR.

```
perl /c/TPP/bin/Mayu.pl \
-A iProphet.pep.xml \
-C
/c/DIA_Course/Data/napedro_3mixed_human_yeast_ecoli_20140403_iRT_reverse
.fasta \
-E reverse_ \
-G 0.01 \
-H 101 \
-I 0 \
&>> Tutorial3_log.txt
```

- o **Note!** At this step, you have to perform the manual inspection of the MAYU output file in order to select an iProphet probability score that corresponds to a global protein FDR of 1%.
  - Open the file ending with "_main_1.07.csv" in Excel
  - Identify the column with the name "protFDR"
  - Go down until you reach the row with the last value that is smaller 0.01
  - Mark the row and find the column "IP/PP", which is the iProphet probability score
    - o Write down the number in your marked row – this is the score cutoff you use for the further analysis (result should be in the range of: 0.7-0.8 in our example)



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
nr_runs	nr_files	mFDR	IP/PPs	target_PSI	decoy_PSI	FP_PSM	TP_PSM	target_pe	decoy_pe	FP_pepID	FP_pepID_	TP_pepID	pepFDR	target_pr	decoy_pr	FP_protID	FP_protID	TP_protID	protFDR	target_pr	decoy_pr
6	1	0	0.99999	16378	0	0	16378	6024	0	0	0	6024	0	1786	0	0	0	1786	0	381	0
6	1	0.0001	0.997695	25786	2	2	25784	8970	1	1	0.064157	8969	0.000111	2152	1	1	0.242567	2151	0.000436	404	0
6	1	0.0002	0.996175	26778	5	5	26773	9318	4	4	0.130748	9314	0.000427	2223	4	4	0.53012	2219	0.001662	420	3
6	1	0.0003	0.994565	27506	8	8	27498	9586	5	5	0.148252	9581	0.000519	2279	5	5	0.597766	2274	0.002024	428	3
6	1	0.0004	0.993263	27919	11	11	27908	9729	6	6	0.163595	9723	0.000614	2308	6	6	0.650952	2302	0.002401	440	2
6	1	0.0005	0.99235	28165	14	14	28151	9825	6	6	0.164397	9819	0.000608	2325	6	6	0.653316	2319	0.002381	442	2
6	1	0.0006	0.989661	28724	17	17	28707	10063	8	8	0.192087	10055	0.000791	2388	8	7	0.762642	2381	0.003085	459	4
6	1	0.0007	0.986842	29189	20	20	29169	10245	8	8	0.193809	10237	0.000777	2424	8	7	0.767696	2417	0.003036	461	2
6	1	0.0008	0.98575	29334	23	23	29311	10301	8	8	0.194336	10293	0.000773	2430	8	7	0.768171	2423	0.003028	449	2
6	1	0.0009	0.982768	29694	26	26	29668	10432	9	9	0.207416	10423	0.000859	2454	9	8	0.817392	2446	0.003371	457	3
6	1	0.001	0.979363	29951	29	29	29922	10538	10	10	0.219728	10528	0.000944	2476	10	9	0.859706	2467	0.003714	463	4
6	1	0.0011	0.969882	30707	33	33	30674	10844	11	11	0.233751	10833	0.001009	2542	11	10	0.908469	2532	0.003973	479	5
6	1	0.0012	0.962601	31088	37	37	31051	11002	13	13	0.255928	10989	0.001176	2580	13	12	0.983796	2568	0.00463	487	6
6	1	0.0013	0.95085	31452	40	40	31412	11144	15	15	0.276647	11129	0.001339	2612	14	13	1.023284	2599	0.004923	493	6
6	1	0.0014	0.944122	31787	44	44	31743	11294	18	18	0.305037	11276	0.001585	2640	17	16	1.139575	2624	0.005903	490	9
6	1	0.0015	0.932164	32061	47	47	32014	11419	19	19	0.315105	11400	0.001655	2669	18	16	1.183315	2653	0.006171	494	9
6	1	0.0016	0.921877	32315	51	51	32264	11527	22	22	0.34062	11505	0.001898	2689	21	19	1.277246	2670	0.007146	492	12
6	1	0.0017	0.915173	32453	55	55	32398	11591	26	26	0.371253	11565	0.002231	2705	25	23	1.394761	2682	0.008456	498	16
6	1	0.0018	0.911602	32528	58	58	32470	11628	27	27	0.37891	11601	0.00231	2711	26	24	1.424334	2687	0.008773	501	17
6	1	0.0019	0.900734	32731	62	62	32669	11728	30	30	0.401064	11698	0.002544	2735	29	26	1.511906	2709	0.009689	517	20
6	1	0.002	0.897645	32785	65	65	32720	11756	32	32	0.414675	11724	0.002707	2742	31	28	1.567184	2714	0.010325	521	21
6	1	0.0021	0.890873	32885	69	69	32816	11803	34	34	0.428253	11769	0.002865	2753	32	29	1.592294	2724	0.010616	526	21
6	1	0.0022	0.878754	33057	72	72	32985	11886	37	37	0.448255	11849	0.003096	2776	34	31	1.644423	2745	0.011182	534	23
6	1	0.0023	0.872292	33164	76	76	33088	11935	39	39	0.461118	11896	0.00325	2786	36	33	1.702282	2753	0.011782	534	23
6	1	0.0024	0.87109	33180	79	79	33101	11945	41	41	0.472952	11904	0.003414	2787	38	35	1.753427	2752	0.012425	535	24
6	1	0.0025	0.862191	33292	83	83	33209	11999	42	42	0.479744	11957	0.003481	2801	39	35	1.788257	2766	0.01267	541	25
6	1	0.0026	0.857735	33345	86	86	33259	12032	44	44	0.491666	11988	0.003637	2808	41	37	1.836721	2771	0.013281	542	27

- Now it's time to generate a spectral library using SpectraST. First, spectra are filtered and retention time alignment is performed
- **Note!** Remember to **change the acquisition setting to –cIHCD for the QE data**

```
spectrast -cNSpecLib \
-cICID-QTOF \
-cf "Protein! ~ reverse_" \
-cP0.900734 \
-c_IRT/data/Data/irtkit.txt \
-c_IRR iProphet.pep.xml \
&>> Tutorial3_log.txt
```

- Afterwards, consensus spectra are generated for each peptide ion:
- **Note!** Remember to **change the acquisition setting to –cIHCD for the QE data**

```
spectrast -cNSpecLib_cons \
-cICID-QTOF \
-cAC SpecLib.splib \
&>> Tutorial3_log.txt
```

- Generate a SpectraST MRM transition list:
- **Note!** Remember to **change the acquisition setting to –cIHCD for the QE data**

```
spectrast -cNSpecLib_pqp \
-cICID-QTOF \
-cM \
SpecLib_cons.splib \
&>> Tutorial3_log.txt
```

- Convert the SpectraST MRM to TraML

```
TargetedFileConverter \
-in SpecLib_pqp.mrm \
-out transitionlist.TraML \
&>> Tutorial3_log.txt
```

- Generate target assays by copying this command in your script and run it:
- **Note**: **When you use the QE data, adjust the windows file accordingly**.

```
OpenSwathAssayGenerator \
-in transitionlist.TraML \
-out transitionlist_optimized.TraML \
-swath_windows_file /c/DIA_Course/Data/swath64_w_header.txt \
&>> Tutorial3_log.txt
```

In order to enable decoy-based error-rate control with pyProphet downstream of OpenSWATH, it is important to append the assay library with decoy-query parameters.

- Append decoy transitions to the spectral library:

```
OpenSwathDecoyGenerator \
-in transitionlist_optimized.TraML \
-out transitionlist_optimized_decoys.TraML \
-method shuffle \
&>> Tutorial3_log.txt
```

- Convert the library to the pqp format for the further OpenSWATH analysis:

```
TargetedFileConverter \
-in transitionlist_optimized_decoys.TraML \
-out transitionlist_optimized_decoys.pqp \
&>> Tutorial3_log.txt
```

Now you have a final library that can be used for targeted, peptide-centric DIA analysis with OpenSWATH ☺.

Finally, to inspect the library, we will convert it back to the tsv format and take a closer look.

- Run TargetedFileConverter

```
TargetedFileConverter \
-in transitionlist_optimized.TraML \
-out transitionlist_optimized.tsv \
&>> Tutorial3_log.txt
```

You no got through the entire library generation workflow and have a script that you can use as a reference and scaffold for the library generation of other datasets ☺. Save the shell script and exit RStudio.

- In Excel, count the number of proteins and peptides for each organism as explained in Tutorial 1.
  - How do the libraries compare?
  - Can you identify a specific difference between the two libraries?

We would like to thank SystemsX for supporting the Zurich DIA / SWATH Course 2018.

SystemsX.ch
The Swiss Initiative in Systems Biology