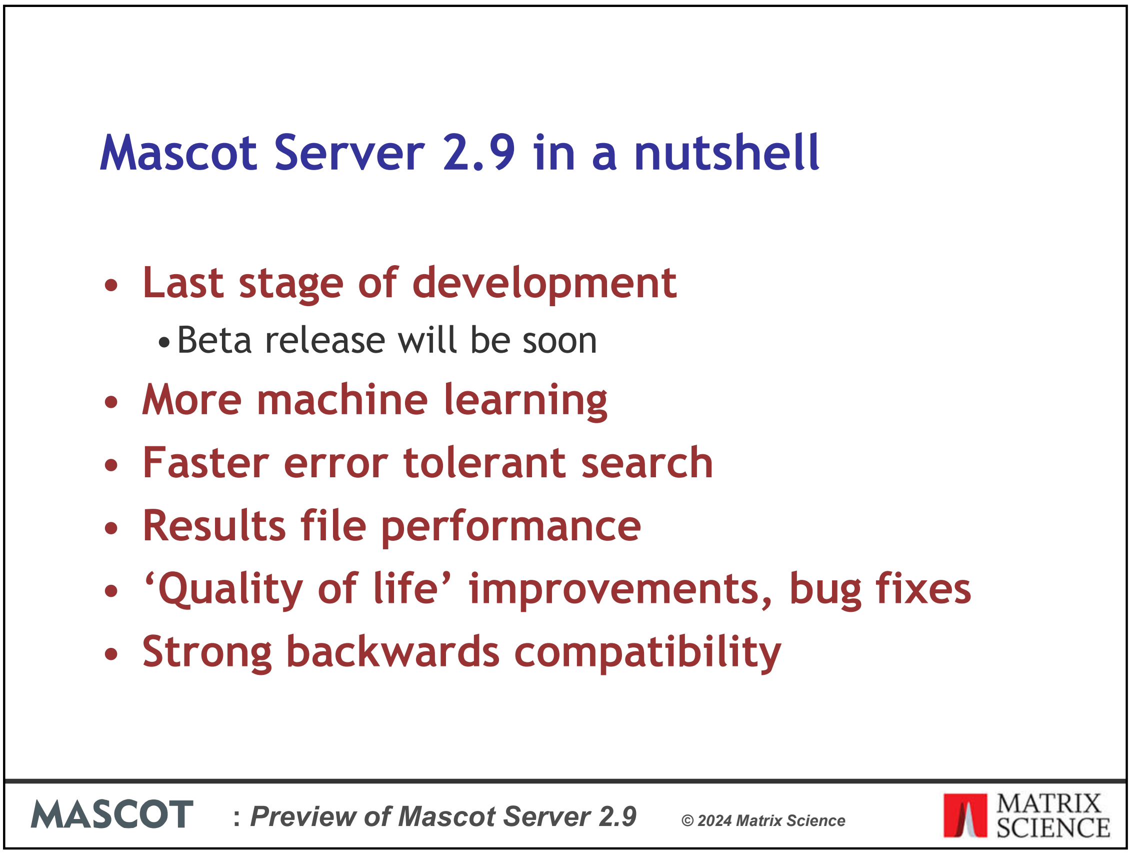
Preview of Mascot server 2.9



Mascot Server 2.9 in a nutshell

- Last stage of development
  - Beta release will be soon
- More machine learning
- Faster error tolerant search
- Results file performance
- ‘Quality of life’ improvements, bug fixes
- Strong backwards compatibility

We’re currently in the last stage of the development cycle for the next release, working towards the first beta release of Mascot Server 2.9. In this talk, I’ll walk you through the major new features and show a few screenshots from the alpha version.

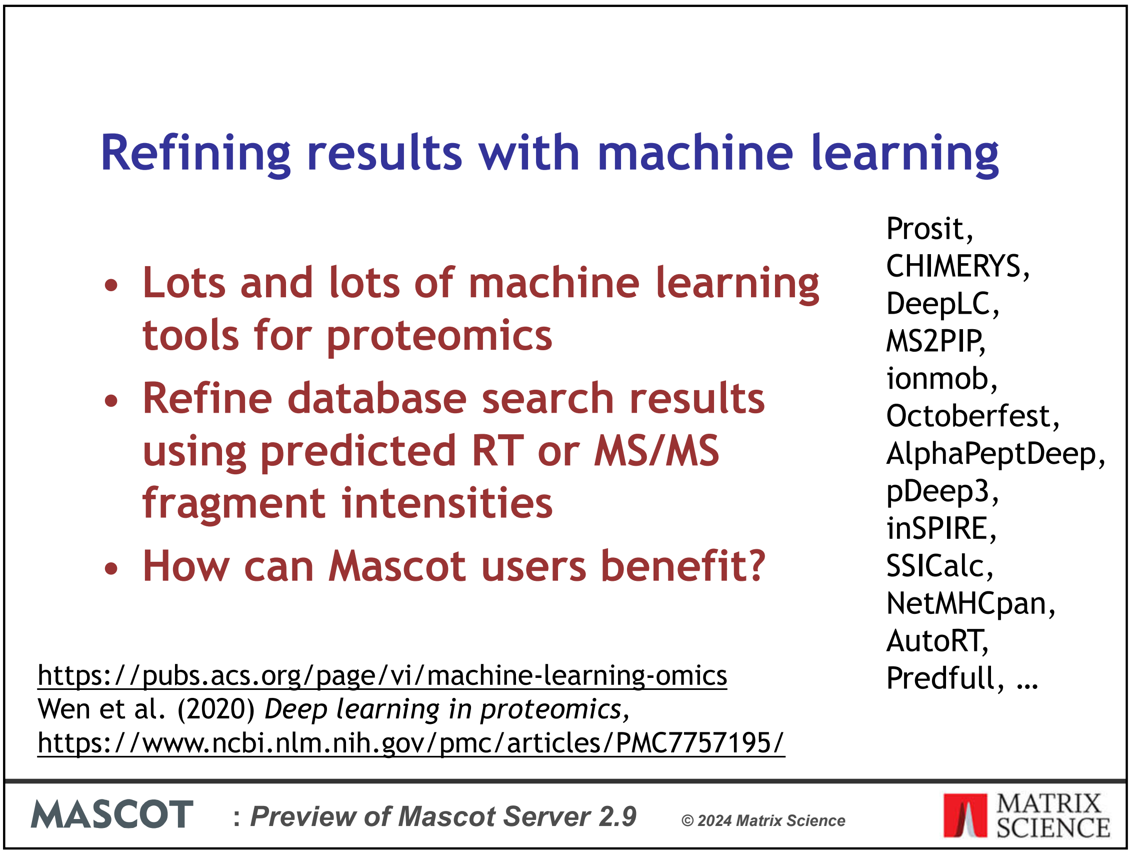
Everyone is talking about machine learning and artificial intelligence, both inside and outside proteomics. Mascot has included some machine learning for a long time: we were among the early adopters of Percolator. In the next version, we’re doubling down and adding more ways to use the latest advances in machine learning. I’ll explain shortly what exactly we mean by this.

We’ve also added new controls for the error tolerant search. The error tolerant search is a two-pass search, where the search engine identifies unsuspected modifications using all the modifications in Unimod. You can now restrict the second pass to a subset of Unimod, which makes the search a lot faster.

We are making internal changes to improve results file performance.

Like previous versions, we are making small ‘quality of life’ improvements, and there are always bug fixes. I don’t have time in this presentation to show a full list.

I’d like to stress that Mascot retains strong backwards compatibility with earlier versions. Mascot Server 2.9 continues the tradition.



Refining results with machine learning

- Lots and lots of machine learning tools for proteomics
- Refine database search results using predicted RT or MS/MS fragment intensities
- How can Mascot users benefit?

Prosit,
CHIMERYS,
DeepLC,
MS2PIP,
ionmob,
Octoberfest,
AlphaPeptDeep,
pDeep3,
inSPIRE,
SSICalc,
NetMHCpan,
AutoRT,
Predfull, …

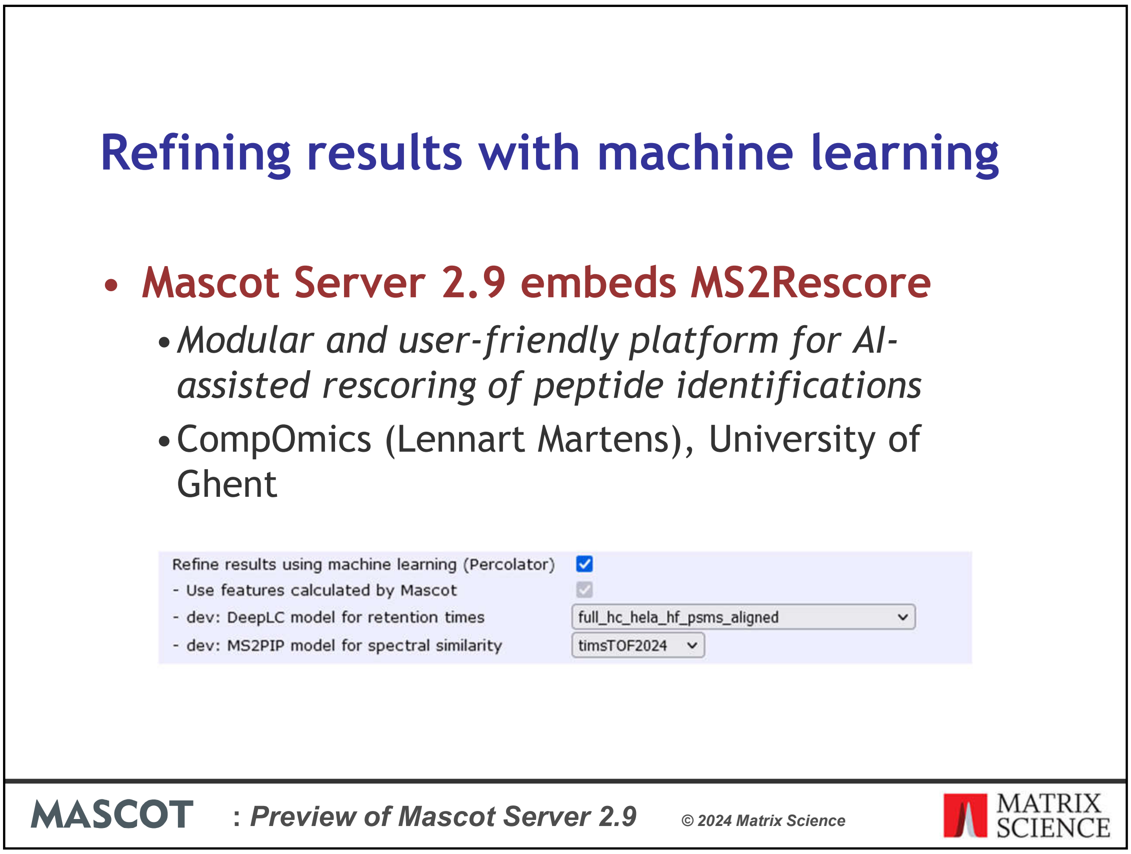
https://pubs.acs.org/page/vi/machine-learning-omics
Wen et al. (2020) *Deep learning in proteomics*,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7757195/

MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

There are lots and lots of machine learning tools for proteomics. New ones seem to be introduced every month, and when I chat with experts, even they say it's hard to keep up with all the new things. On the right of this slide are just a dozen I've heard of. Possibly the most familiar one is Prosit? There are more listed in the ACS Virtual Issue on machine learning as well as the 2020 article Deep learning in proteomics.

The goal of all these tools is to refine database search results so you get better sensitivity and specificity. Basically that means getting more peptide matches at the same or better false discovery rate. For example, Prosit predicts the fragment intensities of b and y fragment ions, given the peptide sequence and variable mods as input. The predicted spectrum can be correlated with the observed spectrum, and the correlation will be higher for correct peptide matches and lower for incorrect peptide matches. This can be used for separating the correct database matches from the incorrect.

The main motivation in the next Mascot version is: How can Mascot users benefit from this? It's actually surprisingly difficult even for experienced software developers to get started with these tools, as you need some expertise to install, configure and run them.



Refining results with machine learning

- Mascot Server 2.9 embeds MS2Rescore
  - *Modular and user-friendly platform for AI-assisted rescoring of peptide identifications*
  - CompOmics (Lennart Martens), University of Ghent

Refine results using machine learning (Percolator) ☑
- Use features calculated by Mascot ☑
- dev: DeepLC model for retention times full_hc_hela_hf_psms_aligned
- dev: MS2PIP model for spectral similarity timsTOF2024

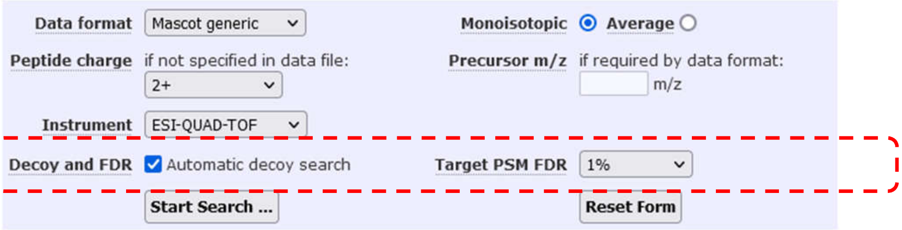
MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

Well, we don’t want to reinvent the wheel and we want to make things easy for you. We will start by embedding MS2Rescore in Mascot. This is a modular and user-friendly platform for AI-assisted rescoring of peptide identifications, developed by Lennart Martens’s group at the University of Ghent. MS2Rescore is a front-end for three prediction tools. There’s DeepLC for modelling retention times; MS2PIP for modelling spectral similarity (like Prosit); and ionmob for modelling ion mobility.

Our goal in Mascot is that you should be able to click a button to make your results better. To do that, we have integrated DeeplC and MS2PIP model selection directly in the format controls of the Protein Family Summary report. You tick the box to refine results using machine learning, then select a DeepLC model that is similar to your LC and an MS2PIP model similar to your mass spec. Mascot will take care of the rest.

Refining results with machine learning

- **Example data set:**
  - HeLa digest with *E. coli*
  - timsTOF, 2h elution, DDA, 238k queries
  - Trypsin, standard modifications
- **Run an automatic target-decoy search**



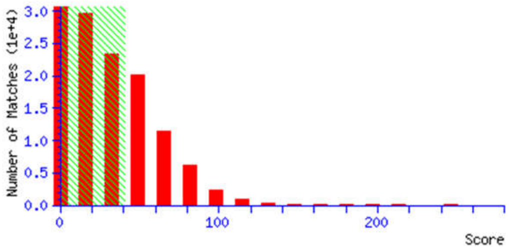
Data format: Mascot generic
Monoisotopic ● Average ○
Peptide charge: if not specified in data file: 2+
Precursor m/z: if required by data format: m/z
Instrument: ESI-QUAD-TOF
Decoy and FDR: ☑ Automatic decoy search
Target PSM FDR: 1%

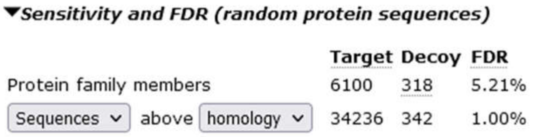
Here's a quick demonstration how it works. I'm using a timsTOF data set as an example. This is a 2h elution of a DDA run of a HeLa digest with *E. coli* spiked in. It's digested with trypsin and using standard fixed and variable modifications – fixed carbamidomethylation and oxidation as variable mod. There are 238k queries, which is not an unusual size for timsTOF runs.

When you submit the search, just tick the box that you want to run an automatic target-decoy search. This is all you need to be able to use the MS2Rescore integration. You can optionally set the target FDR here, which we introduced in the previous version.

Refining results with machine learning

- **Before machine learning:**
  - 34k unique peptide sequences at 1% FDR
  - Many very high scoring matches
  - But no clear separation

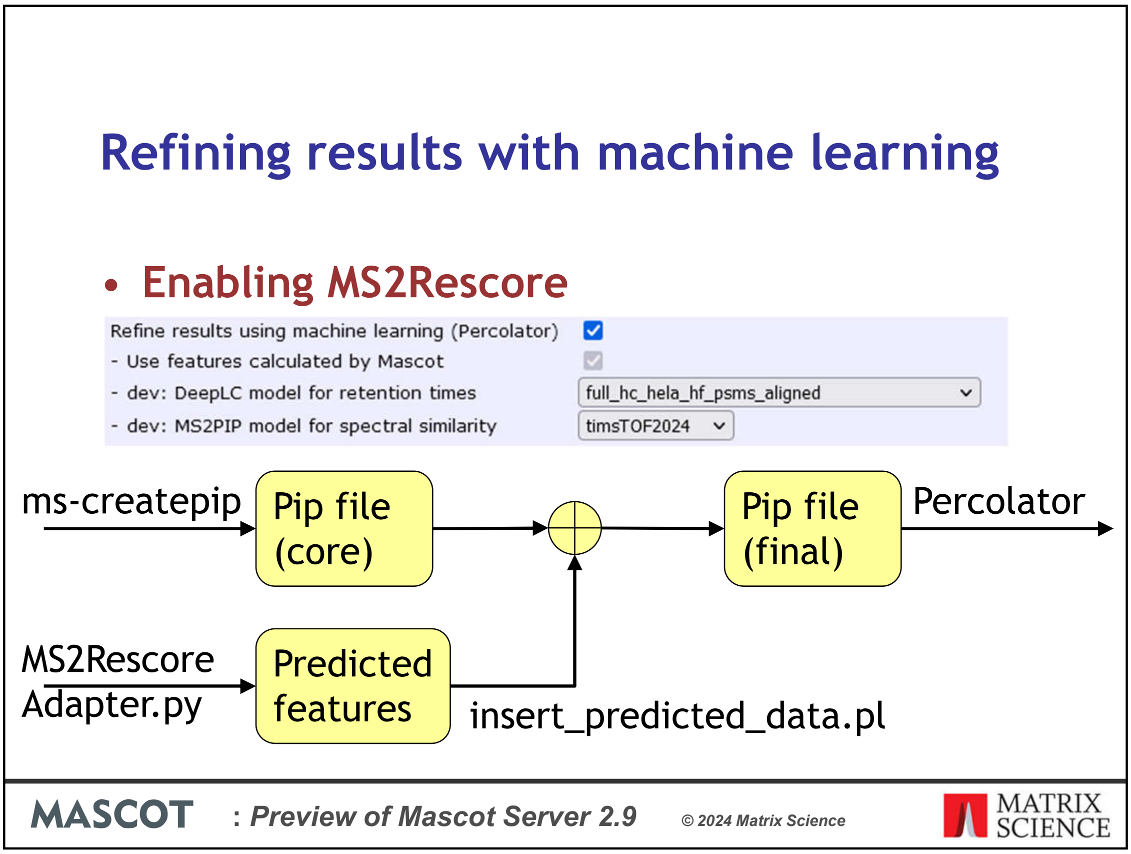




▼*Sensitivity and FDR (random protein sequences)*

	Target	Decoy	FDR
Protein family members	6100	318	5.21%
Sequences ⌄ above homology ⌄	34236	342	1.00%

When the results come back, Mascot has found matches to 34k peptide sequences in the target database at 1% FDR, which map to about 6000 protein hits. The score histogram shows there are many very high scoring matches, which is what we often see with timsTOF data. However, there doesn't seem to be a clear separation between correct and incorrect matches in the histogram. It's pretty obvious that the matches in the right tail will almost all be correct matches, but it's quite difficult to draw a line to say where the incorrect matches end and correct matches begin.



Refining results with machine learning

- **Enabling MS2Rescore**

Refine results using machine learning (Percolator) ☑
- Use features calculated by Mascot ☑
- dev: DeepLC model for retention times: full_hc_hela_hf_psms_aligned
- dev: MS2PIP model for spectral similarity: timsTOF2024

ms-createpip → Pip file (core) → ⊕ → Pip file (final) → Percolator

MS2Rescore Adapter.py → Predicted features → insert_predicted_data.pl → ⊕

MASCOT : *Preview of Mascot Server 2.9*

MATRIX SCIENCE

Let’s tick the box to refine the results using machine learning. Select a suitable DeepLC model for retention times. Here, I’ve used the HeLa model shipped with DeepLC. Then select a suitable MS2PIP model. I’ve used the timsTOF2024 model shipped with MS2PIP, which is trained on tryptic and chymotryptic peptides.

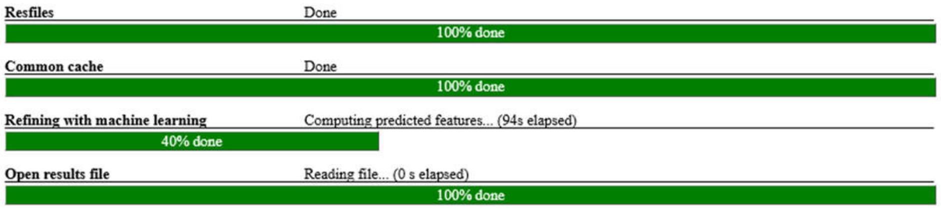
When you select these options, Mascot runs three steps. It uses a component called ms-createpip to calculate core features. The core features are things like precursor mass error, charge state, peptide length, the amount of intensity that is matched in the spectrum and so on. There are 26 core features enabled by default.

Then Mascot calls MS2RescoreAdapter, which is a Python script we developed and it will be shipped as part of Mascot. MS2Rescore has a very good Python API. The adapter calls MS2Rescore using the models you’ve selected in the format controls. It predicts the retention time and MS/MS spectrum based on the peptide sequence and variable mods, for both target and decoy matches.

Finally, a Mascot component called insert_predicted_data combines the predicted features with the core features to create a final pip file. This is sent to Percolator. The predicted features are metrics like the difference between predicted and observed retention time, and the correlation between the predicted and observed spectrum.

Refining results with machine learning

- Enable MS2Rescore



Resfiles Done

100% done

Common cache Done

100% done

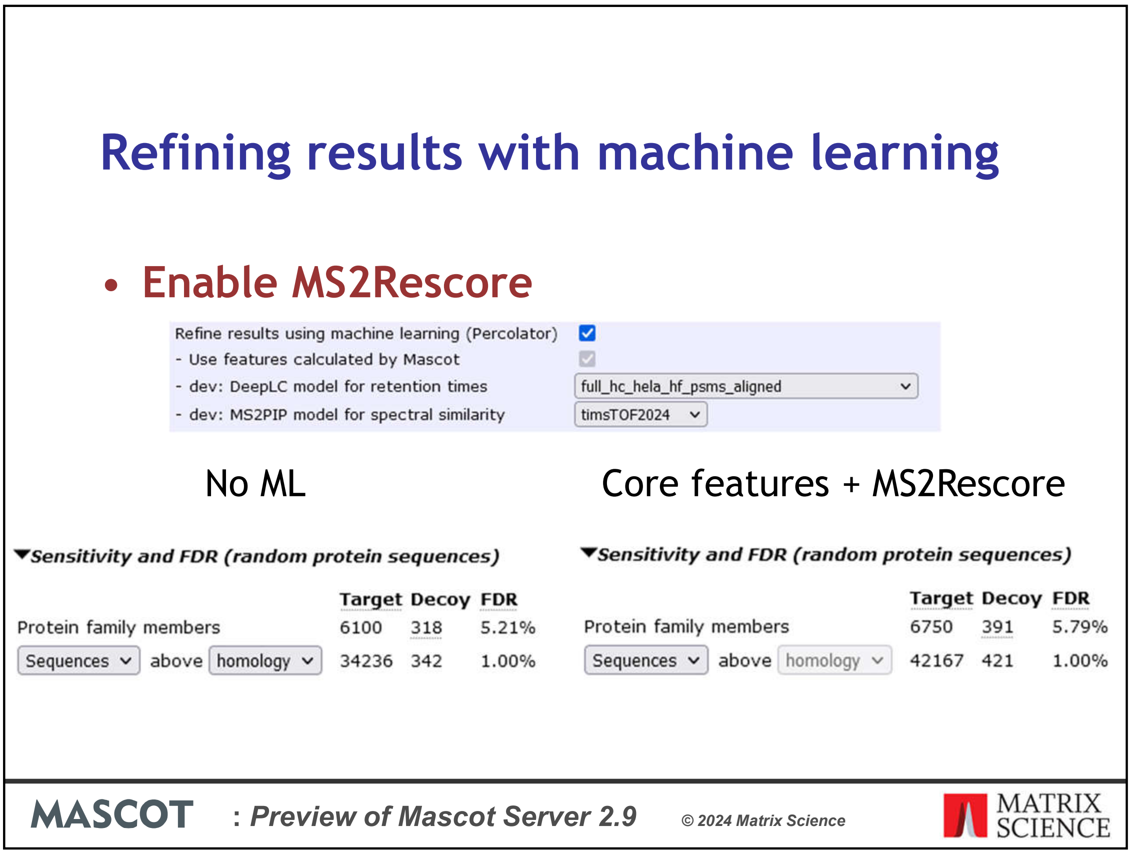
Refining with machine learning Computing predicted features... (94s elapsed)

40% done

Open results file Reading file... (0 s elapsed)

100% done

So, click the button to enable MS2Rescore, you’ll see a progress bar…



Refining results with machine learning

- Enable MS2Rescore

Refine results using machine learning (Percolator)	☑
- Use features calculated by Mascot	☑
- dev: DeepLC model for retention times	full_hc_hela_hf_psms_aligned
- dev: MS2PIP model for spectral similarity	timsTOF2024

No ML

▼*Sensitivity and FDR (random protein sequences)*

	Target	Decoy	FDR
Protein family members	6100	318	5.21%
Sequences above homology	34236	342	1.00%

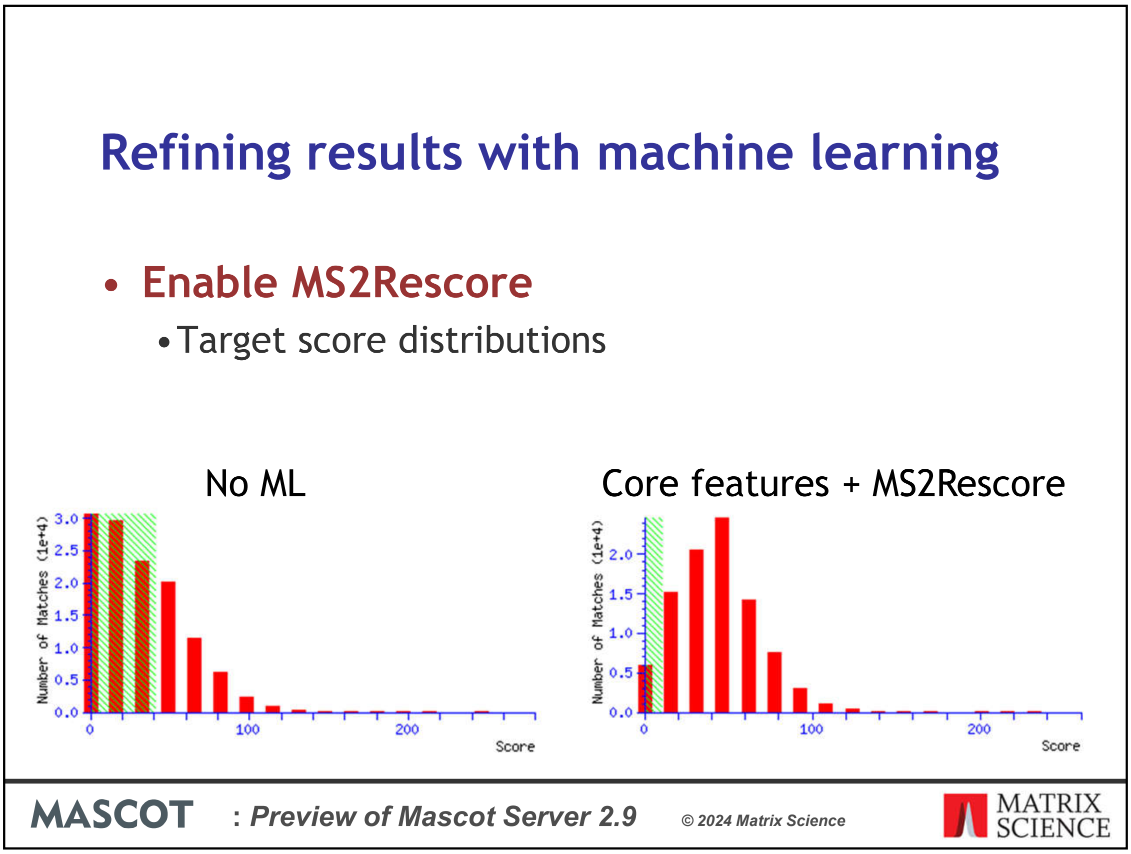
Core features + MS2Rescore

▼*Sensitivity and FDR (random protein sequences)*

	Target	Decoy	FDR
Protein family members	6750	391	5.79%
Sequences above homology	42167	421	1.00%

MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

We now get 42k peptide sequences at the same FDR, compared to 34k without machine learning. Protein FDR is a bit high in this data set, nearly 6%, but it can be easily fixed by selecting a tighter FDR threshold. Nonetheless, we're getting hundreds more protein hits by using machine learning.



Refining results with machine learning

- **Enable MS2Rescore**
  - Target score distributions

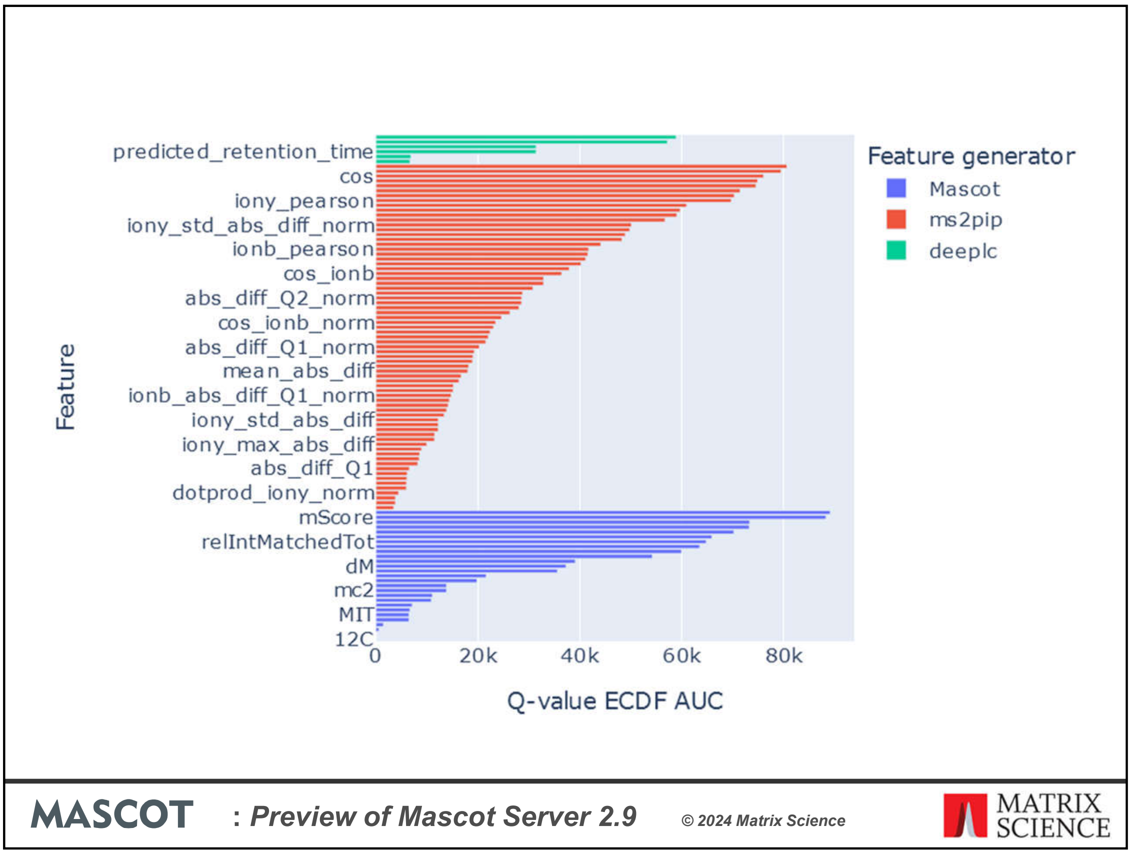
No ML

Core features + MS2Rescore

MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

Here are the target match score distributions before any machine learning, and with MS2Rescore enabled. The shape of the distribution is much better when machine learning is enabled. Even in this standard tryptic HeLa search, it is giving better sensitivity at the same FDR.

Why does machine learning make the results better? Neither Percolator nor MS2Rescore invent new peptide matches. You get better sensitivity because you're using many dimensions of extra information that isn't available to the database search engine.

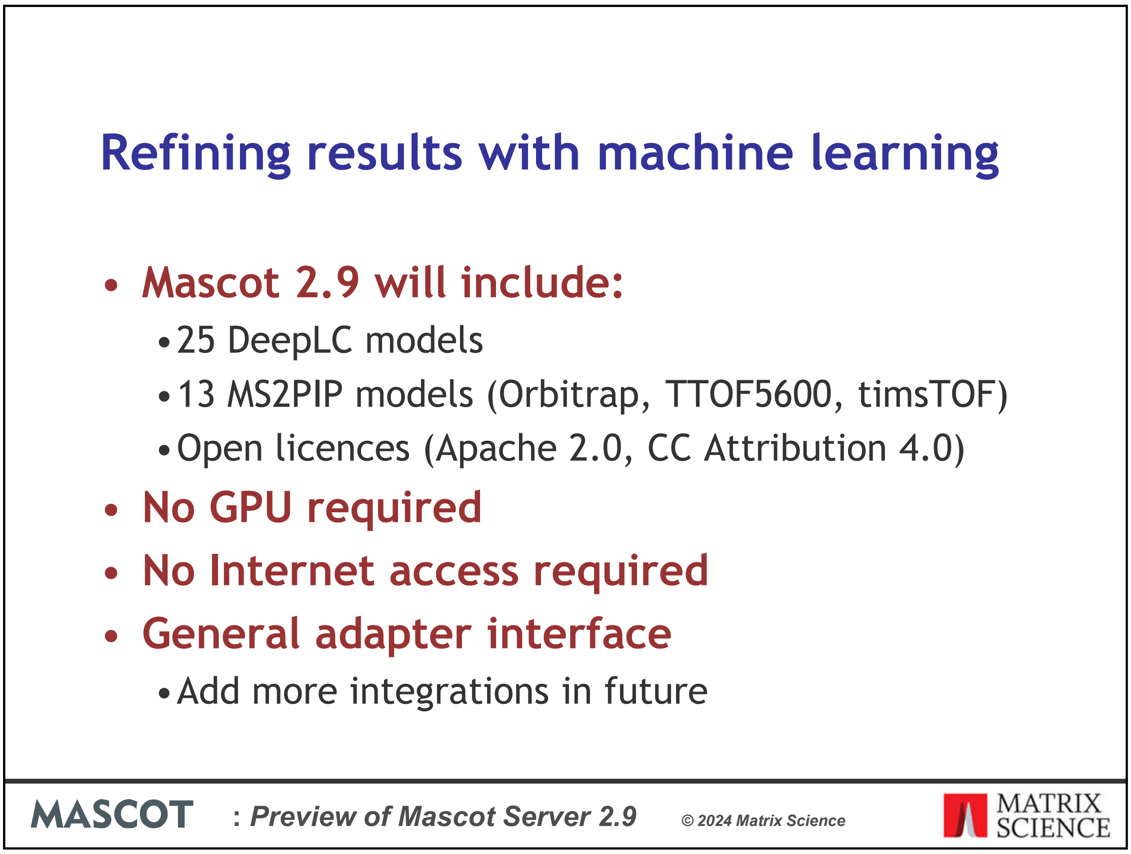


MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

This chart is generated by MS2Rescore and will be included as part of Mascot reports. For each feature, it basically shows how good that feature is for discriminating between correct and incorrect matches. The ones at the top in green are DeepLC retention time features. The red ones are MS2PIP spectral correlation metrics (there are quite a few of them!). At the bottom in blue are the core features calculated by Mascot.

In the Mascot core features, you can see that mScore, which is the Mascot score, is pretty good at separating the wheat from the chaff. The MS2PIP features are prominent too, showing the cosine similarity between predicted spectrum and observed spectrum as an excellent discriminator. The DeepLC features are bit less prominent, but keep in mind the retention time features are completely orthogonal to the other match data. They provide information that isn't available to the search engine.

The orthogonal features simply provide a boost to peptides that the database search found, but which maybe don't have good fragmentation evidence. Percolator learns the characteristics of correct and incorrect matches from the decoy results. If a peptide didn't have a good Mascot score, but it does fit all the other criteria of being a correct match, then the machine learning will boost it accordingly.



Refining results with machine learning

- Mascot 2.9 will include:
  - 25 DeepLC models
  - 13 MS2PIP models (Orbitrap, TTOF5600, timsTOF)
  - Open licences (Apache 2.0, CC Attribution 4.0)
- No GPU required
- No Internet access required
- General adapter interface
  - Add more integrations in future

Just to recap, Mascot Server 2.9 includes MS2Rescore with DeepLC and MS2PIP.

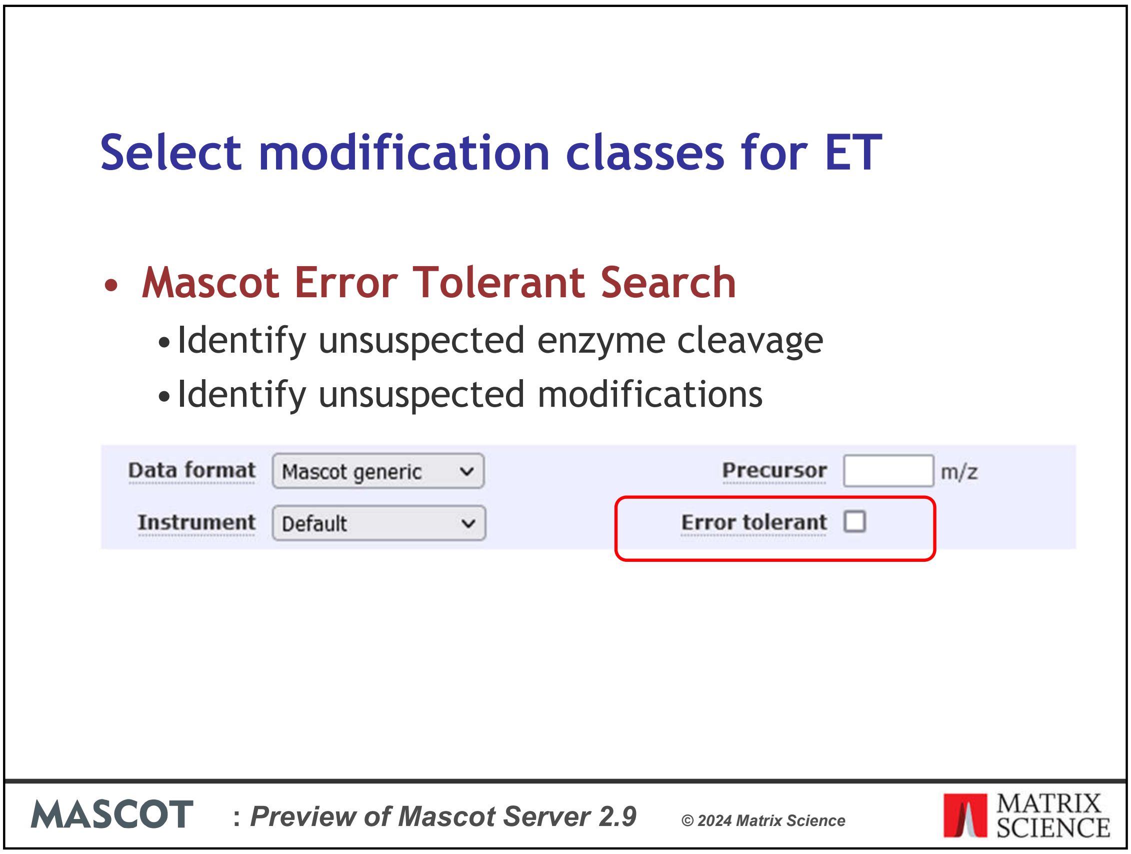
DeepLC has 25 built-in models for various tryptic and non-tryptic peptides, some with fixed or variable mods.

MS2PIP has 13 models for different instrument types, enzymatic cleavage and variable mods. Mascot will ship with models for Thermo Orbitrap, Sciex TTOF5600 and Bruker timsTOF.

The models have been trained by the MS2Rescore developers and used in several publications already. They are based on carefully selected and curated, publicly available data sets. The model files themselves are publicly available, although we will also ship them with Mascot. They are licensed under the Apache 2.0 licence and Creative Commons Attribution 4.0, which means they are free for commercial use. You should always cite the MS2Rescore papers when you use them!

This all runs on the CPU, so no GPU is initially required. No Internet access is required to use MS2Rescore in Mascot. All the software and models are installed locally as part of Mascot Server, so none of your data is sent outside Mascot.

I don’t have time to go into implementation details in this talk, but the MS2Rescore integration is implemented using a general adapter interface. We plan to add more integrations in future, and you can add your own too.



Select modification classes for ET

- Mascot Error Tolerant Search
  - Identify unsuspected enzyme cleavage
  - Identify unsuspected modifications

Data format: Mascot generic

Instrument: Default

Precursor: m/z

Error tolerant □

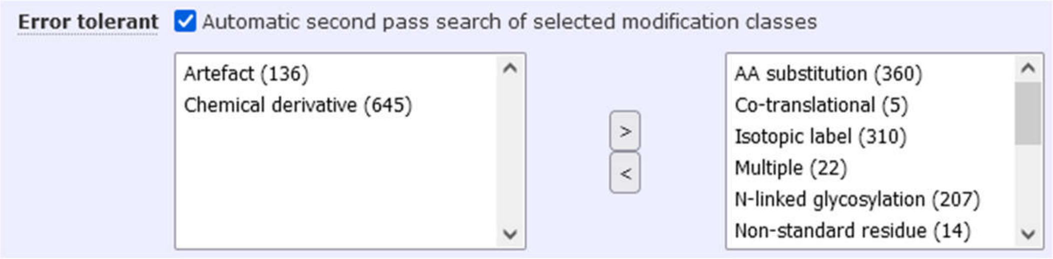
MASCOT : *Preview of Mascot Server 2.9* *© 2024 Matrix Science*

I’ll quickly demonstrate another useful feature in the next release. The Mascot error tolerant search is a two-pass search. The first pass is a normal search. Mascot selects proteins for the second pass based on significant peptide matches. In the second pass, Mascot relaxes enzyme specificity and also tries to find unsuspected modifications in the selected proteins.

In the previous version, you enable error tolerant searching by ticking this small checkbox in the search form.

Select modification classes for ET

- New: select modification classes



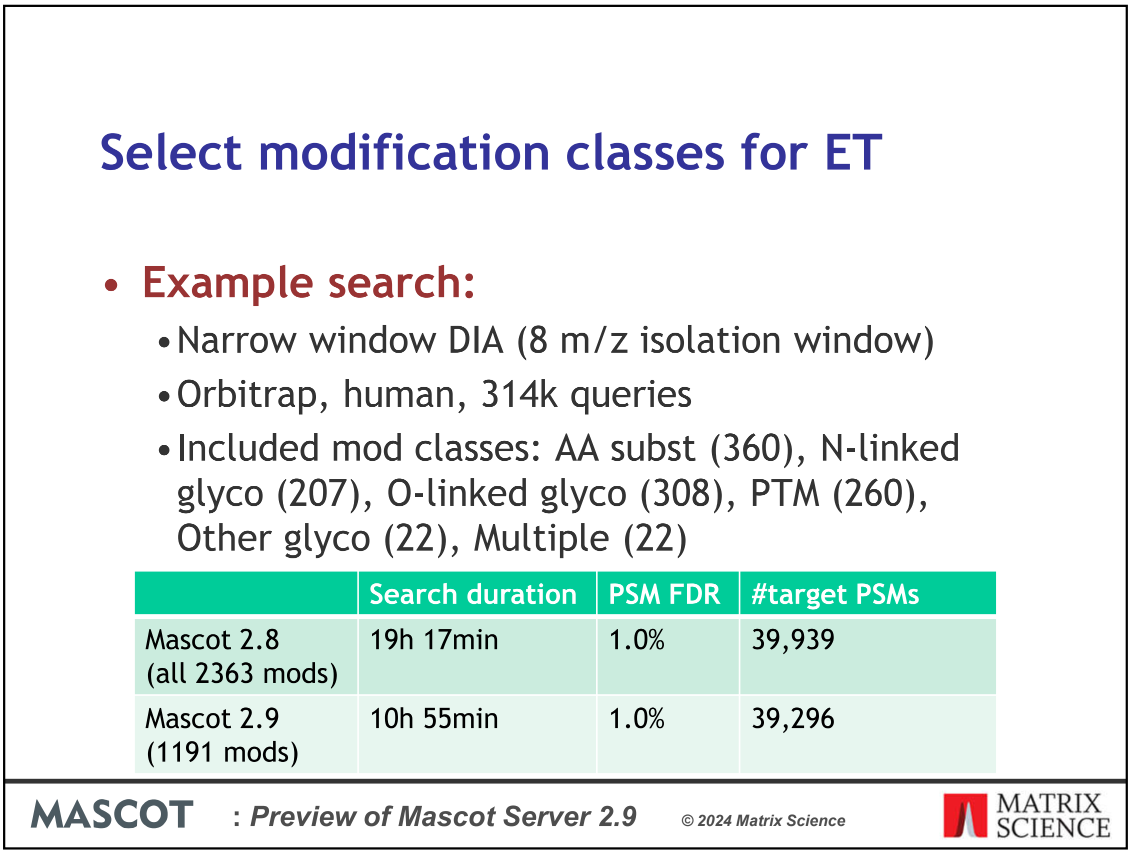
Error tolerant ☑ Automatic second pass search of selected modification classes

Artefact (136)
Chemical derivative (645)

AA substitution (360)
Co-translational (5)
Isotopic label (310)
Multiple (22)
N-linked glycosylation (207)
Non-standard residue (14)

In the new version, we have made the checkbox more prominent. We have also added controls for selecting modification classes. Normally, the error tolerant search tries every entry in Unimod, which is over 2000 modifications. However, most of the time, you have a rough idea what is definitely not in the sample. In this screenshot, I've selected just artefacts and chemical derivatives for the second pass search. I've deselected AA substitutions, isotopic labels, glycosylation and so on.

There are several use cases here. You could choose to search only N-linked glycosylation. Or you could choose to search only post-translational modifications. The error tolerant search is a tool for exploratory analysis.



Select modification classes for ET

- **Example search:**
  - Narrow window DIA (8 m/z isolation window)
  - Orbitrap, human, 314k queries
  - Included mod classes: AA subst (360), N-linked glyco (207), O-linked glyco (308), PTM (260), Other glyco (22), Multiple (22)

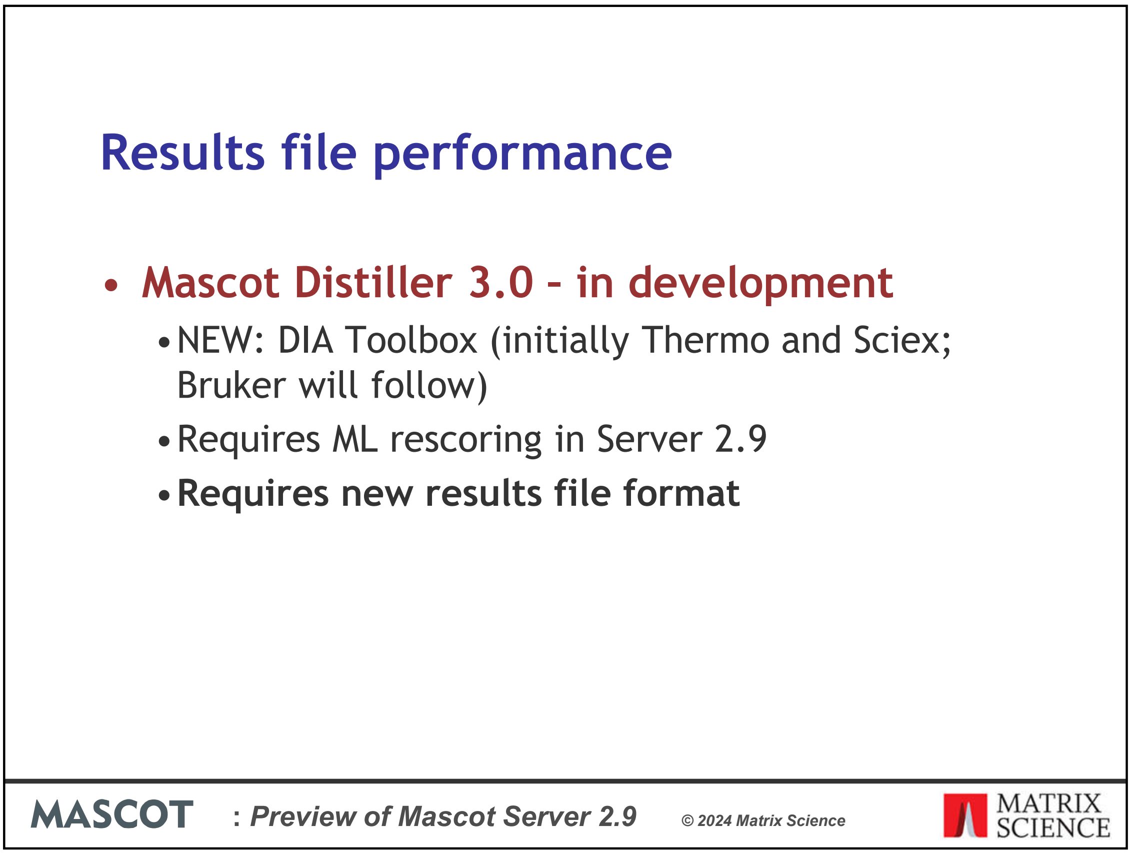
	Search duration	PSM FDR	#target PSMs
Mascot 2.8 (all 2363 mods)	19h 17min	1.0%	39,939
Mascot 2.9 (1191 mods)	10h 55min	1.0%	39,296

MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

Just to show some concrete numbers, this is from a narrow window DIA search that we’re experimenting with. It’s a human sample with 314k queries from a Thermo Orbitrap.

Using the new search controls, I’ve included these modification classes for the ET search: amino acid substitutions, N-linked and O-linked glycosylation, PTMs, other glycosylation and mods in multiple categories. This is about half the modifications available in Unimod. The other half is things like artefacts, isotopic labels and chemical derivatives.

The search duration is from an in-house server that is now a few years old. Using the new controls, the search is much faster, almost 30% faster, because you’re searching half as many modifications. You’re getting pretty much the same results at the same FDR. I don’t have space here for a detailed comparison of the modifications discovered, so we’ll write that up in our blog later.



Results file performance

- **Mascot Distiller 3.0 – in development**
  - NEW: DIA Toolbox (initially Thermo and Sciex; Bruker will follow)
  - Requires ML rescoring in Server 2.9
  - **Requires new results file format**

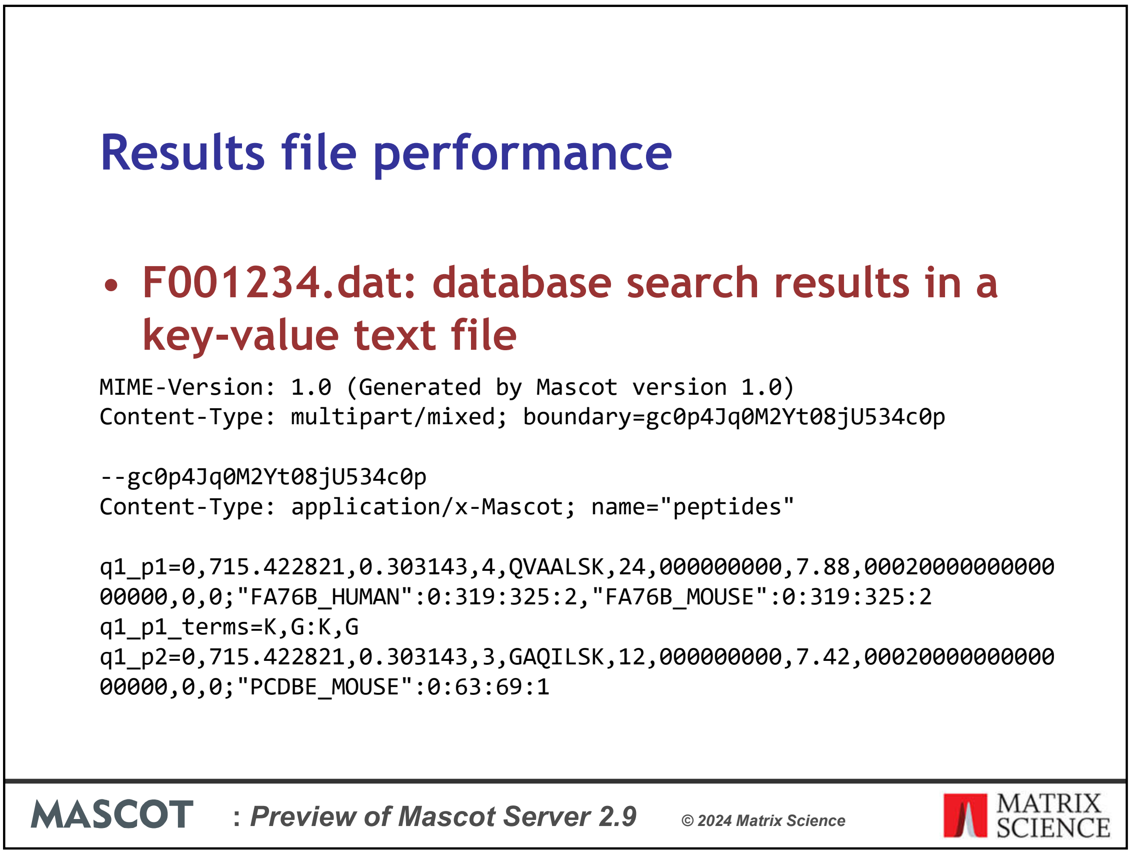
MASCOT : *Preview of Mascot Server 2.9*

MATRIX SCIENCE

I just hinted at DIA support. As well as Mascot Server 2.9, we are also working on the next version of Mascot Distiller, which will be version 3.0. It's too early to show screenshots today. However, I can tell you this much: we will introduce a brand new toolbox for DIA, initially for Thermo and Sciex raw files.

The DIA toolbox requires the new ML rescoring feature in Server 2.9, which I've just shown to you. Distiller will make extensive use of machine learning with DIA and label-free quantitation to get the best results.

Because DIA searches are a lot larger than DDA searches, DIA support will only be feasible if we dramatically improve the results file performance. We're making the backend changes in Mascot Server 2.9 in preparation for Distiller 3.0.



Results file performance

- **F001234.dat: database search results in a key-value text file**

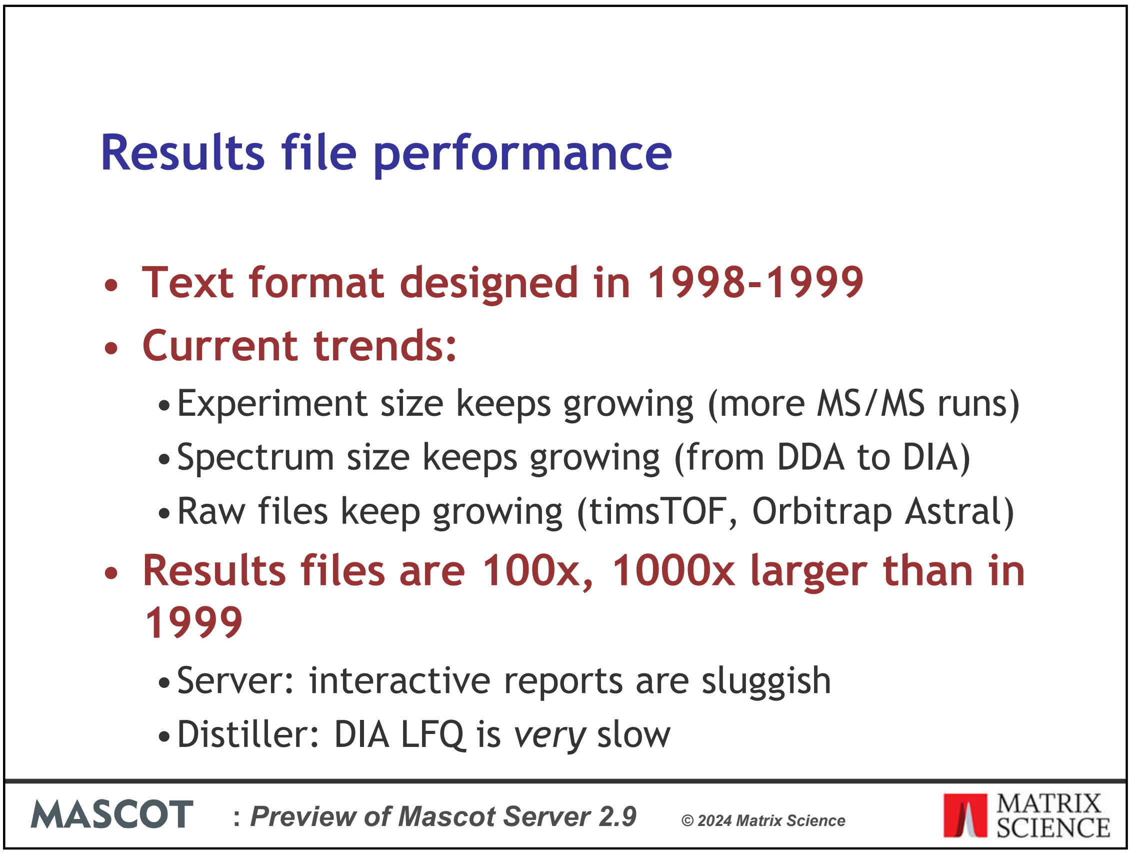
```
MIME-Version: 1.0 (Generated by Mascot version 1.0)
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08jU534c0p

--gc0p4Jq0M2Yt08jU534c0p
Content-Type: application/x-Mascot; name="peptides"

q1_p1=0,715.422821,0.303143,4,QVAALSK,24,000000000,7.88,00020000000000
00000,0,0;"FA76B_HUMAN":0:319:325:2,"FA76B_MOUSE":0:319:325:2
q1_p1_terms=K,G:K,G
q1_p2=0,715.422821,0.303143,3,GAQILSK,12,000000000,7.42,00020000000000
00000,0,0;"PCDBE_MOUSE":0:63:69:1
```

Mascot currently saves database search results in a key-value text file with a .dat extension.

Here’s a small example. It’s a MIME-format file where the sections are separated by a boundary string. Each section has key-value fields. In the case of peptides, the values are actually comma-separated and semicolon-separated lists with internal structure.



Results file performance

- **Text format designed in 1998-1999**
- **Current trends:**
  - Experiment size keeps growing (more MS/MS runs)
  - Spectrum size keeps growing (from DDA to DIA)
  - Raw files keep growing (timsTOF, Orbitrap Astral)
- **Results files are 100x, 1000x larger than in 1999**
  - Server: interactive reports are sluggish
  - Distiller: DIA LFQ is *very* slow

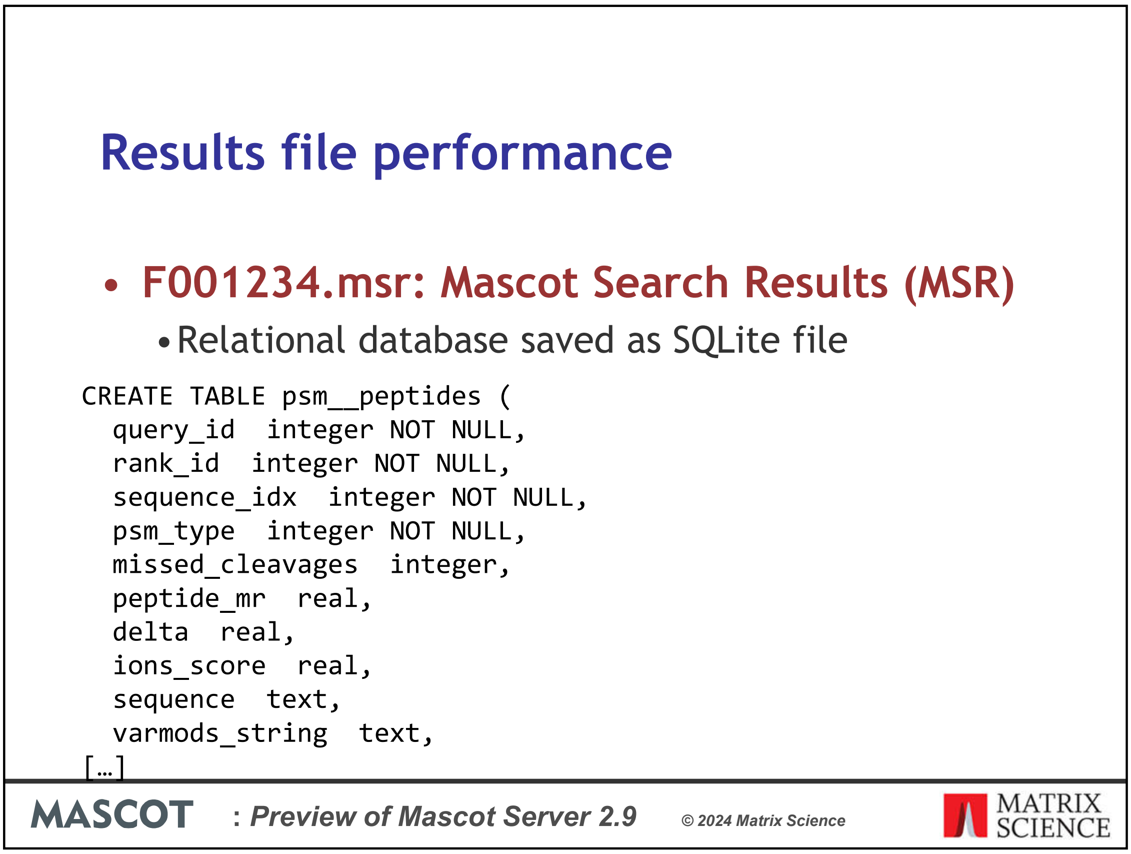
MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

The text format was designed back in 1999 for the first version of Mascot, although it has been extended several times over the years. The advantage of a text format is that it's quick for Mascot to write and it's also easy for a human to open the file in a text editor.

However, there are a few problems. Chief among them is reading speed and throughput. Accessing a gigabyte text file is slow, especially if you need just a few lines from the middle of the file.

There are three trends that are making the situation worse. Experiment size keeps growing: you have more MS/MS runs to merge. Spectrum size keeps growing as people are moving from DDA to DIA, because DIA spectra have more fragment peaks. And raw files keep growing in size as instrument scan rates keep increasing. A single DIA raw file typically has hundreds of thousands of MS/MS scans.

As a result, the text-based results files are now 100 or even 1000 larger than when the file format was designed. Merging text files into a single report, for example for label-free quantitation, can be done, but data access is a performance bottleneck in Distiller. It's especially bad with DIA LFQ projects we've experimented with.



Results file performance

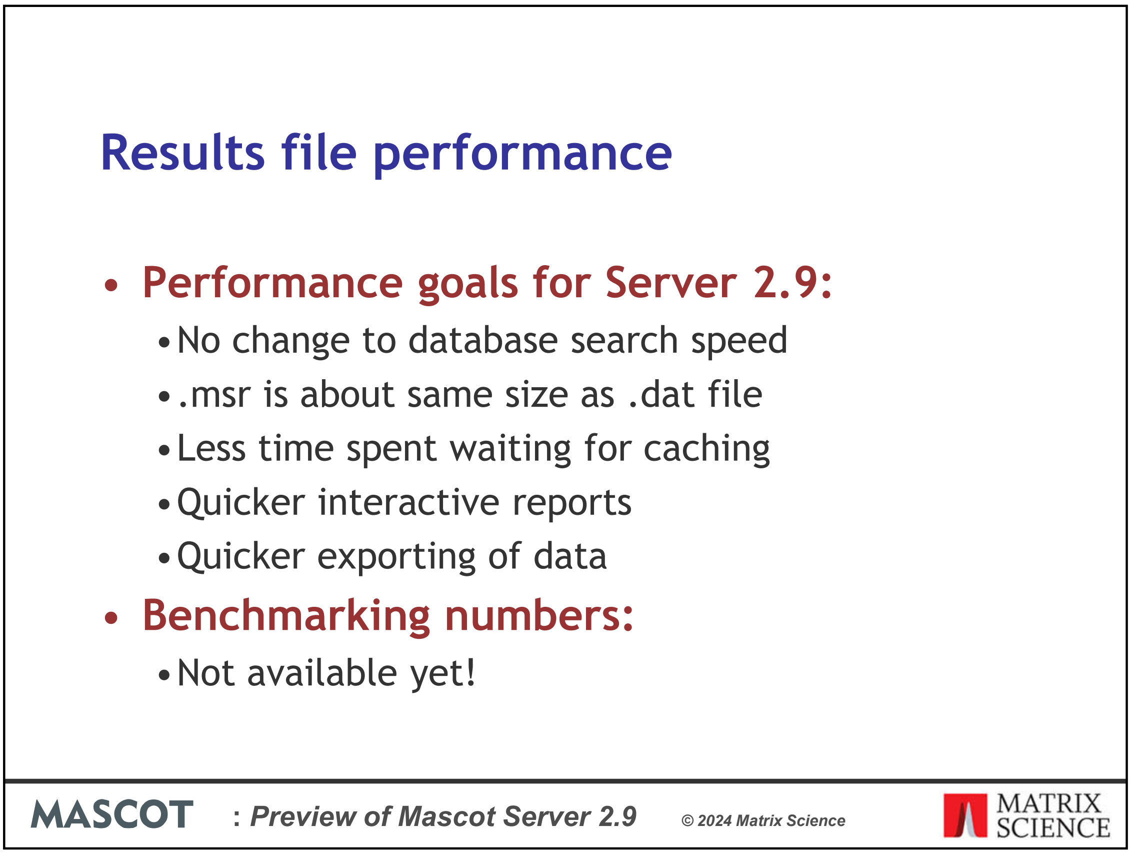
- **F001234.msr: Mascot Search Results (MSR)**
  - Relational database saved as SQLite file

```
CREATE TABLE psm__peptides (
  query_id  integer NOT NULL,
  rank_id  integer NOT NULL,
  sequence_idx  integer NOT NULL,
  psm_type  integer NOT NULL,
  missed_cleavages  integer,
  peptide_mr  real,
  delta  real,
  ions_score  real,
  sequence  text,
  varmods_string  text,
[…]
```

So, in the next version, we have designed a brand new format for Mascot Search Results. The file extension is MSR. This is a proper relational database saved as an SQLite file. Relational databases are really great for random access and retrieving slices of data.

SQLite is used elsewhere in proteomics, for example Thermo Proteome Discover files are SQLite databases, and Bruker timsTOF raw files use SQLite for storing scan metadata. SQLite is also used as an application file format in the Firefox web browser and many other places. We chose SQLite because it's a tried and trusted format with very good performance.

Here's a short example of the table definition for peptide matches. It's standard SQL where the data fields are divided into columns.



Results file performance

- **Performance goals for Server 2.9:**
  - No change to database search speed
  - .msr is about same size as .dat file
  - Less time spent waiting for caching
  - Quicker interactive reports
  - Quicker exporting of data
- **Benchmarking numbers:**
  - Not available yet!

We've finished designing the new file format and we're finalising the implementation. Although this change is primarily for Mascot Distiller, these are the performance goals for we have in mind for Server 2.9:

There should be no change to database search speed compared to the old format.

The new file should be about the same size as the old format. We don't want to increase disk usage for no reason.

There's always a caching step at the end of the search and before the results report loads. We plan to reduce the time you need to wait for caching to finish.

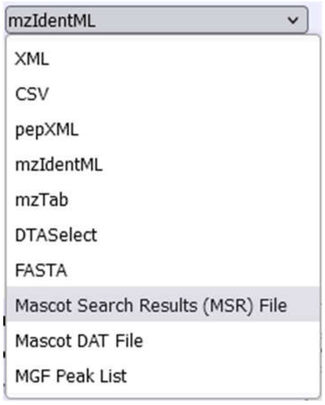
We are also trying to reduce the delays when you click on items in the interactive reports.

We also plan to reduce the time it takes to export the results as mzIdentML or XML.

Unfortunately, we're still benchmarking and optimising the performance, so I can't give concrete numbers yet. We'll publish benchmarking figures when we're closer to the release.

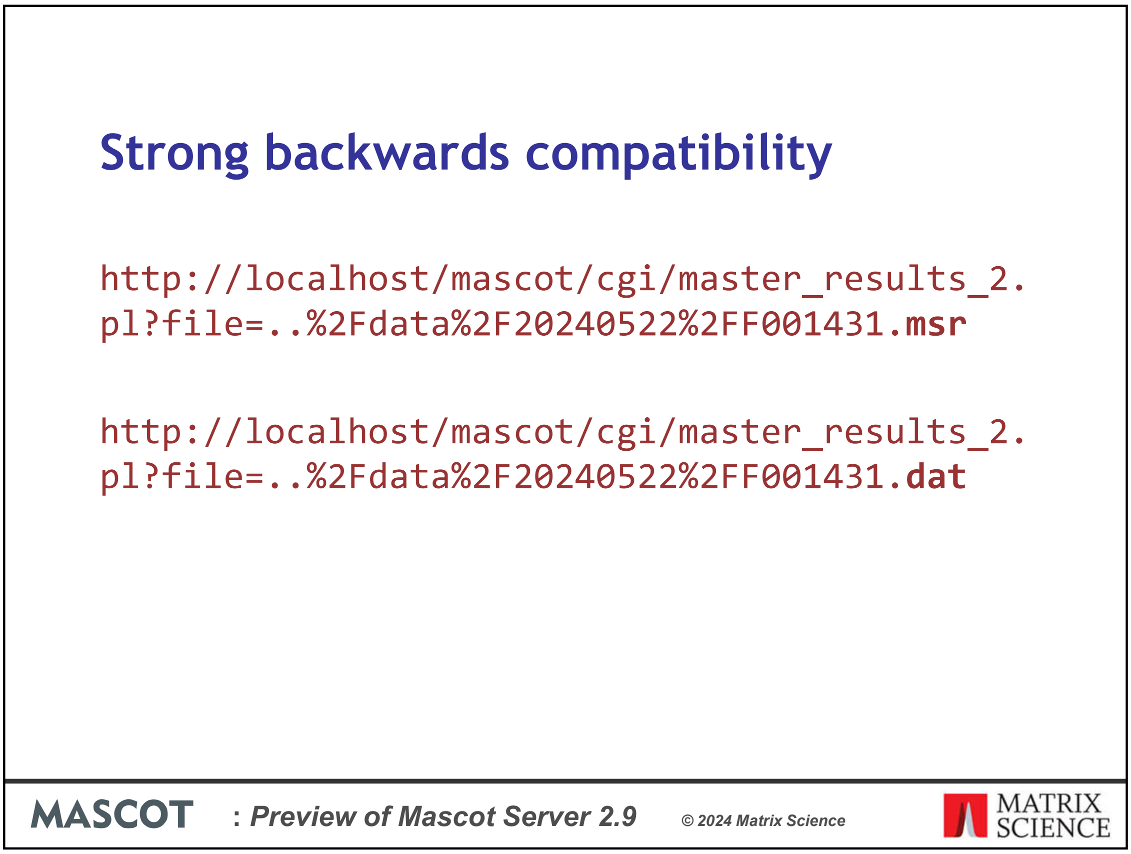
Strong backwards compatibility

- Export MSR files in all formats
  - mzIdentML, mzTab, Mascot CSV, Mascot XML
  - Export in dat28 format
  - Download as an SQL dump



We’ve worked hard to support every use case and maintain backwards compatibility. All search types are saved in MSR format by default. There’s no visible change in the results reports, which seamlessly support both the old and new file formats.

You can export the results in all existing formats the same way as before. You can also export the MSR file in the old text format, which we have named the dat28 format (it didn’t have a formal name before). And you can now export the search results as an SQL dump, which may be of interest if you’re a software developer.

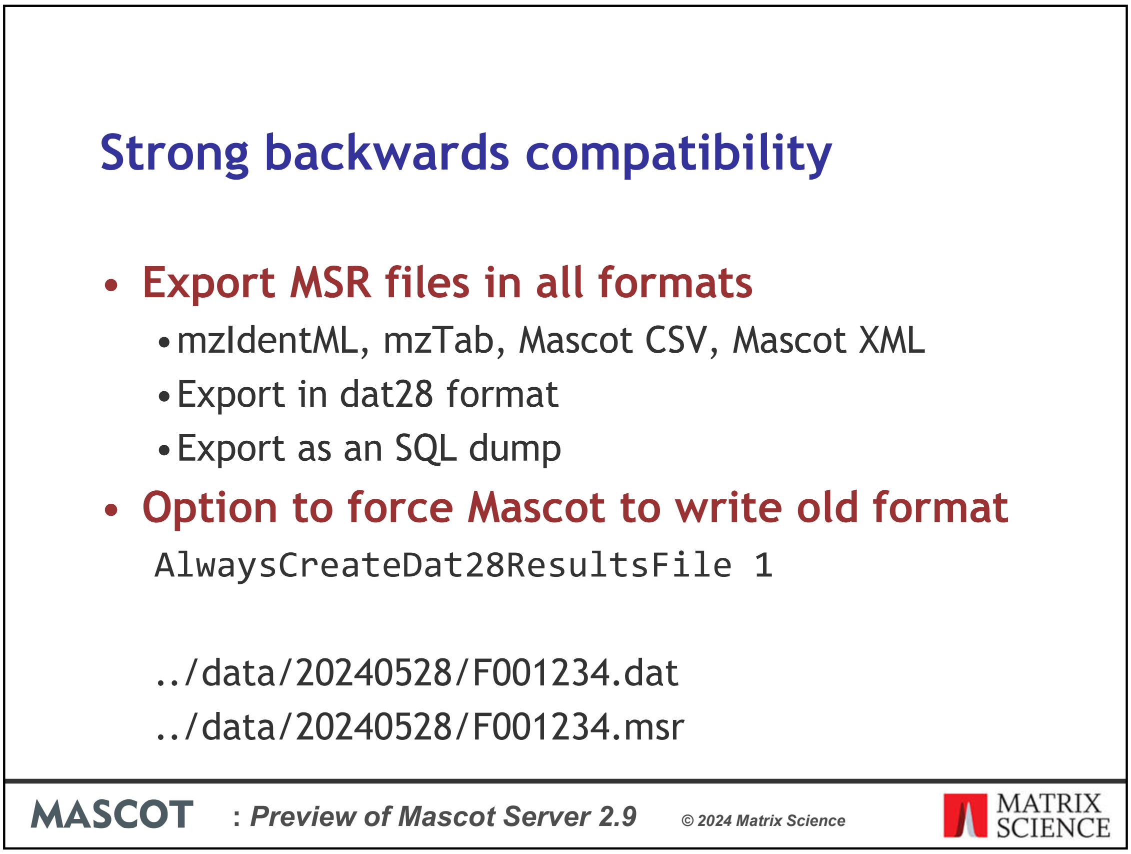


Strong backwards compatibility

http://localhost/mascot/cgi/master_results_2.pl?file=..%2Fdata%2F20240522%2FF001431.**msr**

http://localhost/mascot/cgi/master_results_2.pl?file=..%2Fdata%2F20240522%2FF001431.**dat**

There is literally no difference between the results report of the search saved in MSR format and the same search in dat28 format. The only difference you can see is the file extension in the URL.



Strong backwards compatibility

- Export MSR files in all formats
  - mzIdentML, mzTab, Mascot CSV, Mascot XML
  - Export in dat28 format
  - Export as an SQL dump
- Option to force Mascot to write old format

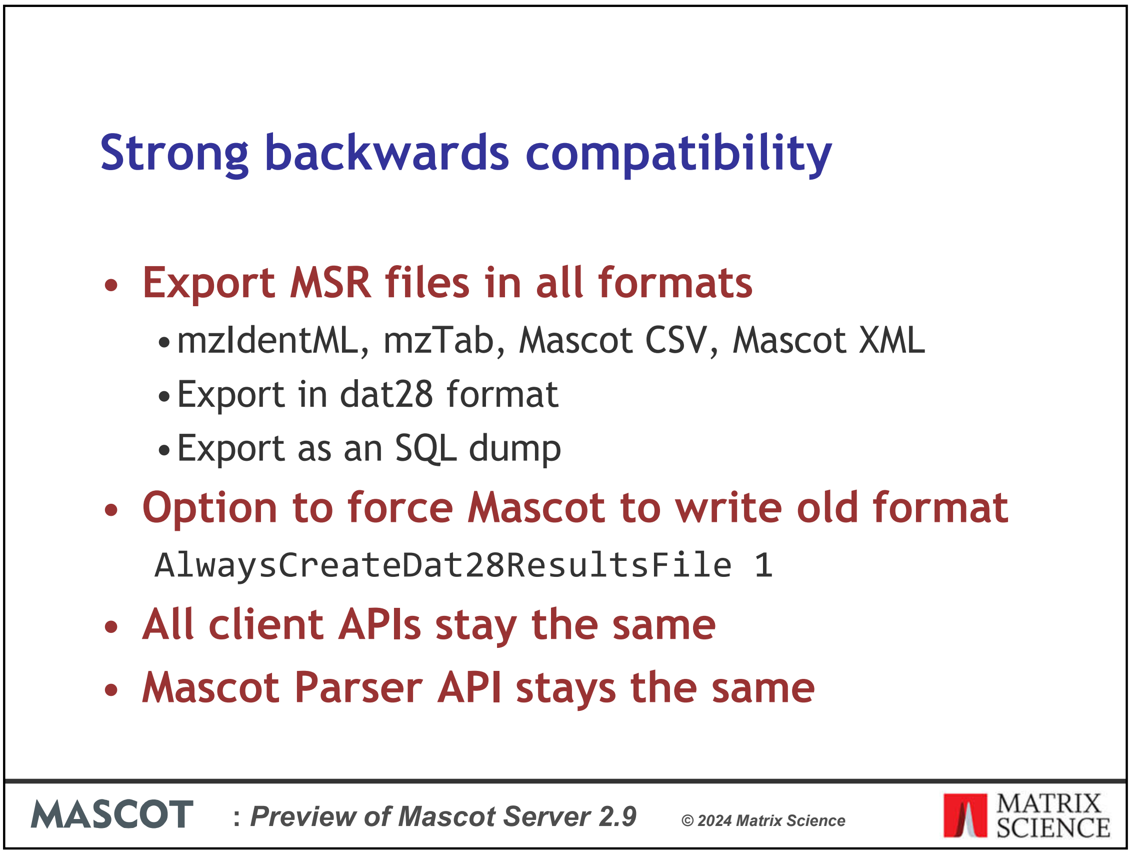
```
AlwaysCreateDat28ResultsFile 1
```

```
../data/20240528/F001234.dat
../data/20240528/F001234.msr
```

MASCOT : *Preview of Mascot Server 2.9* © 2024 Matrix Science MATRIX SCIENCE

Most applications download Mascot results using the client API. If your application reads the results file directly from the Mascot server filesystem, you can force Mascot to create the dat28 file at the end of the search. The option is called AlwaysCreateDat28ResultsFile. This is turned off by default. If you enable it, you will double your disk space usage but this guarantees 100% backwards compatibility.

Most client applications download the results using the client.pl API or using export_dat_2.pl, so you won’t need this option.



Strong backwards compatibility

- Export MSR files in all formats
  - mzIdentML, mzTab, Mascot CSV, Mascot XML
  - Export in dat28 format
  - Export as an SQL dump
- Option to force Mascot to write old format

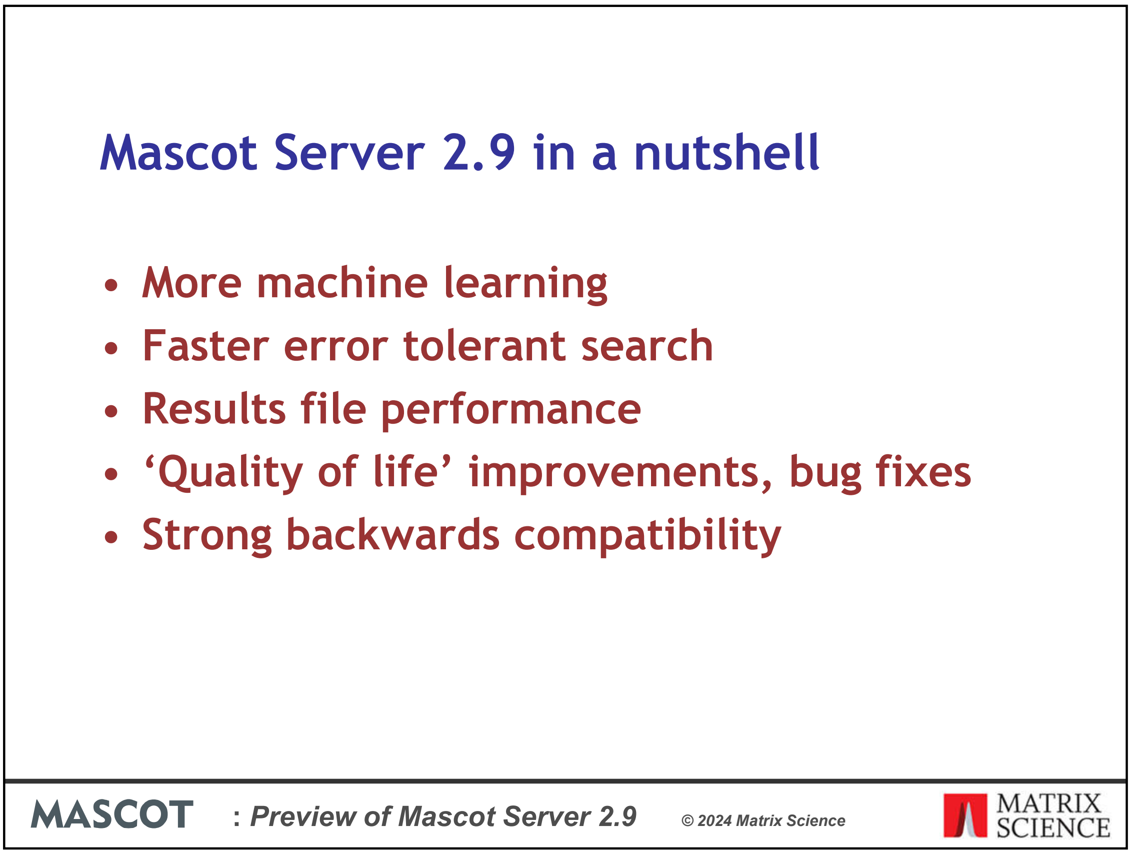
```
AlwaysCreateDat28ResultsFile 1
```

- All client APIs stay the same
- Mascot Parser API stays the same

All the client APIs for downloading results stay the same. You can use older versions of Mascot Distiller with the new Mascot and you can continue using third-party software like Thermo Proteome Discoverer. We've worked hard not to break existing integrations.

We have even succeeded in keeping the Mascot Parser API exactly the same between the two file formats. If your application uses Mascot Parser, just update to the new version. You need to change one constructor call that autodetects the file format and that's it.

If your application has custom code for parsing the old text format, I strongly recommend switching to Mascot Parser, as it will save you a lot of headaches.



Mascot Server 2.9 in a nutshell

- More machine learning
- Faster error tolerant search
- Results file performance
- ‘Quality of life’ improvements, bug fixes
- Strong backwards compatibility

Thank you for your attention. Here’s a summary of what’s in the new release.