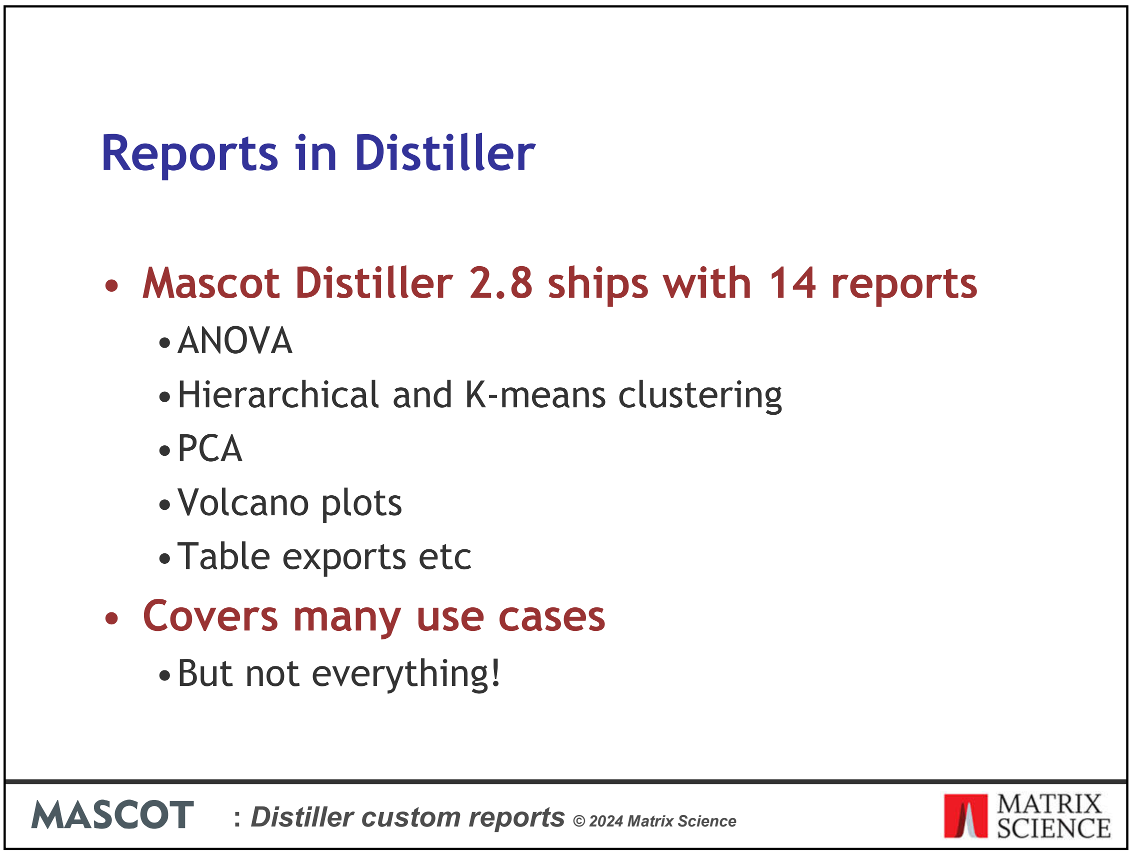
Custom reporting with Mascot Distiller

Patrick Emery

Matrix Science Ltd

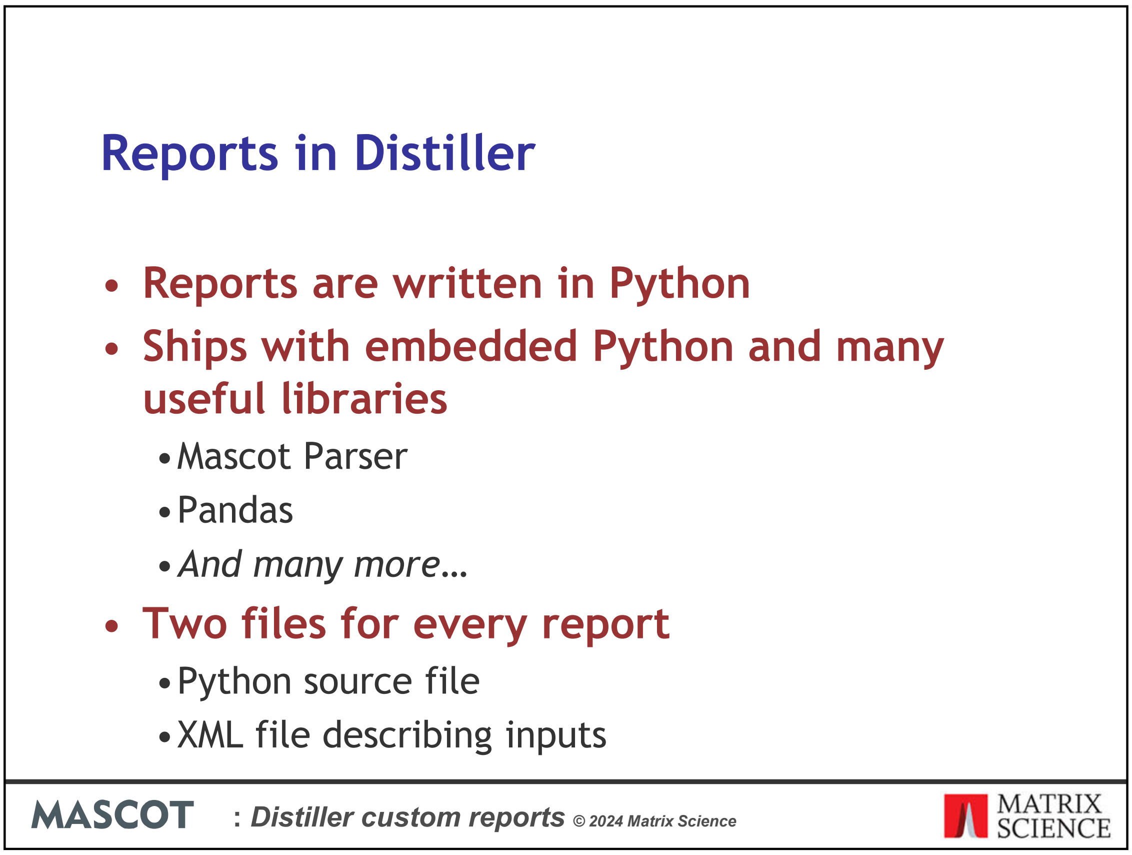


Reports in Distiller

- Mascot Distiller 2.8 ships with 14 reports
  - ANOVA
  - Hierarchical and K-means clustering
  - PCA
  - Volcano plots
  - Table exports etc
- Covers many use cases
  - But not everything!

Mascot Distiller 2.8 ships with 14 different reports for analysing and exporting your quantitation results. These include reports such as Analysis of Variants (ANOVA), clustering reports such a hierarchical, K-means clustering and Principal Component Analysis. Standard reports such as the Volcano plot, box-plot are available and there are table exports which make it easy to get your data out of Distiller and into 3rd party packages such as Excel, R or Perseus from the Max Planck institute.

These cover many use cases, but we can't cover every possible analysis type ourselves and you may have some specific requirement that calls for a custom report.



Reports in Distiller

- Reports are written in Python
- Ships with embedded Python and many useful libraries
  - Mascot Parser
  - Pandas
  - *And many more…*
- Two files for every report
  - Python source file
  - XML file describing inputs

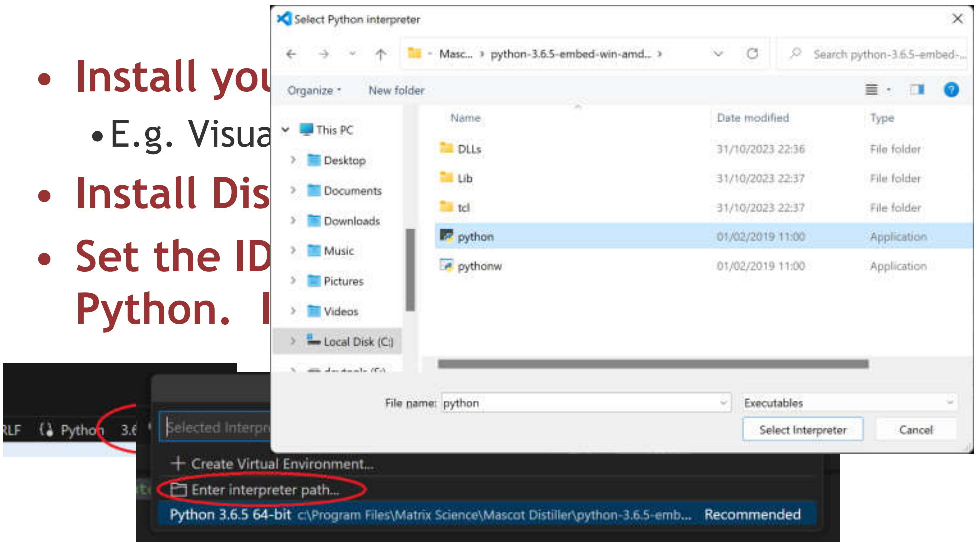
In older versions of Distiller, reports were written in XSLT – an XML transformation language. This was powerful, but not commonly used and without access to many of the features you might need when creating a report.

In Mascot Distiller 2.8, reports are instead written in Python. Distiller ships with it's own embedded copy of Python with many useful libraries for data analysis.

Each report is comprised of two files – the Python report source file, and an XML file describing the report inputs and defining any Wizard to be displayed in the Distiller GUI.

Getting started

- Install you
  - E.g. Visua
- Install Dis
- Set the ID
  Python. I



Select Python interpreter
Masc... › python-3.6.5-embed-win-amd... ›
Search python-3.6.5-embed-...
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Select Interpreter
Cancel
RLF
Python
3.6
Selected Interpr
Create Virtual Environment...
Enter interpreter path...

Name	Date modified	Type
DLLs	31/10/2023 22:36	File folder
Lib	31/10/2023 22:37	File folder
tcl	31/10/2023 22:37	File folder
python	01/02/2019 11:00	Application
pythonw	01/02/2019 11:00	Application

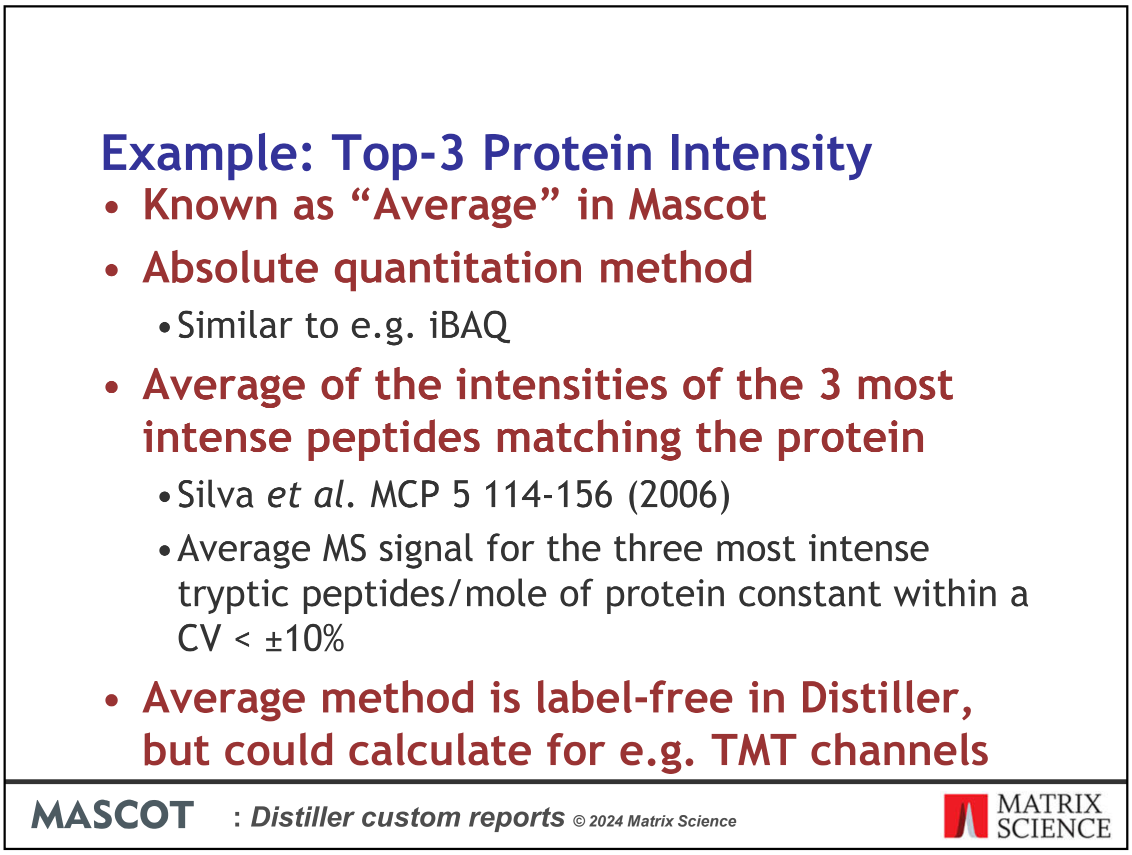
File name: python   Executables

Python 3.6.5 64-bit c:\Program Files\Matrix Science\Mascot Distiller\python-3.6.5-emb... Recommended

With a bit of knowledge of Python and the Mascot Parser library therefore you can write your own reports for use in Distiller. For this, you'll need an Integrated Development Environment (IDE) that supports Python, such as the free Visual Studio Code from Microsoft.

Install Distiller on your development PC. Without a licence, Distiller will run in viewer mode, but this will still install the embedded copy of Python and the various 3rd party libraries. Once you've done that, tell your IDE to use the version of Python installed with Distiller.

Here's an example of how to do that in Visual Studio Code



Example: Top-3 Protein Intensity

- Known as “Average” in Mascot
- Absolute quantitation method
  - Similar to e.g. iBAQ
- Average of the intensities of the 3 most intense peptides matching the protein
  - Silva *et al.* MCP 5 114-156 (2006)
  - Average MS signal for the three most intense tryptic peptides/mole of protein constant within a CV < ±10%
- Average method is label-free in Distiller, but could calculate for e.g. TMT channels

As an example of the steps involved in, and the utility of, creating a custom report, we’ll write a report to calculate absolute protein intensity using the top-3 method.

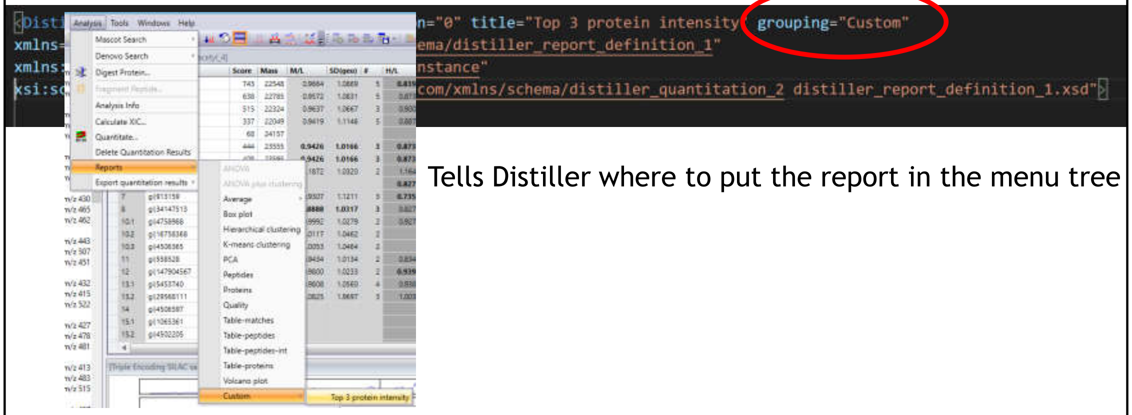
This is known as “Average” label free quantitation in Distiller and is an absolute quantitation method similar to iBAQ

The protein intensity is calculated from the average of the intensities of the 3 most intense peptides matching a protein. The observation from the original paper was that the average MS signal for the three most intense tryptic peptide/mole of protein is constant within Coefficient of Variation of +/- 10%

Average quantitation is label-free in Distiller, but there’s no reason you couldn’t calculate it for other quantitation techniques.

Example: Top-3 Protein Intensity

- XML file defines parameters and inputs for the report Wizard

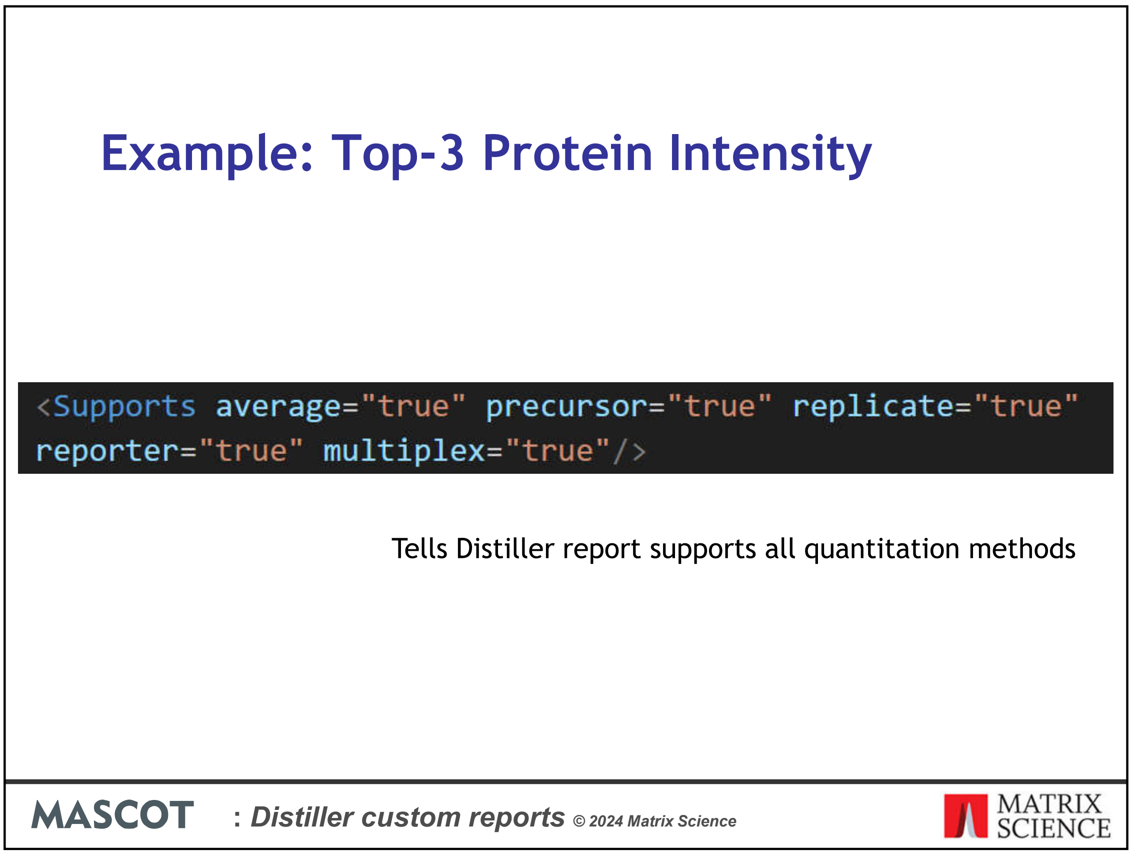


```
<Disti                                    n="0" title="Top 3 protein intensity" grouping="Custom"
xmlns=                                    ema/distiller_report_definition_1"
xmlns:                                    nstance"
xsi:sc                                    com/xmlns/schema/distiller_quantitation_2 distiller_report_definition_1.xsd">
```

Tells Distiller where to put the report in the menu tree

Once we've setup our IDE, we can start work on the report itself. As mentioned earlier, a report in Mascot Distiller is comprised of two files – the actual Python .py source file for the report and an XML file which defines the report inputs and Wizard for the Distiller GUI.

Starting with the XML file, here, we're telling Distiller where to put the report on the menu tree – in this case in a subfolder called Custom

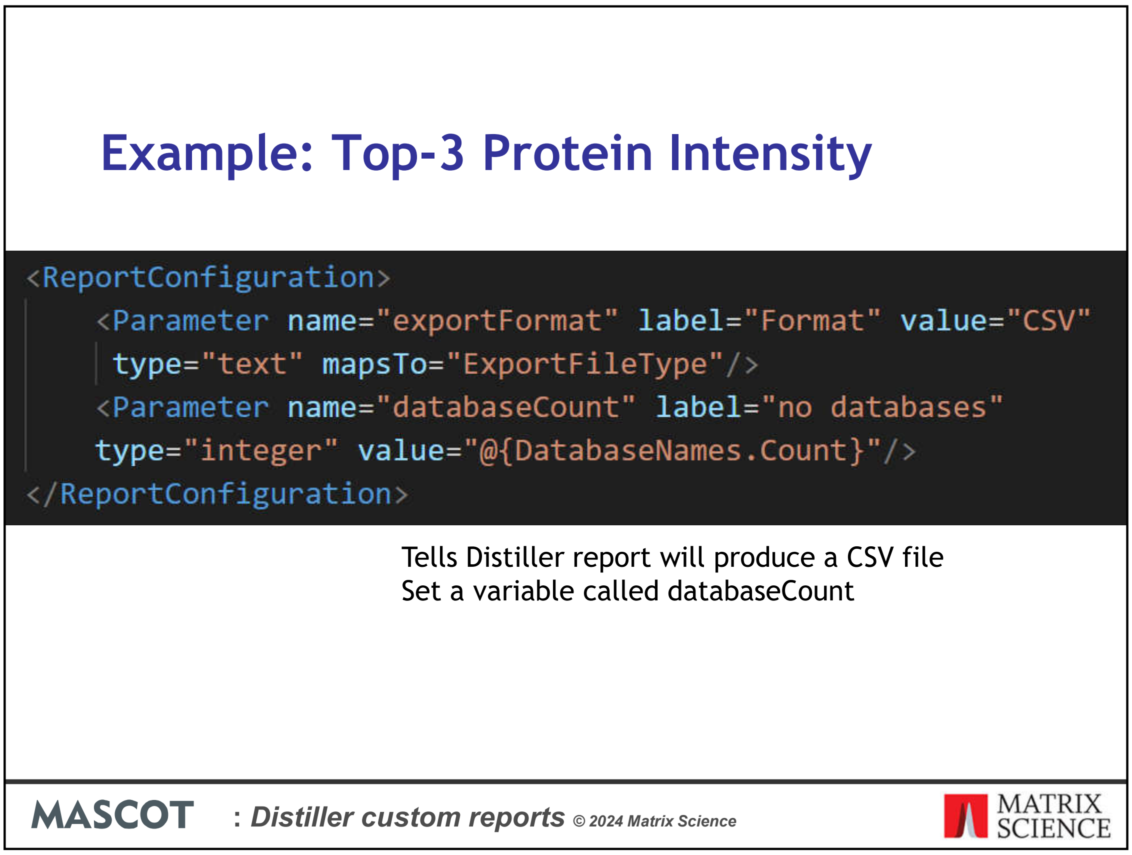


Example: Top-3 Protein Intensity

```
<Supports average="true" precursor="true" replicate="true"
reporter="true" multiplex="true"/>
```

Tells Distiller report supports all quantitation methods

And here we're telling Distiller that our report will support all quantitation methods.



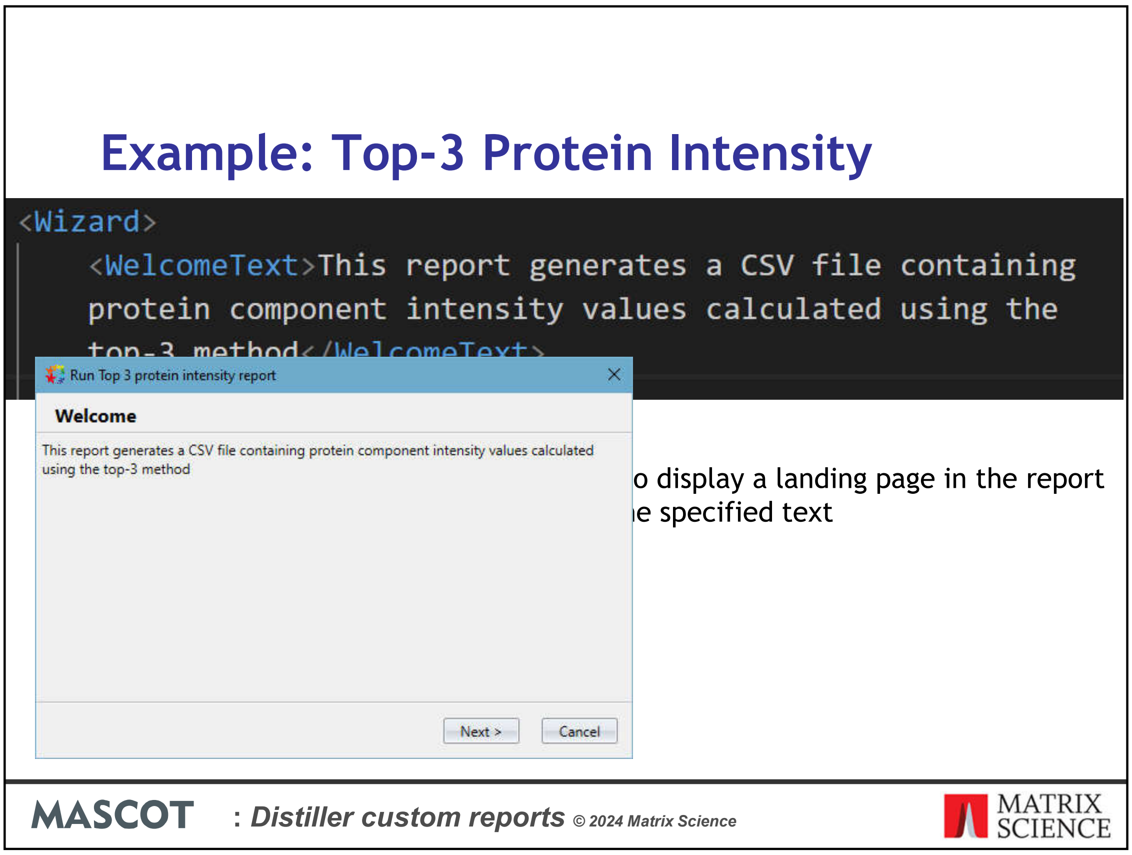
Example: Top-3 Protein Intensity

```
<ReportConfiguration>
    <Parameter name="exportFormat" label="Format" value="CSV"
     type="text" mapsTo="ExportFileType"/>
    <Parameter name="databaseCount" label="no databases"
    type="integer" value="@{DatabaseNames.Count}"/>
</ReportConfiguration>
```

Tells Distiller report will produce a CSV file
Set a variable called databaseCount

MASCOT : *Distiller custom reports* © 2024 Matrix Science MATRIX SCIENCE

This is telling Distiller it will produce a CSV file and sets the number of sequence databases searched to a variable called databaseCount



Example: Top-3 Protein Intensity

```
<Wizard>
    <WelcomeText>This report generates a CSV file containing
    protein component intensity values calculated using the
    top-3 method</WelcomeText>
```

Run Top 3 protein intensity report

Welcome

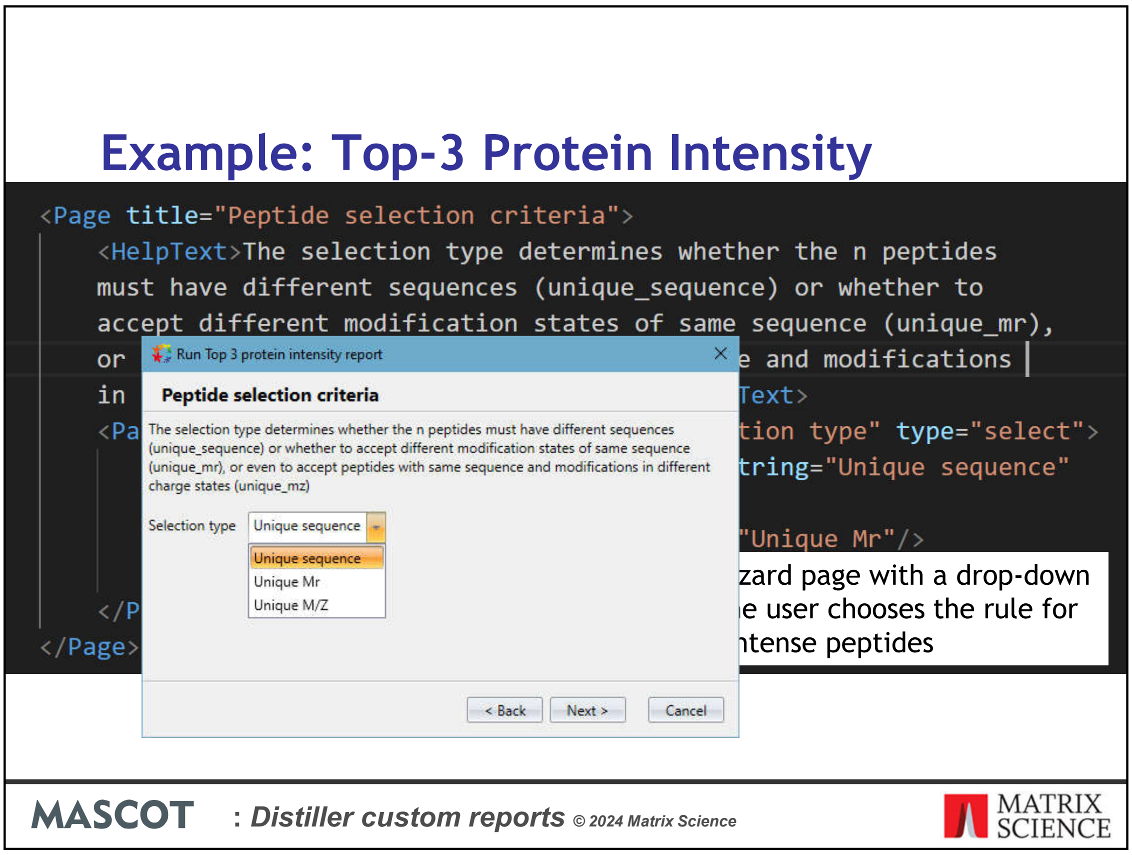
This report generates a CSV file containing protein component intensity values calculated using the top-3 method

o display a landing page in the report
ie specified text

MASCOT : *Distiller custom reports* © 2024 Matrix Science MATRIX SCIENCE

A separate section in the XML can then be used to define the Wizard to display in the GUI. This can be used to capture input and settings information from the user before running the report.

Here we're defining the landing page for the Wizard – this should define some welcome text to explain to the user running the report what it is going to calculate.



Example: Top-3 Protein Intensity

```
<Page title="Peptide selection criteria">
    <HelpText>The selection type determines whether the n peptides
    must have different sequences (unique_sequence) or whether to
    accept different modification states of same sequence (unique_mr),
    or                                                e and modifications
    in                                                Text>
    <Pa                                               tion type" type="select">
                                                      tring="Unique sequence"

                                                      "Unique Mr"/>

    </P
</Page>
```

zard page with a drop-down
e user chooses the rule for
ntense peptides

Run Top 3 protein intensity report

Peptide selection criteria

The selection type determines whether the n peptides must have different sequences (unique_sequence) or whether to accept different modification states of same sequence (unique_mr), or even to accept peptides with same sequence and modifications in different charge states (unique_mz)

Selection type: Unique sequence

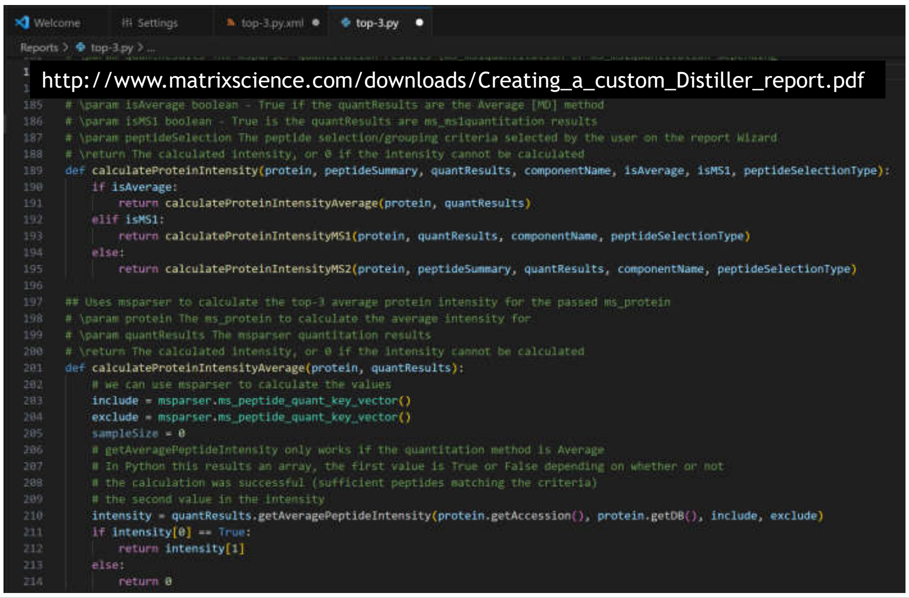
MASCOT : *Distiller custom reports* © 2024 Matrix Science MATRIX SCIENCE

Here we're defining an actual input page with a parameter. In this case, we're defining a drop-down selection control where the end user will set the criteria for selecting the top-3 most intense peptides.

The default is Unique Sequence – where different charge and modification states of the same peptide are treated as a single match. Selecting Unique Mr would mean different modification states are counted as separate peptide matches, while Unique m/z means the different charge and modification states of the same peptide sequence are treated as different matches.

Here's the Wizard page defined by that XML displayed in Distiller. We can define as many input pages as required to capture the inputs for our report. Entered values are then output to the Python script as comma separate values.

Example: Top-3 Protein Intensity



Welcome | Settings | top-3.py.xml | top-3.py

Reports > top-3.py > ...

http://www.matrixscience.com/downloads/Creating_a_custom_Distiller_report.pdf

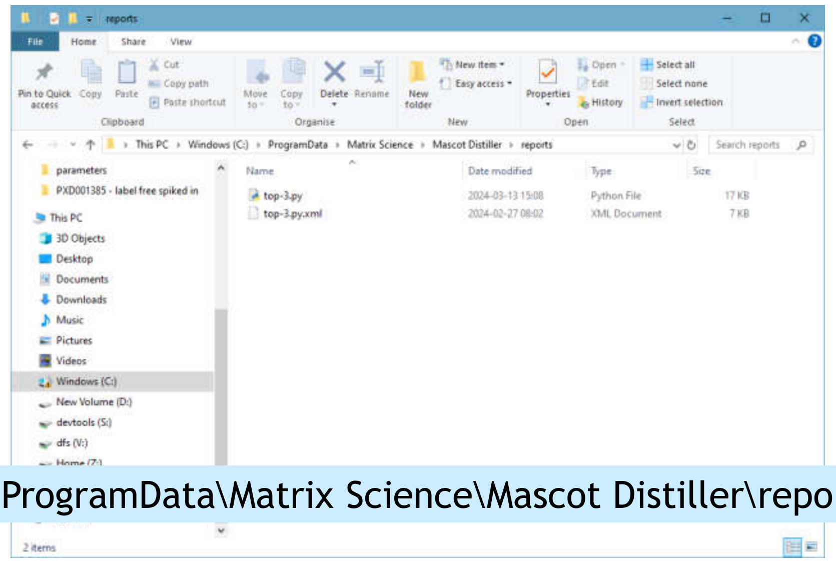
```python
# \param isAverage boolean - True if the quantResults are the Average [MD] method
# \param isMS1 boolean - True is the quantResults are ms_ms1quantitation results
# \param peptideSelection The peptide selection/grouping criteria selected by the user on the report Wizard
# \return The calculated intensity, or 0 if the intensity cannot be calculated
def calculateProteinIntensity(protein, peptideSummary, quantResults, componentName, isAverage, isMS1, peptideSelectionType):
    if isAverage:
        return calculateProteinIntensityAverage(protein, quantResults)
    elif isMS1:
        return calculateProteinIntensityMS1(protein, quantResults, componentName, peptideSelectionType)
    else:
        return calculateProteinIntensityMS2(protein, peptideSummary, quantResults, componentName, peptideSelectionType)

## Uses msparser to calculate the top-3 average protein intensity for the passed ms_protein
# \param protein The ms_protein to calculate the average intensity for
# \param quantResults The msparser quantitation results
# \return The calculated intensity, or 0 if the intensity cannot be calculated
def calculateProteinIntensityAverage(protein, quantResults):
    # we can use msparser to calculate the values
    include = msparser.ms_peptide_quant_key_vector()
    exclude = msparser.ms_peptide_quant_key_vector()
    sampleSize = 0
    # getAveragePeptideIntensity only works if the quantitation method is Average
    # In Python this results an array, the first value is True or False depending on whether or not
    # the calculation was successful (sufficient peptides matching the criteria)
    # the second value in the intensity
    intensity = quantResults.getAveragePeptideIntensity(protein.getAccession(), protein.getDB(), include, exclude)
    if intensity[0] == True:
        return intensity[1]
    else:
        return 0
```

Once we've defined our inputs in the XML, we need to create our Python report file. This is a standard Python script – we're using Mascot Parser to access the search results, and also some useful script libraries we've provided to help with loading and formatting the data. You'll need to be familiar with the Python 3 programming language to create your own reports, and also spend a bit of time getting to know Mascot Parser and any other libraries you want to use.

Here we're looking at a snippet of the code involved in calculating the top-3 protein intensity - a slightly different method being required depending on the quantitation method used in Mascot and Distiller.

It's a bit beyond the scope of a presentation like this to run through all of this, but you can find a detailed tutorial pdf on our public website here:

Example: Top-3 Protein Intensity



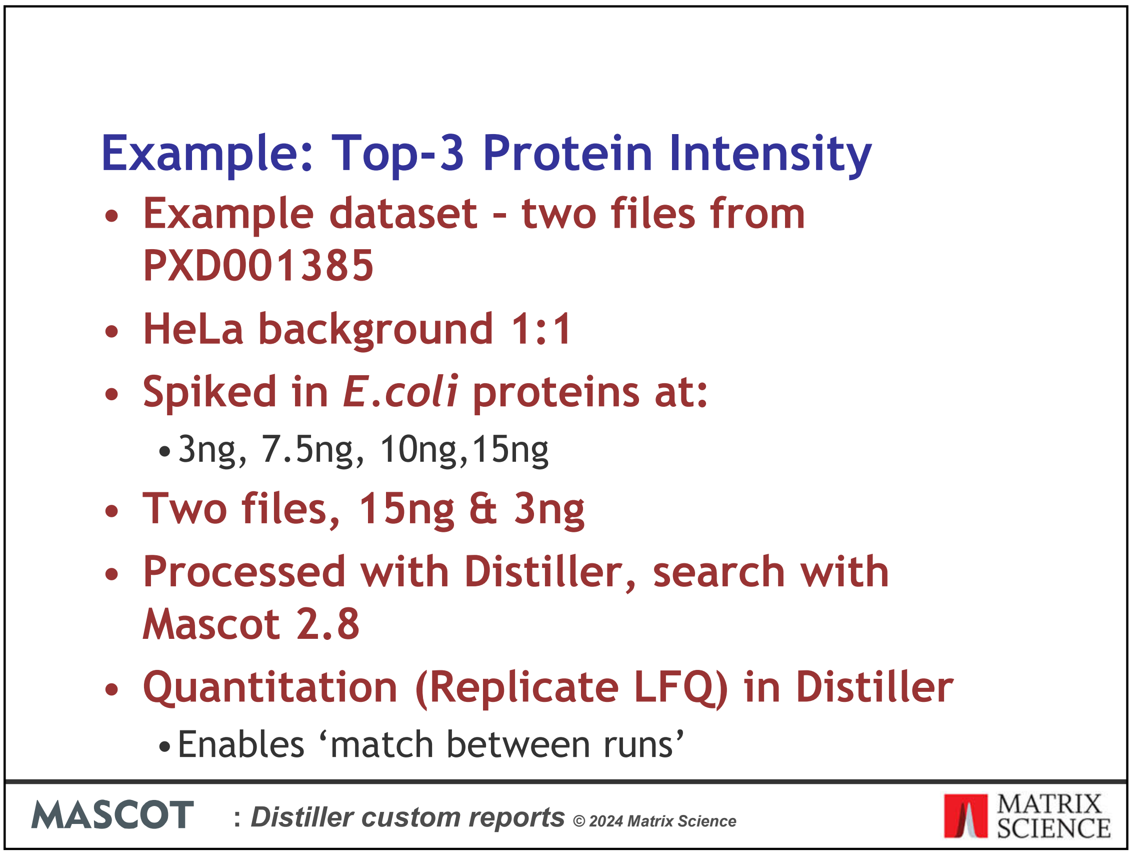
Name	Date modified	Type	Size
top-3.py	2024-03-13 15:08	Python File	17 KB
top-3.py.xml	2024-02-27 08:02	XML Document	7 KB

C:\ProgramData\Matrix Science\Mascot Distiller\reports

The name of your XML file should be the same as the Python .py report script with .XML appended to the end. When both files are written, copy them into the c:\ProgramData\Matrix Science\Mascot Distiller\reports directory and restart Mascot Distiller – your new report should now appear in the Analysis->Reports menu in Distiller in the subfolder that you've specified in the XML.

Note, the ProgramData folder is hidden in Windows by default, but if you type the path into the navigation bar in Explorer, it will open.

Once you've copied the files into place and restarted Distiller, you can update the Python file without restarting Distiller, but changes to the XML file may require Distiller to be restarted in order to take effect.

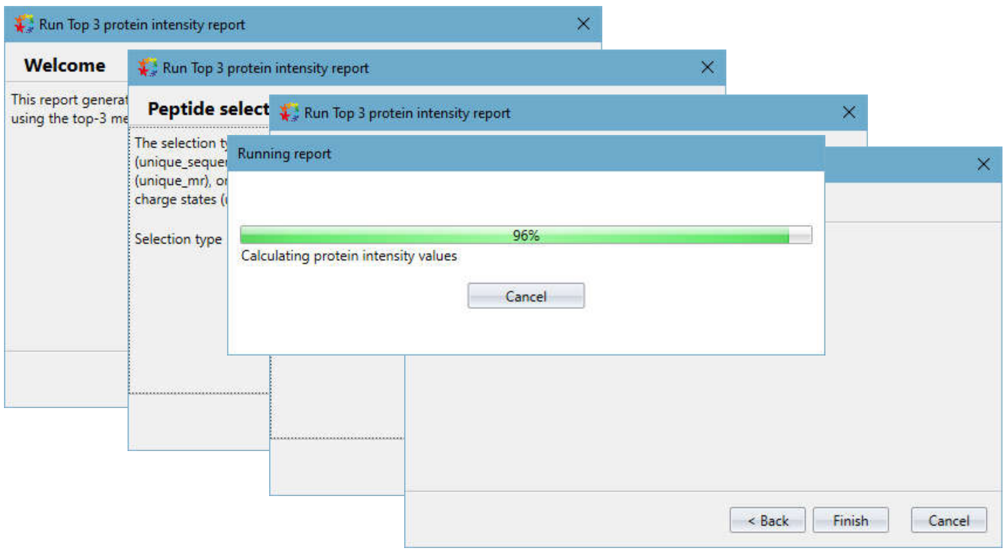


Example: Top-3 Protein Intensity

- Example dataset – two files from PXD001385
- HeLa background 1:1
- Spiked in *E.coli* proteins at:
  - 3ng, 7.5ng, 10ng,15ng
- Two files, 15ng & 3ng
- Processed with Distiller, search with Mascot 2.8
- Quantitation (Replicate LFQ) in Distiller
  - Enables 'match between runs'

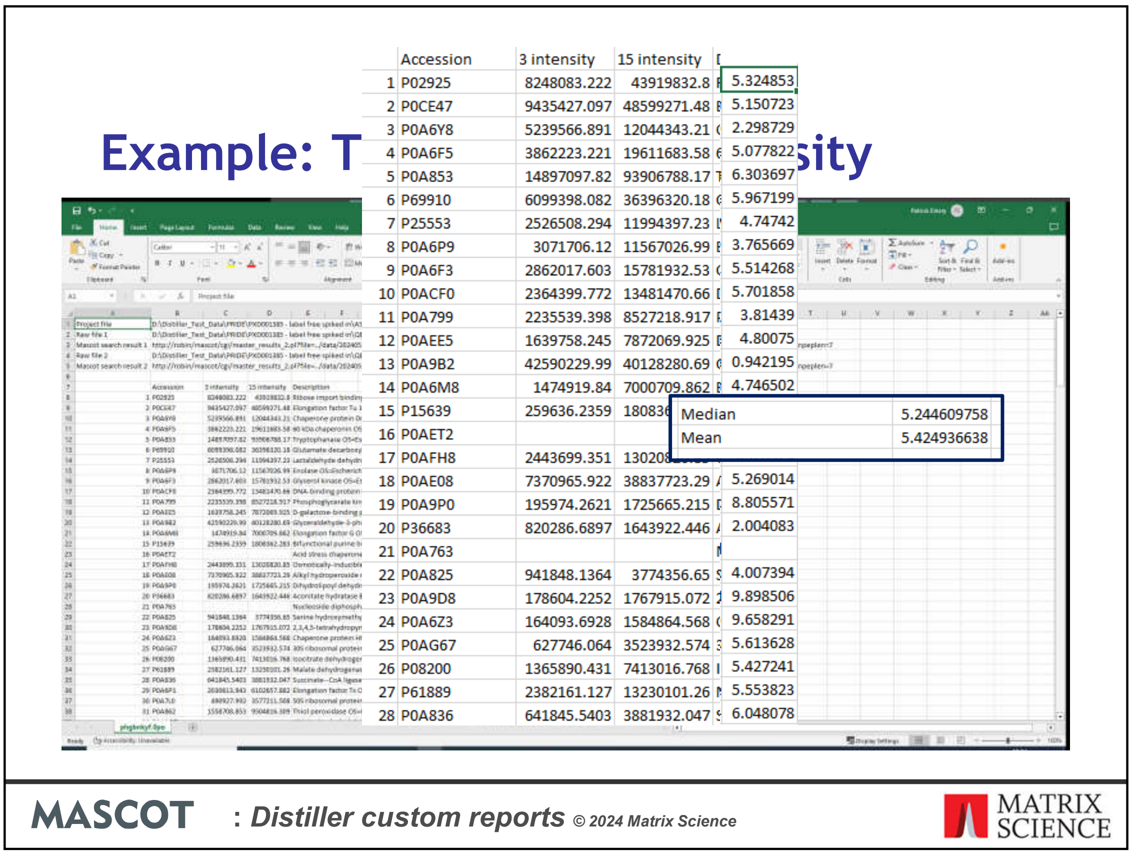
Let's use our new report on an example dataset. We've taken two files from the following dataset in the PRIDE repository. This is a benchmarking dataset, where we have a background of human proteins from a HeLa extract at 1:1, and then a series of samples where E.coli protein extracts were spiked in at 3ng, 7.5ng, 10ng and 15ng. We took two of the files, one of the 15ng and the matched 3ng file, processed them with Mascot Distiler 2.8 and search against the Uniprot Ecoli proteome with Mascot 2.8, before running quantitation in Distiller using the "Replicate" LFQ method (which enabled 'match between runs' type behaviour)

Example: Top-3 Protein Intensity



MASCOT : *Distiller custom reports* © 2024 Matrix Science MATRIX SCIENCE

Here's our complete Wizard, as defined by the report XML file. When we're ready, click 'Finish' on the final page and Distiller will run the report. When it completes, the generated CSV file will be automatically opened in our spreadsheet application (Excel in this case)



Example: T sity

	Accession	3 intensity	15 intensity	
1	P02925	8248083.222	43919832.8	5.324853
2	P0CE47	9435427.097	48599271.48	5.150723
3	P0A6Y8	5239566.891	12044343.21	2.298729
4	P0A6F5	3862223.221	19611683.58	5.077822
5	P0A853	14897097.82	93906788.17	6.303697
6	P69910	6099398.082	36396320.18	5.967199
7	P25553	2526508.294	11994397.23	4.74742
8	P0A6P9	3071706.12	11567026.99	3.765669
9	P0A6F3	2862017.603	15781932.53	5.514268
10	P0ACF0	2364399.772	13481470.66	5.701858
11	P0A799	2235539.398	8527218.917	3.81439
12	P0AEE5	1639758.245	7872069.925	4.80075
13	P0A9B2	42590229.99	40128280.69	0.942195
14	P0A6M8	1474919.84	7000709.862	4.746502
15	P15639	259636.2359	180836	
16	P0AET2			
17	P0AFH8	2443699.351	130208	
18	P0AE08	7370965.922	38837723.29	5.269014
19	P0A9P0	195974.2621	1725665.215	8.805571
20	P36683	820286.6897	1643922.446	2.004083
21	P0A763			
22	P0A825	941848.1364	3774356.65	4.007394
23	P0A9D8	178604.2252	1767915.072	9.898506
24	P0A6Z3	164093.6928	1584864.568	9.658291
25	P0AG67	627746.064	3523932.574	5.613628
26	P08200	1365890.431	7413016.768	5.427241
27	P61889	2382161.127	13230101.26	5.553823
28	P0A836	641845.5403	3881932.047	6.048078

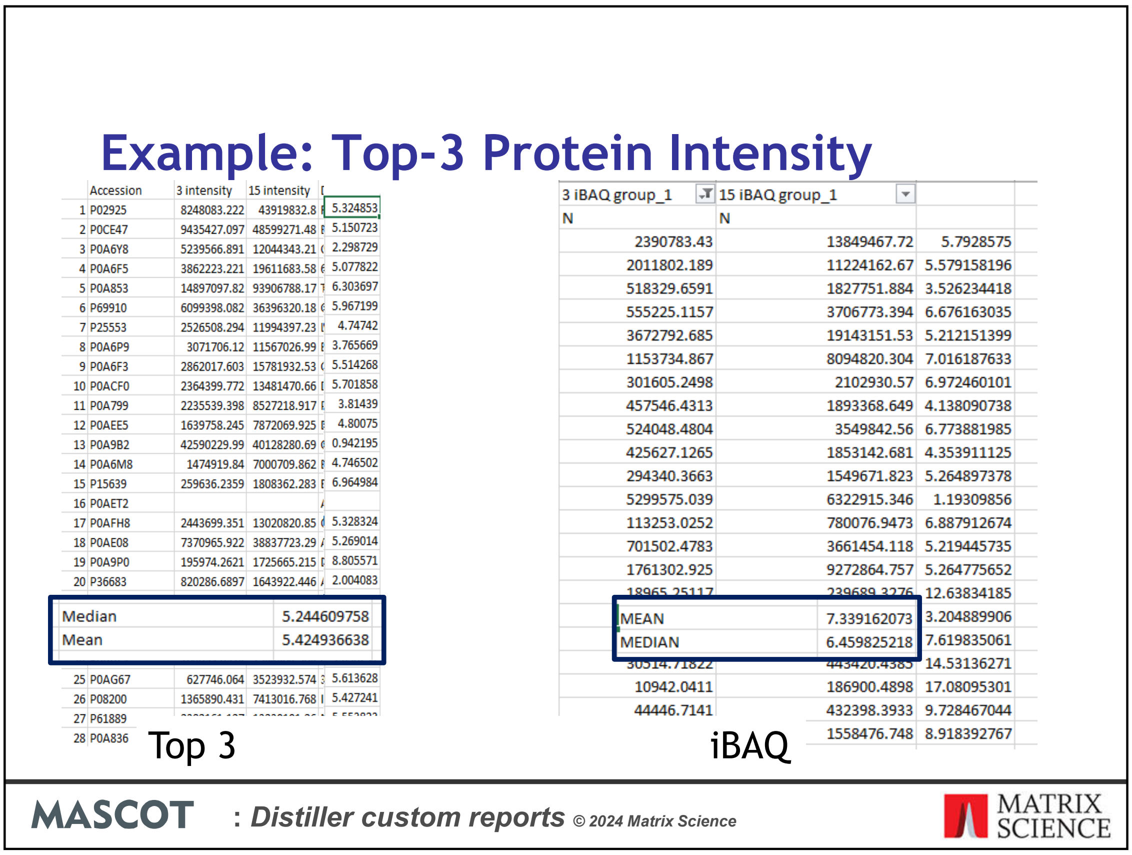
Median	5.244609758
Mean	5.424936638

MASCOT : *Distiller custom reports* © 2024 Matrix Science

MATRIX SCIENCE

And here's our generated report – at the top we have some basic header information about the raw files and search results. Below that is our table of results, with calculated intensity values for the 3ng and 15ng samples. Within Excel you could calculate relative abouts of each protein against any control protein of known amount, or to the total intensity of all proteins etc. In this case, we have values for one of the 3ng Ecoli and 15ng Ecoli samples – so the protein intensities for the 15ng sample should be 5x greater than the intensities from the 3ng sample – obviously, I could have had the report calculate the ratio, but it's very easy to do in Excel.

So, there's a bit of spread (as you'd expect), but overall, those look like pretty good numbers – as confirmed by the Median and Mean ratios for the dataset.



Example: Top-3 Protein Intensity

	Accession	3 intensity	15 intensity	
1	P02925	8248083.222	43919832.8	5.324853
2	P0CE47	9435427.097	48599271.48	5.150723
3	P0A6Y8	5239566.891	12044343.21	2.298729
4	P0A6F5	3862223.221	19611683.58	5.077822
5	P0A853	14897097.82	93906788.17	6.303697
6	P69910	6099398.082	36396320.18	5.967199
7	P25553	2526508.294	11994397.23	4.74742
8	P0A6P9	3071706.12	11567026.99	3.765669
9	P0A6F3	2862017.603	15781932.53	5.514268
10	P0ACF0	2364399.772	13481470.66	5.701858
11	P0A799	2235539.398	8527218.917	3.81439
12	P0AEE5	1639758.245	7872069.925	4.80075
13	P0A9B2	42590229.99	40128280.69	0.942195
14	P0A6M8	1474919.84	7000709.862	4.746502
15	P15639	259636.2359	1808362.283	6.964984
16	P0AET2			
17	P0AFH8	2443699.351	13020820.85	5.328324
18	P0AE08	7370965.922	38837723.29	5.269014
19	P0A9P0	195974.2621	1725665.215	8.805571
20	P36683	820286.6897	1643922.446	2.004083

Median	5.244609758
Mean	5.424936638

25	P0AG67	627746.064	3523932.574	5.613628
26	P08200	1365890.431	7413016.768	5.427241
27	P61889	[illegible]	[illegible]	[illegible]
28	P0A836			

Top 3

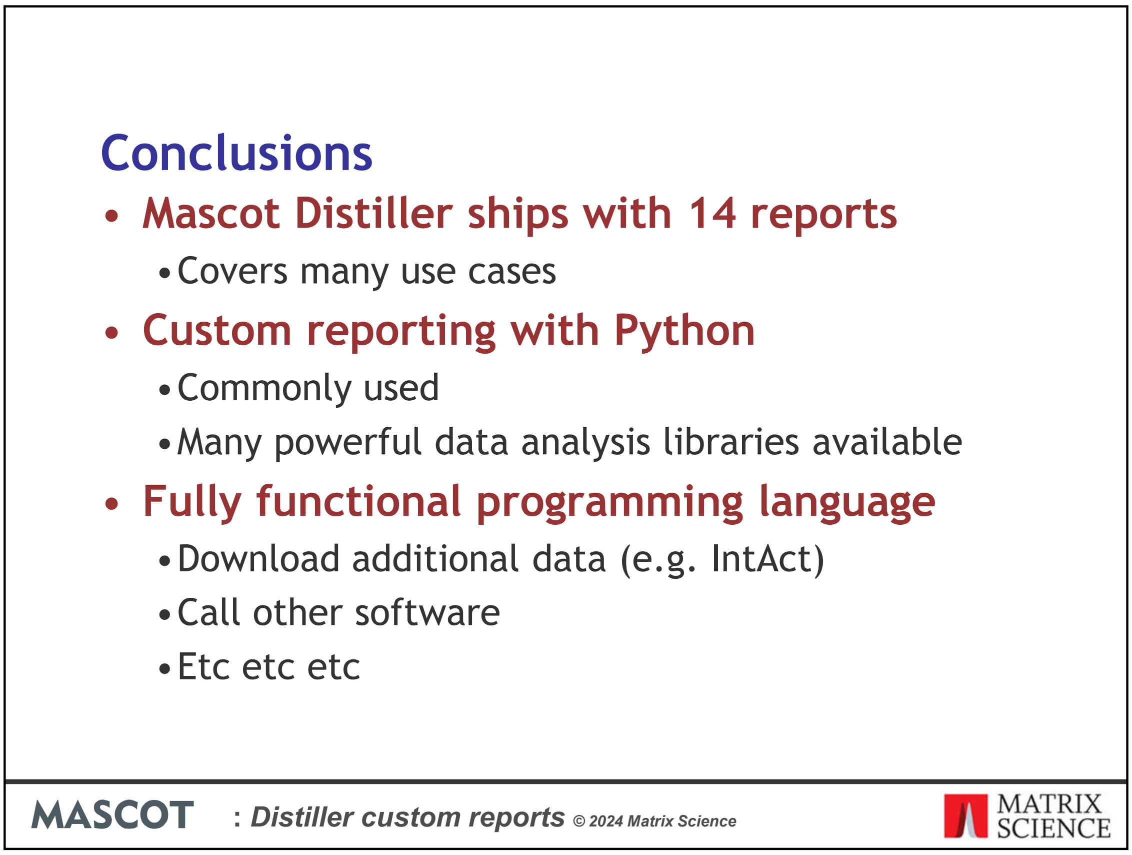
3 iBAQ group_1	15 iBAQ group_1	
N	N	
2390783.43	13849467.72	5.7928575
2011802.189	11224162.67	5.579158196
518329.6591	1827751.884	3.526234418
555225.1157	3706773.394	6.676163035
3672792.685	19143151.53	5.212151399
1153734.867	8094820.304	7.016187633
301605.2498	2102930.57	6.972460101
457546.4313	1893368.649	4.138090738
524048.4804	3549842.56	6.773881985
425627.1265	1853142.681	4.353911125
294340.3663	1549671.823	5.264897378
5299575.039	6322915.346	1.19309856
113253.0252	780076.9473	6.887912674
701502.4783	3661454.118	5.219445735
1761302.925	9272864.757	5.264775652
18965.25117	239689.3276	12.63834185
		3.204889906
		7.619835061
30514.71822	443420.4385	14.53136271
10942.0411	186900.4898	17.08095301
44446.7141	432398.3933	9.728467044
	1558476.748	8.918392767

MEAN	7.339162073
MEDIAN	6.459825218

iBAQ

How does that compare to another technique such as iBAQ? Well, looks like the ratio of the iBAQ values for these proteins aren't bad, but they're actually a bit too high, as shown by the Mean and Median ratios. So, in this case Top-3 has out performed iBAQ and is a simpler calculation to boot. Unlike iBAQ you don't need the protein sequence – you could calculate it yourself very easily in a spreadsheet with just the protein's peptide intensity values.

While top-3 has done better in this instance, that isn't always the case of course. In general both techniques perform similarly, and generally surprisingly well. The main disadvantage of top-3 compared with iBAQ is the requirement for 3 distinct peptide matches to calculate the average intensity from – iBAQ allows you to calculate intensities for proteins with fewer distinct matches – which may be why top-3 out performs iBAQ here; we're only using the most intense peptides in the calculation.



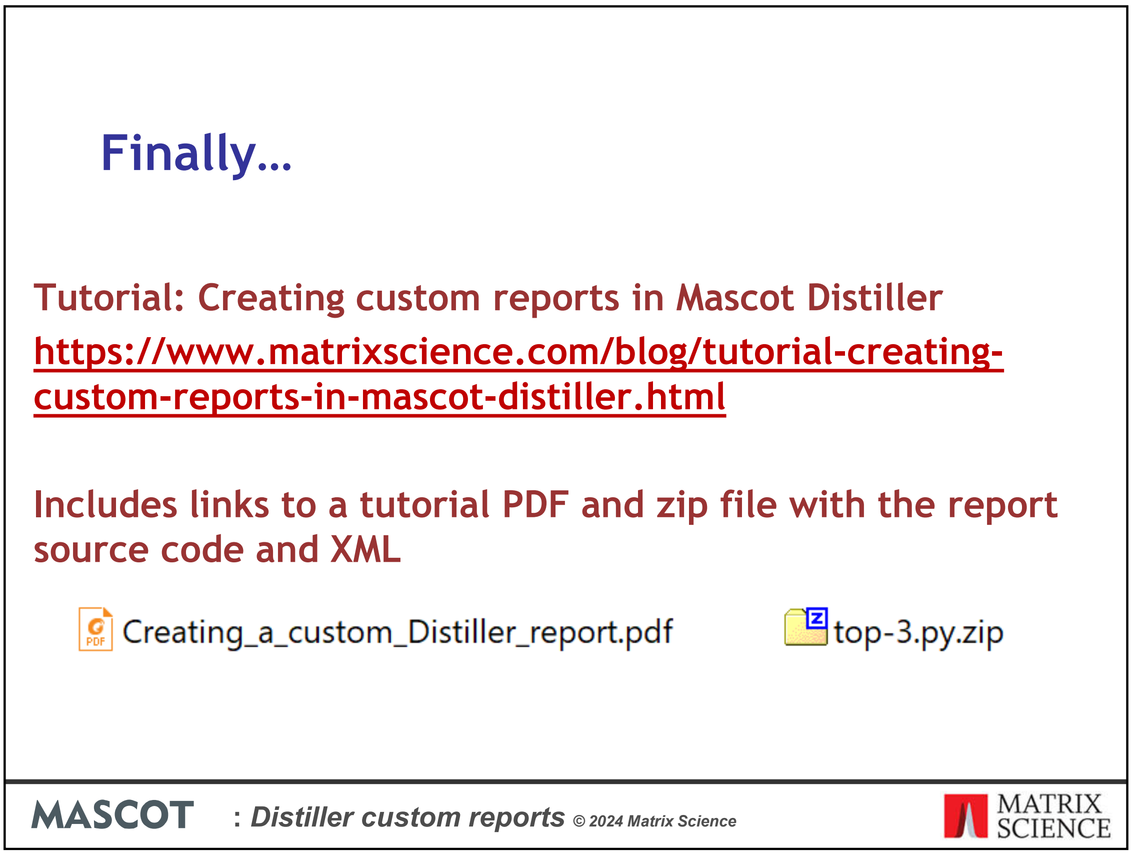
Conclusions

- Mascot Distiller ships with 14 reports
  - Covers many use cases
- Custom reporting with Python
  - Commonly used
  - Many powerful data analysis libraries available
- Fully functional programming language
  - Download additional data (e.g. IntAct)
  - Call other software
  - Etc etc etc

Mascot Distiller 2.8 ships with 14 standard reports. These cover many use cases, but they can't cover every type of analysis you could ever want to carry out.

Reports are written in Python – this is a commonly used programming language for data analysis and it has many powerful libraries available to help with that.

Unlike in earlier versions of Distiller, we're using a fully function programming language for reporting. That means you could get very creative with your reports – for example, you could download protein interaction data from the EBI's IntAct database and incorporate that into a report, or you could call another tool, carry out some downstream analysis and then incorporate that into a report.



Finally…

Tutorial: Creating custom reports in Mascot Distiller

https://www.matrixscience.com/blog/tutorial-creating-custom-reports-in-mascot-distiller.html

Includes links to a tutorial PDF and zip file with the report source code and XML

Creating_a_custom_Distiller_report.pdf

top-3.py.zip

MASCOT : *Distiller custom reports* © 2024 Matrix Science

MATRIX SCIENCE

Finally, you can find a blog article about this here

A detailed tutorial pdf about creating the top-3 report here

And the Python source code and XML file for the report can be downloaded from here