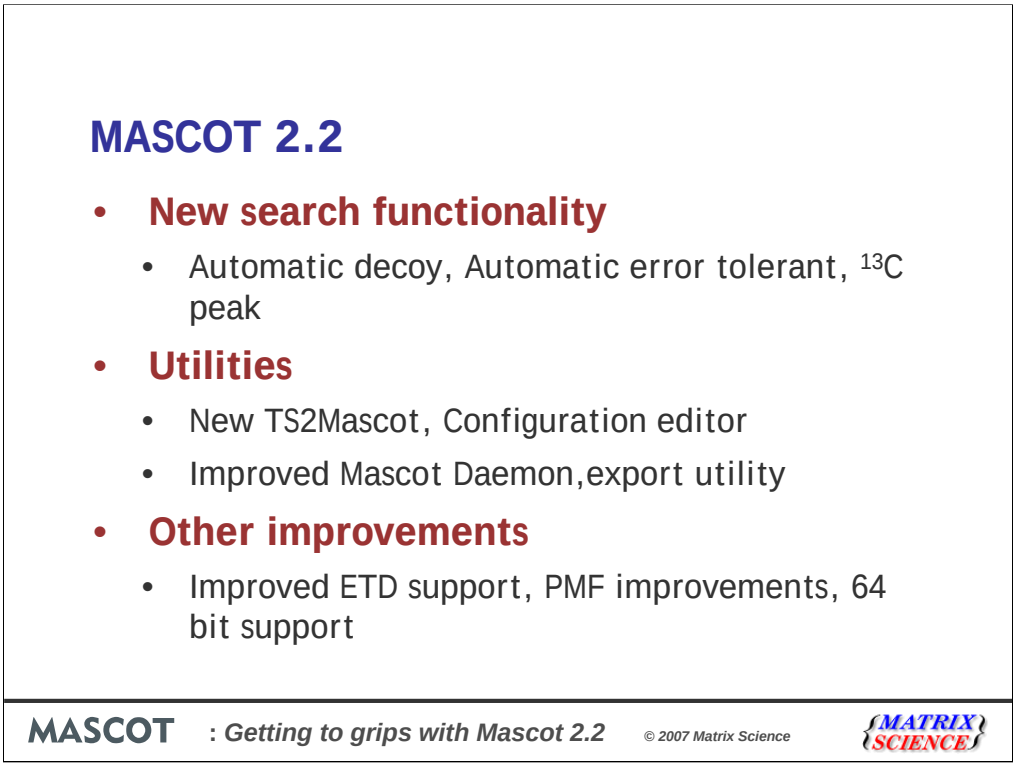
Getting to grips with Mascot 2.2

MASCOT

MATRIX SCIENCE



In this session, I'll cover some of the new features available in Mascot 2.2 which we released a few months ago.

Quantitation support was covered in an earlier talk, so I won't be going over that again now.

I'll start by describing the major new functionality for the search engine, including the automatic decoy search and the automatic error tolerant facility.

Next I'll describe the new and improved utilities that are provided with the Mascot Server software.

Finally, there are a large number of other improvements in Mascot 2.2, but I've selected just three which hopefully are of interest to a good number of people here today.

Automatic Decoy

"Decoy" database

- Direct estimate of false positive rate (FPR)
- Requires large dataset to get accurate estimate of FPR
- Not a substitute for a reliable scoring scheme
- What makes a good decoy database?

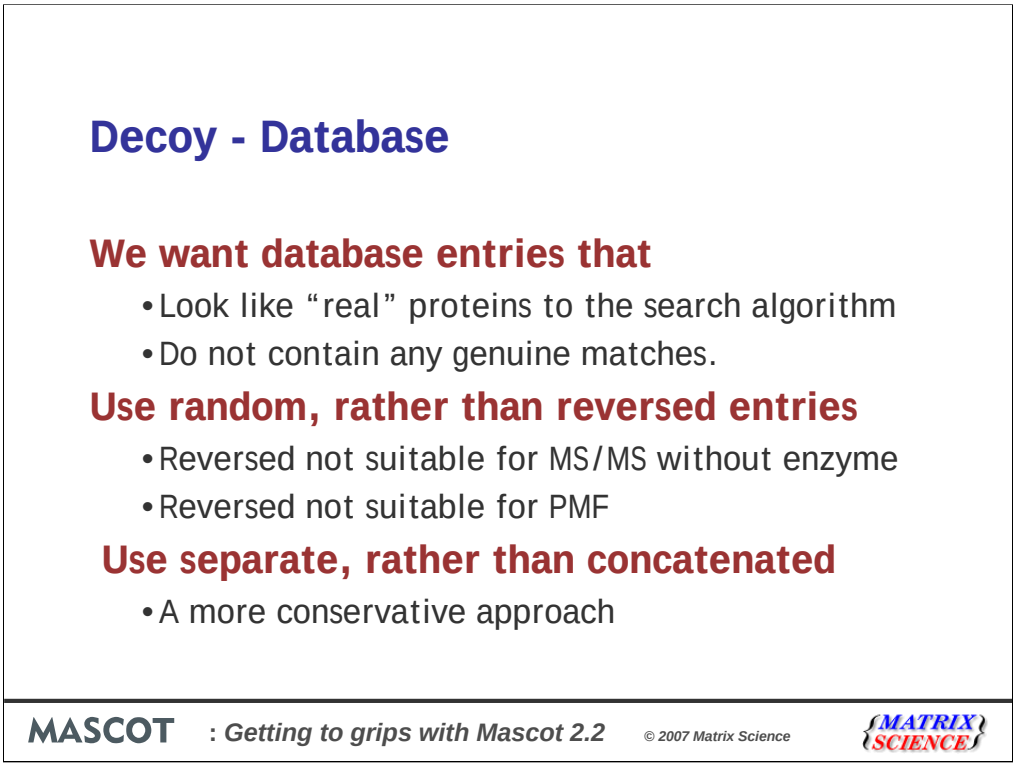
Firstly, the automatic decoy.

I think it was the Gygi group first coined the term "decoy database" for this approach. The idea is to repeat the search, using identical search parameters, against a database in which the sequences have been reversed or scrambled.

You do not expect to get any significant matches from the decoy database. So, the number of matches that are found is an excellent estimate of the false positive rate in the original search.

This is an excellent validation method for MS/MS searches of large data sets. It is not as useful for a search of a small number of spectra, because the numbers are too low to give an accurate estimate of the false positive rate. Hence, it is not a substitute for a reliable scoring algorithm.

What are the requirements for a decoy database?



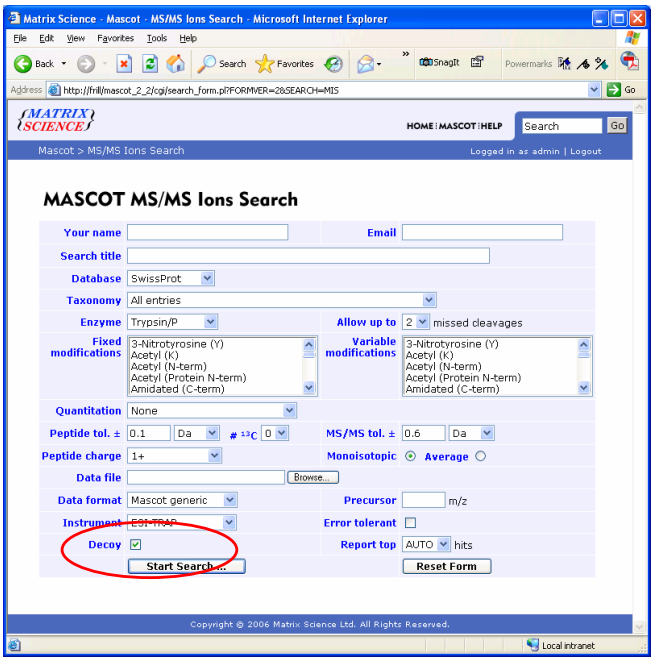
This is actually a very difficult question. Briefly:

We want database entries that look like "real" proteins to the search algorithm

We want database entries that don't contain genuine peptide sequences (for MS/MS) or real peptide masses (for MS)

We've chosen to use a random database, rather than a database with each sequence reversed. A reversed database is not suitable for no-enzyme MS/MS searches, especially when there are several variable mods, because it is possible to get mass shift at each end of a reversed peptide sequence that just happens to transform a genuine y series match into a false b series match or vice versa. It's also not suitable for PMF searches.

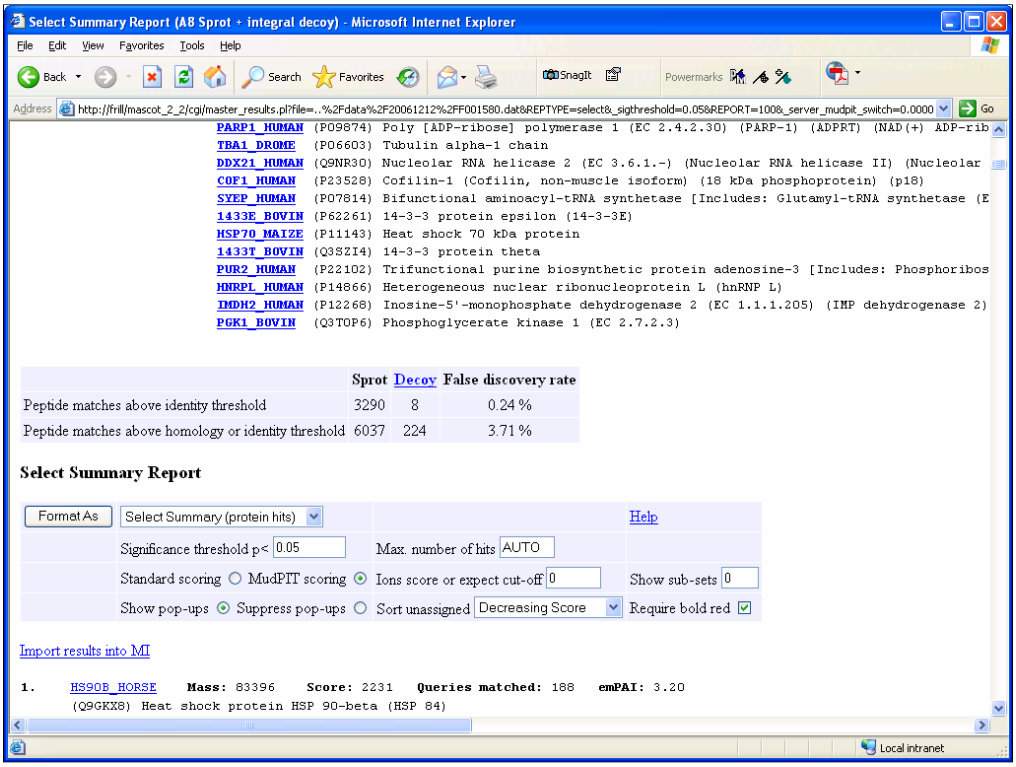
We've also chosen to treat the randomised database as a separate database rather than concatenating the original database and the randomised database. Consider a case where the top match two matches to an ms-ms spectrum are both significant, and the first match is to the real database, while the second match is to the randomised database. For a concatenated database, we only consider the top match so this isn't a false positive, whereas it would be if the databases were searched separately.



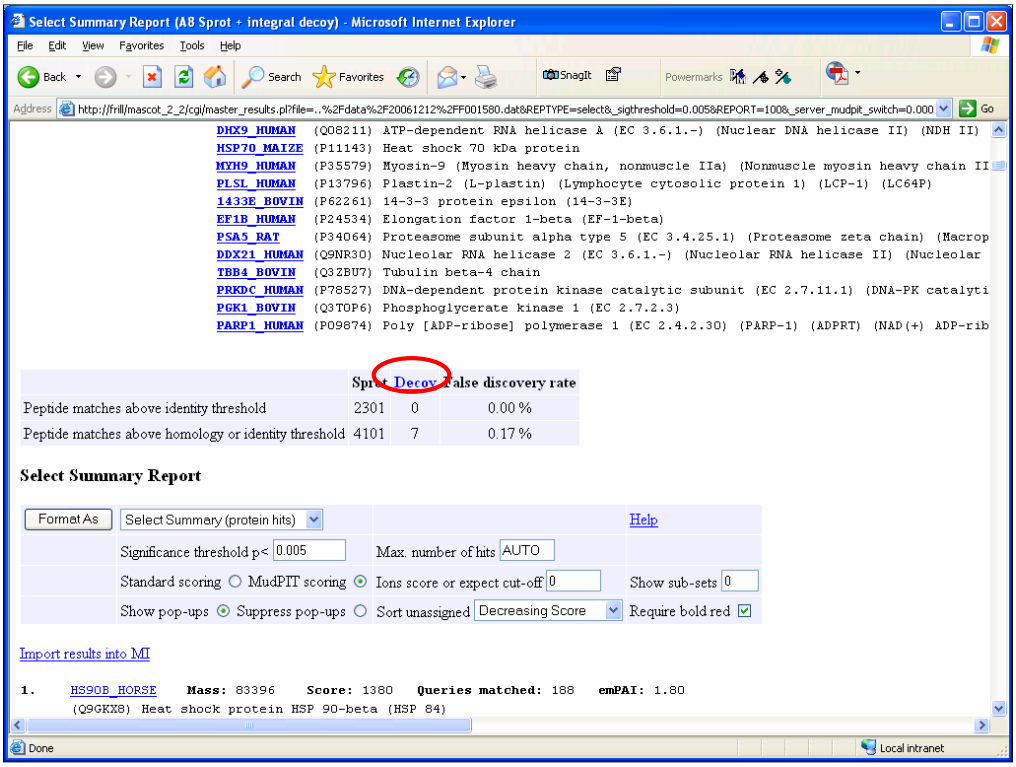
On our public web site there is a help page devoted to decoy database searches. There is a link there to a small utility program that allows you to create a randomised or reversed database. If you have an earlier version of Mascot, or if you want to verify the results from another search engine, I recommend that you use that utility.

Because more and more people wish to perform decoy searches routinely, we've added this into Mascot as a built-in part of the search. If you choose the Decoy checkbox on the search form, then every time a protein or peptide sequence from the "forward" database is tested, a random sequence of the same length is automatically generated and tested. The average amino acid composition of the random sequences is the same as the average composition of the forward database. The matches and scores for the random sequences are recorded separately in the result file. The effect is identical to searching a separate database rather than a concatenated database.

For our testing, we spent some time comparing the results from the new automatic decoy search and using a separately generated database.Obviously, you can do the same thing.

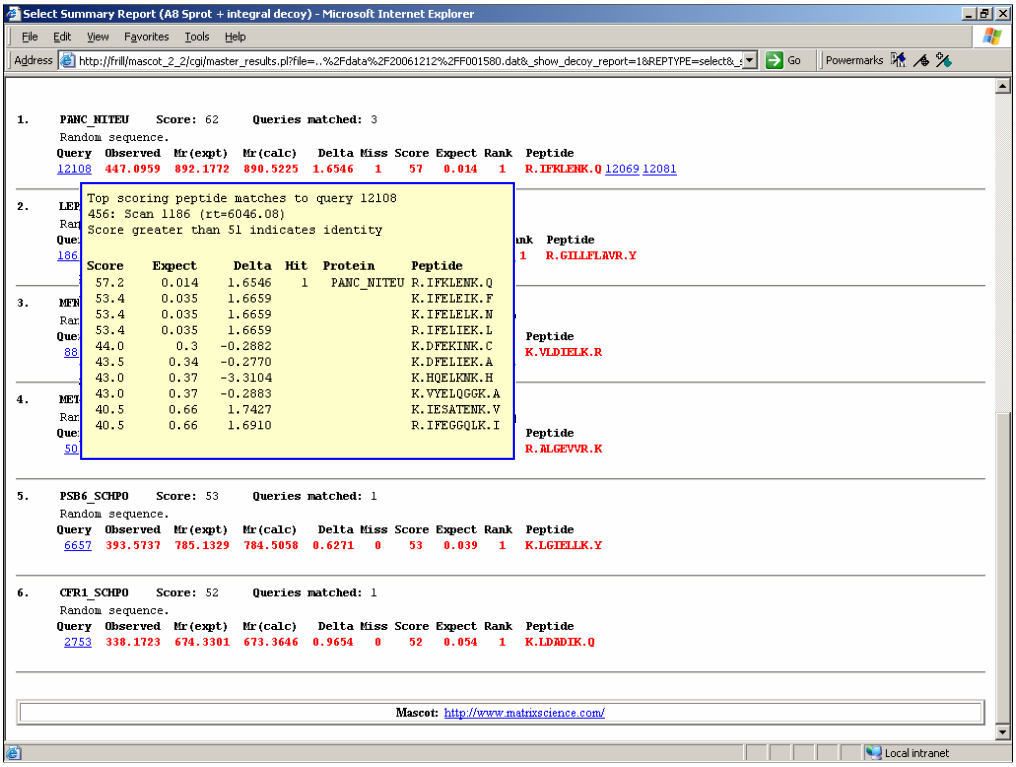


When the search is complete, the statistics for matches to the random sequences, which are effectively sequences from a decoy database, are reported in the result header. If you change the significance threshold, the numbers are recalculated. For example, if we increase the threshold from 5% to 0.5% …



The false discovery rate drops accordingly. Of course, so does the number of true positives

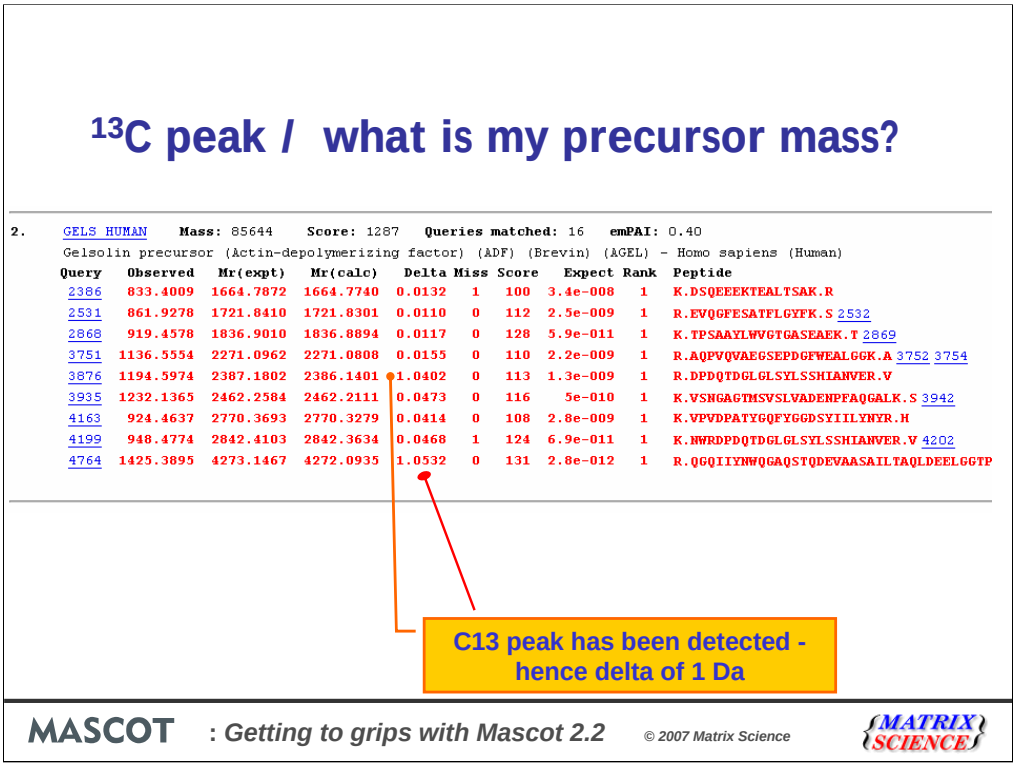
If you click the link here, then you will see the results from searching the randomised database.



Rather than just show you a blank screen, I clicked on the link from the previous slide where we saw that 8 spectra gave matches above significance threshold.

The results from the matches to the randomised sequences are saved in new sections of the results file on the Mascot server. This means that we can view these results in exactly the same way as if we had performed a separate search against a randomised database that we had created manually. For example, we can see the yellow popups and click on the links to see the matches of the random peptide sequences to individual spectra.

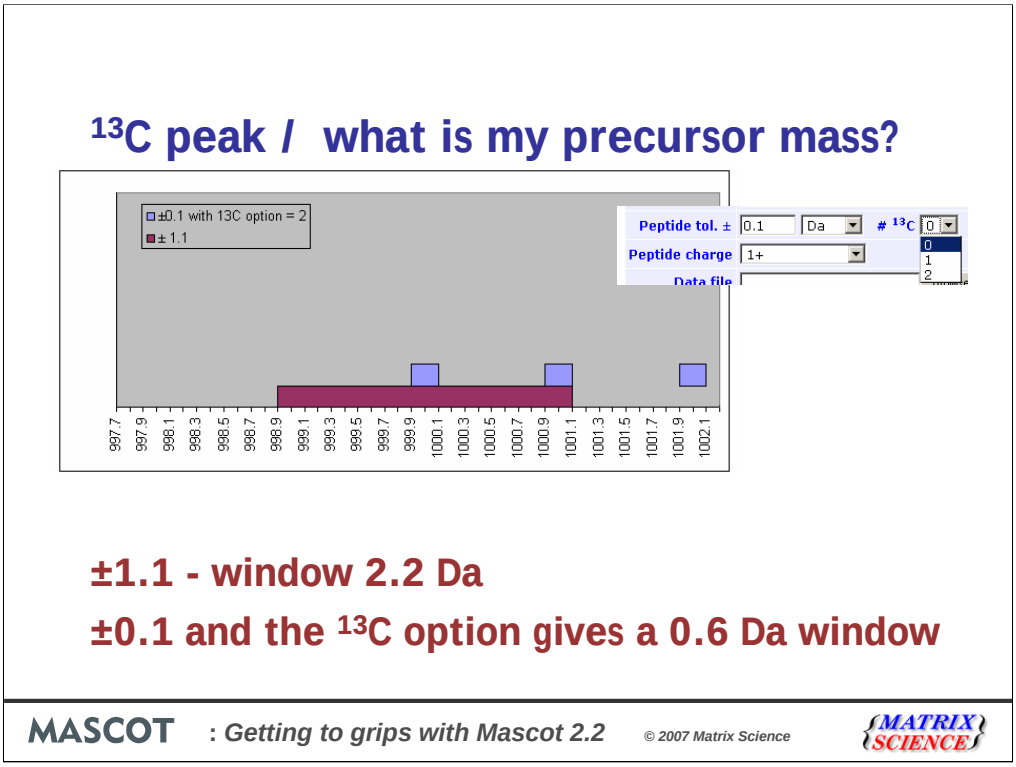
Moving on now to the next change that you'll notice on the search form.



It seems that almost every year we get a new instruments with higher accuracy and faster acquisition times. The greater accuracy means that we can perform database searches with tighter tolerances. This gives us improved specificity and also shorter search times, enabling us to keep up with the faster acquisition rates and growing database sizes.

However, there is a problem… the faster acquisition means that the instrument has less time to accurately determine the precursor mass and will often pick the carbon 13 or even carbon 14 peak rather than the carbon 12 peak. When I searched some pretty accurate data with a wide tolerance, I saw this:

You can hopefully see two peptides with great ions scores, but almost exactly 1 dalton out.

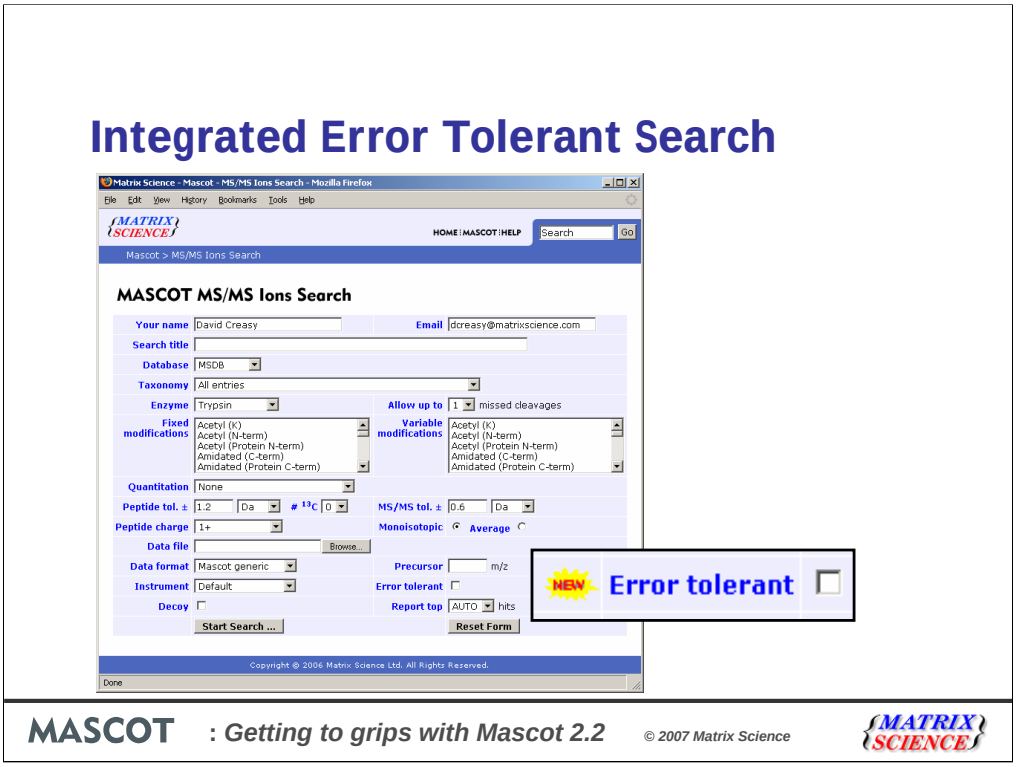


As an example, imagine that we have an m/z value of 1000.

With this type of data, in Mascot 2.1 and earlier, we need to search this with a tolerance of, say +/- 1.1 Daltons. This gives us a window of 2.2 Daltons. And even then, if the peak detection software had chosen the carbon 14 peak, we would still fail to get a match.

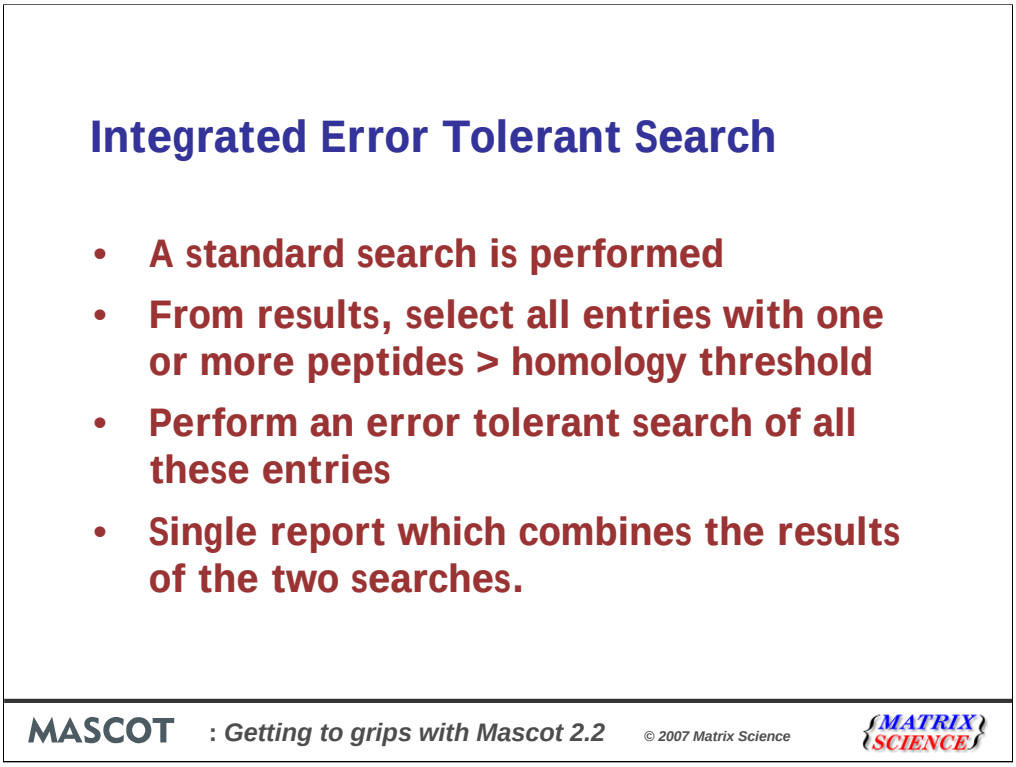
With Mascot 2.2, we have a new option in the search form. With the settings shown here, it will search +/- 0.1 Da around the m/z value of 1000 and +/- 0.1Da around the mass of 1001. If we select '2' rather one in the search form it will also look for peptide matches at 1002 +/- 0.1

If you are using a very high accuracy instrument, note that the precise shifts are the carbon isotope spacings of 1.00335 and 2.00670, rather than 1 and 2.



The next new addition I'll describe in Mascot 2.2 is enabled by clicking on the "Error tolerant" checkbox on the search form.

If you were familiar with earlier versions of Mascot, you would remember that it was possible to perform an error tolerant search by performing a standard search, selecting one or more protein and then repeating the search on those proteins.This was a manual process and many people didn't understand how to perform this.



When this button is checked, a standard, first pass search is performed using the search parameters specified in the form.

From the results of the first pass search, all of the database entries that contain one or more peptide matches with scores at or above the homology threshold, (or identity threshold if there is no homology threshold), are selected for an error tolerant, second pass search.

At the completion of the second pass search, a single report is generated, combining the results from both passes.

Integrated Error Tolerant Search - 2nd stage

- Selected enzyme becomes semi-specific and missed cleavages increased by 1
- Complete list of mods tested serially
- Substitution of residue / base
- If modified and unmodified peptides are within precursor window, discard modified.

For the second stage of the search, the selected enzyme becomes semi-specific, (that is, only one end of a peptide needs to match the cleavage specificity), and the value of the missed cleavage parameter is increased by 1

The complete list of modifications is tested, serially. This is currently about 700 different modifications

For a protein, the set of substitutions that can arise from single base substitutions is tested. For a nucleic acid sequence, all single base insertions, deletions, and substitutions are tested.

Only one of the above is allowed per peptide. That is, an individual peptide can be semi-specific OR have one unsuspected modification OR have one primary sequence mutation.

If the modified and unmodified peptides are both within the precursor mass tolerance window, the modification is rejected. This eliminates modifications that are meaningless given the estimated mass error, like Q->K, in most cases.

Integrated Error Tolerant Search - 1st stage

- Must be fully specific enzyme
- Only 2 modifications can be selected (configurable)
- Cannot be combined with a decoy search
- Cannot be combined with quantitation
- Cannot include error tolerant sequence tag.

The following constraints apply to the standard, first pass search:

Enzyme must be fully specific

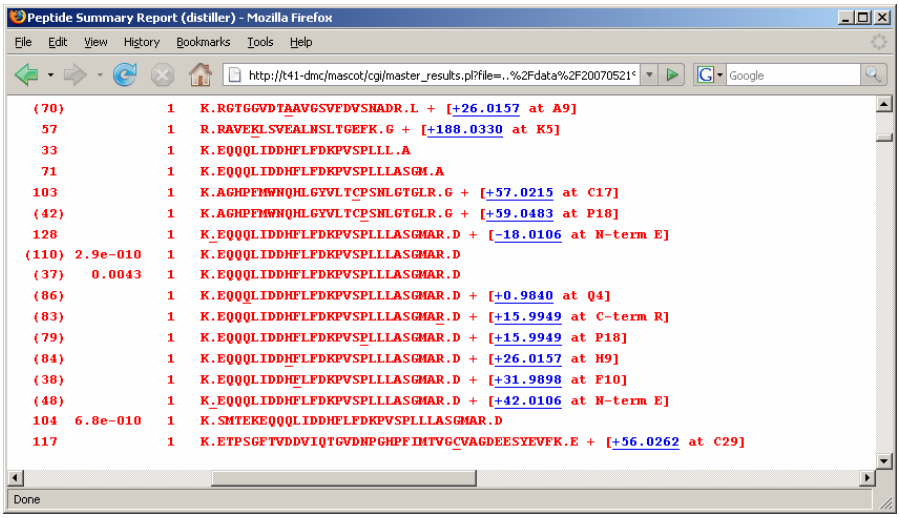
There is a reduced ceiling on the number of variable modifications, (default is 2, but this can be changed globally in mascot.dat or for a user group in Mascot security)

Cannot be combined with an automatic decoy database search

Cannot be combined with quantitation

Search cannot include error tolerant sequence tag

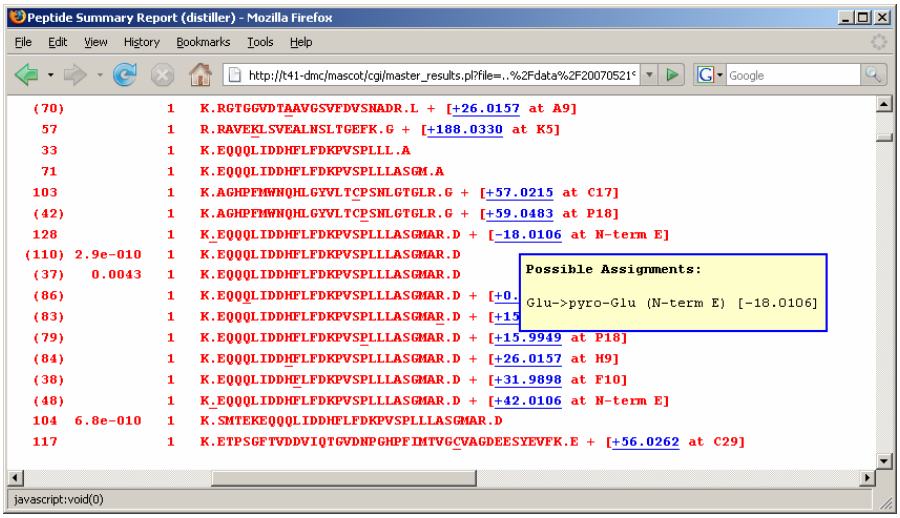
Integrated Error Tolerant Search - Example



This set of potential matches picked out by the error tolerant search shows some interesting examples and potential pitfalls.

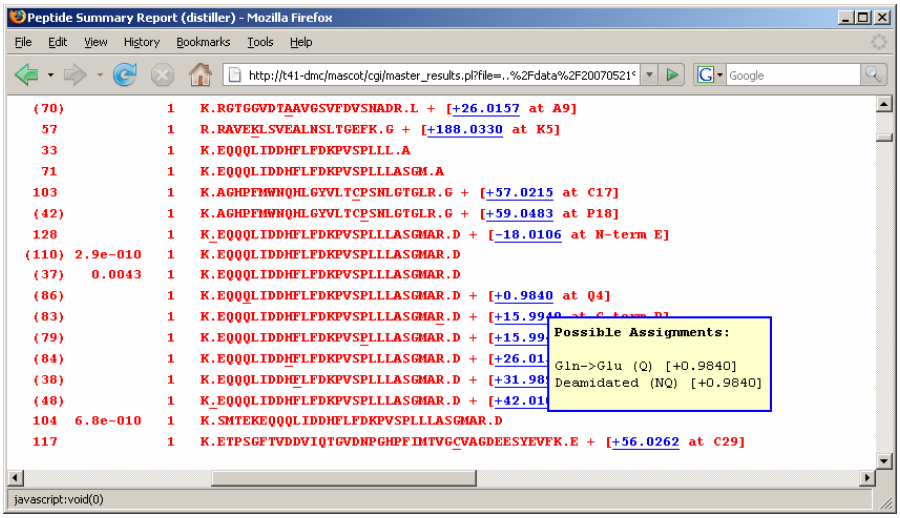
Firstly, I should point out that this set here doesn't increase the sequence coverage because the unmodified peptide is present.

Integrated Error Tolerant Search - Example



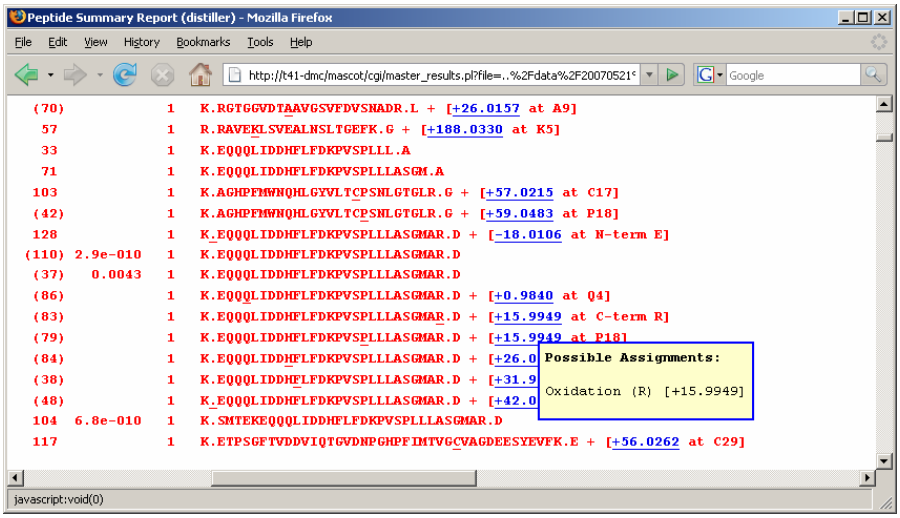
The first delta that it has found is minus 18 and is very likely to be Pyro-Glu as detected by Mascot.

Integrated Error Tolerant Search - Example



In the second case, there are two possible assignments. The substitution is unlikely because there are so many other matches to the ‘correct’ sequence in the database, and hence deamidation is likely to be correct

Integrated Error Tolerant Search - Example

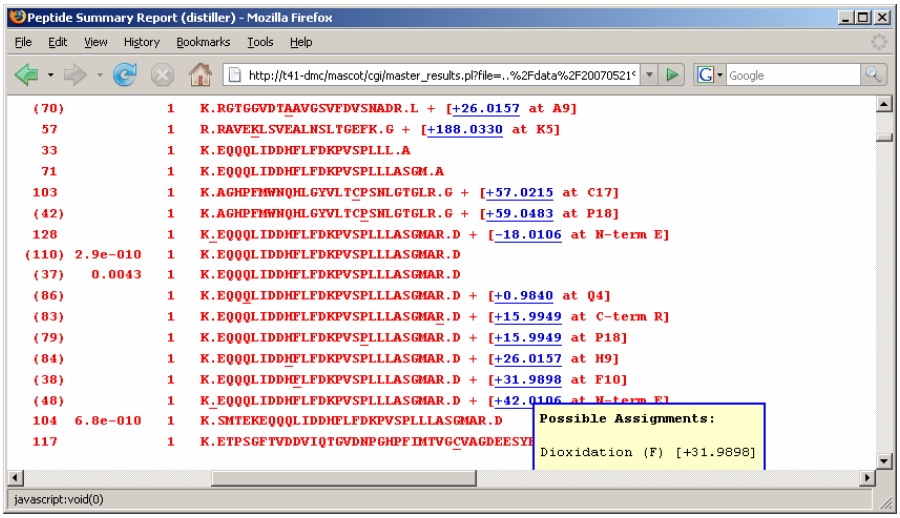


The next example shows oxidation of arginine at the C terminus, which is possible. However, if you look carefully, you will see a methionine just 2 residues away. If we were to scroll to the left and then hover the mouse over the query button, we would see that the scores were very similar for the different oxidation sites. I think in this case most of us would put our money on the methionine being oxidised.

The next peptide match shows an oxidised proline. This time the score for the oxidised methionine is much lower, so maybe it is correct.

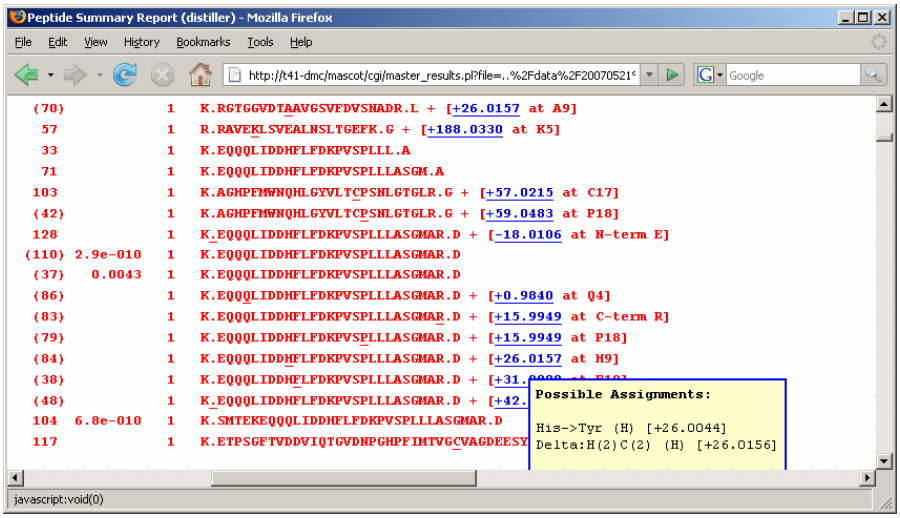
The sample does seem to be very heavily oxidised.

Integrated Error Tolerant Search - Example



As is indicated by the Dioxidation of Phenylalanine.

Integrated Error Tolerant Search - Example



Stepping back to the delta of 26.01 here, we can see that Mascot suggests two possible modifications. The first is a substitution of histidine to tyrosine and the second is a modification called “Delta H(2)C(2)

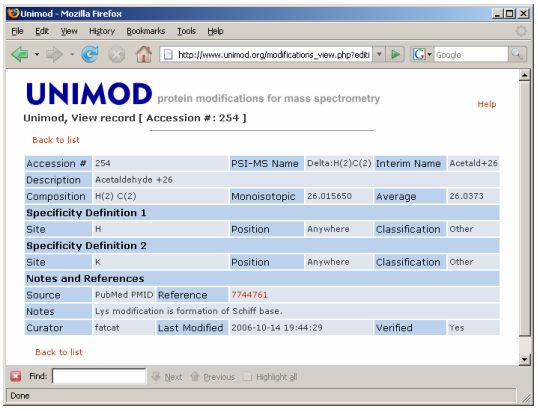
How can we find out which, if either, of these is likely?

Integrated Error Tolerant Search - Example

- **Delta 26.0, 'His->Tyr' or 'Delta:H(2)C(2)'**

Blast search for substitutions

www.unimod.org

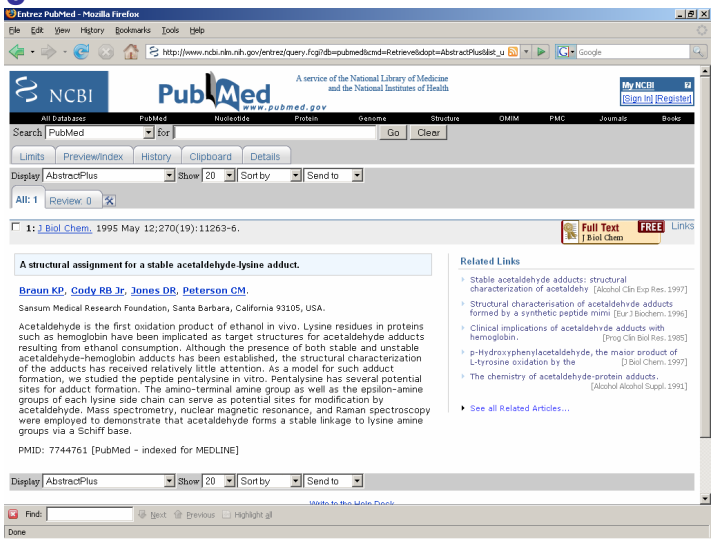


To determine if the substitution is likely, we can do a blast search.In this case, I was unable to find any matches to the substituted peptide

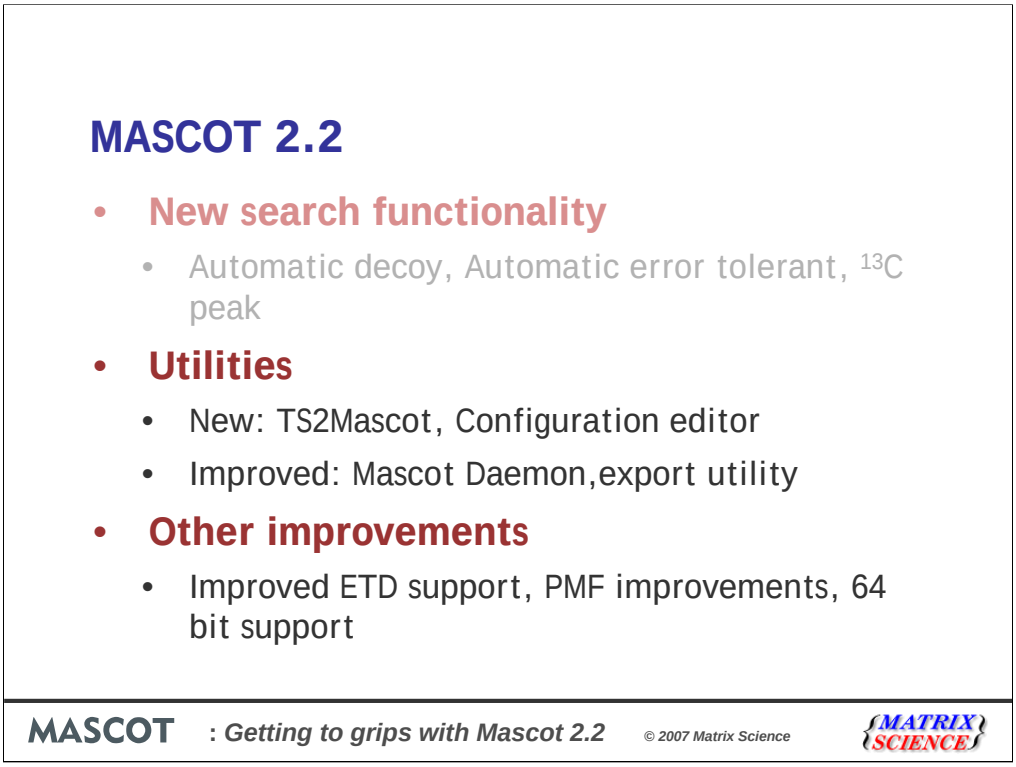
For the modification, we need to go to www.unimod.org and search for delta:H(2)C(2) which shows us this record.

To get further information, we can click on the references link:

Integrated Error Tolerant Search - Example

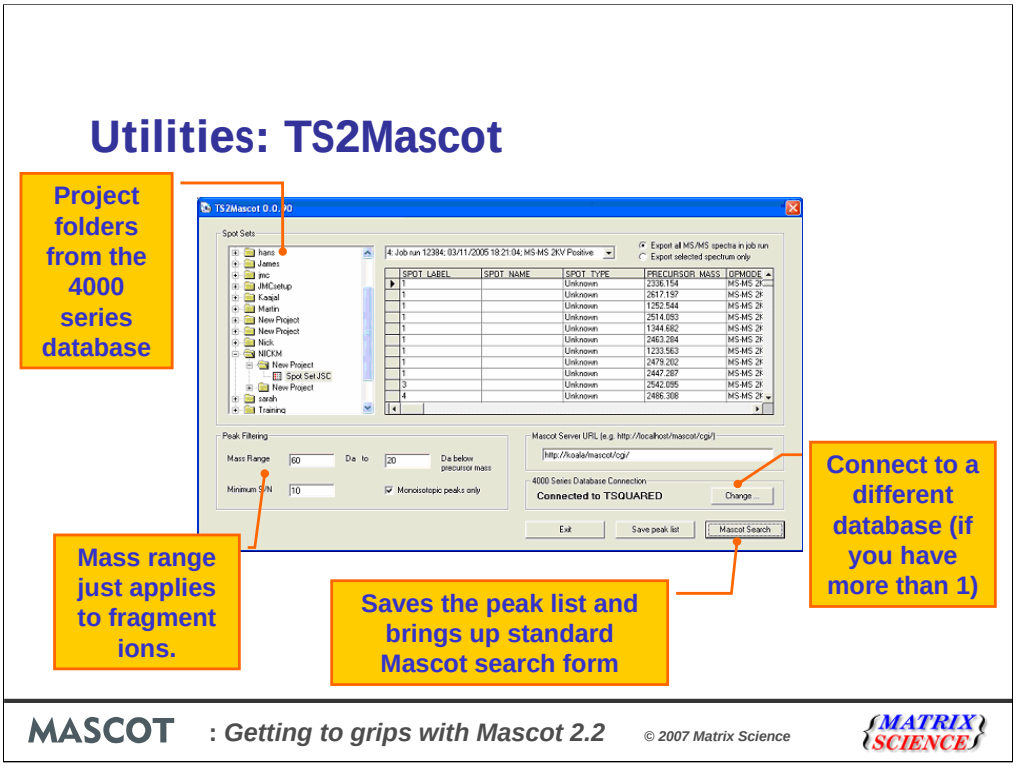


And we can then see if this a likely modification for our heavily oxidised sample (which it does seem to be)



We've seen the major new search functionality - the decoy search, Carbon 13 peak error and the automatic error tolerant search.

Next I'll cover two new utilities - TS2Mascot and the configuration editor plus some improvements to Mascot Daemon and the export utility.



TS2Mascot is a small utility to get peak lists from the 4700/4800 instruments and submit them to Mascot. The utility ensures that correct intensities are generated for iTRAQ reporter ions

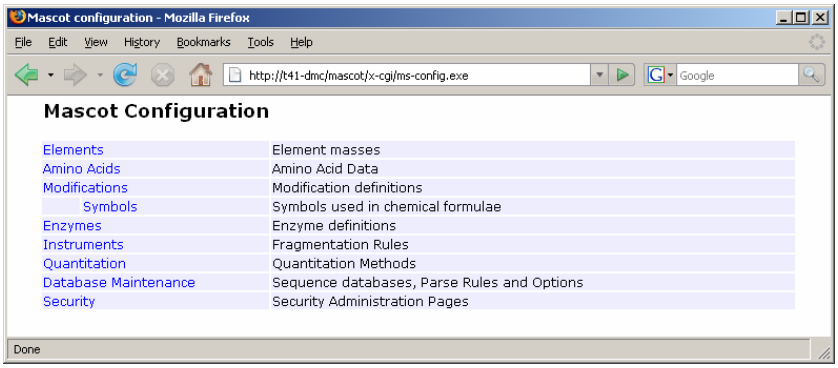
TS2Mascot is very simple to use. It is launched from the Windows Start menu, (Programs; Mascot; TS2Mascot). The folder tree in the Spot Sets frame is populated with project folders from the 4000 series database. If you have more than one 4000 series database, choose Change to connect to a different database. Navigate the folder tree and select a spot set icon. This will populate the adjacent drop down list with the job runs for the spot set. The grid area displays details of individual job run items for the selected job run.

Peak filtering settings are equivalent to those used in Applied Biosystems GPS Explorer. The mass range limits only apply to MS/MS fragment ion peaks. The contents of the Peak Filtering and Mascot Server URL fields are sticky. If you change them, then process a spot set, the new values will be remembered. Save peak list invokes a standard file selection dialog before processing the data. Before choosing Mascot Search, ensure that you have entered a valid URL for a Mascot server. You can enter the URL of the Matrix Science public web site, but remember that this has a limit of 300 spectra in a single search.

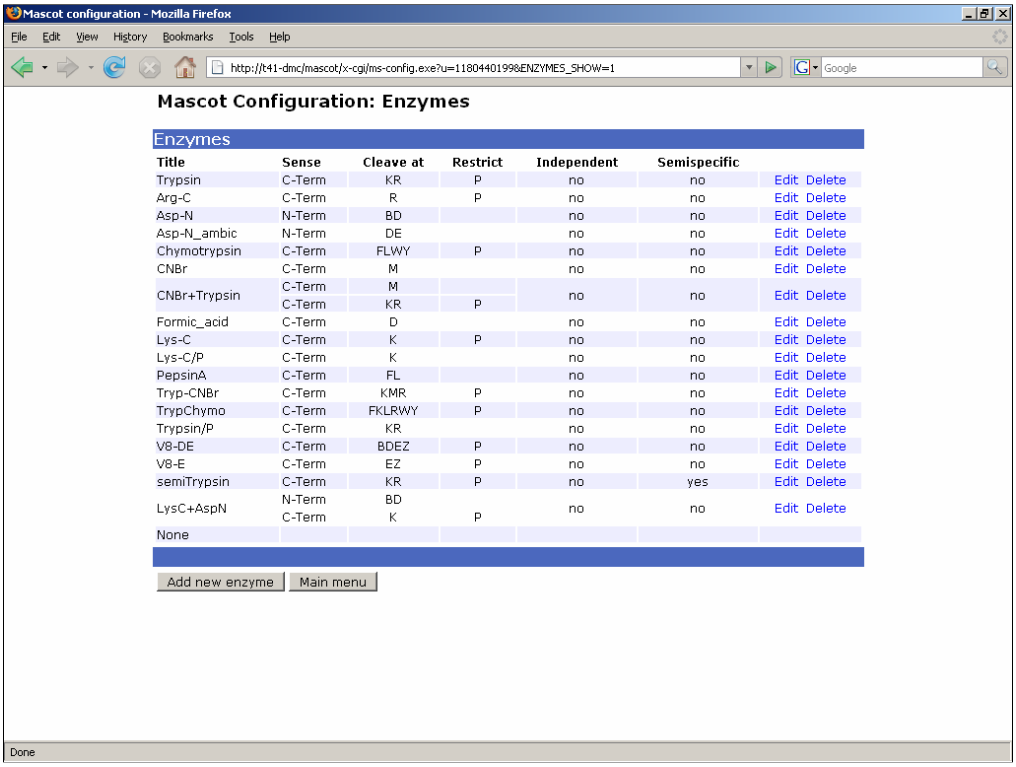
While the peak list is being created, a progress bar is displayed, and all the controls are disabled apart from a Cancel button. Once peak list export is complete, if you have chosen Mascot Search, the default web browser will be launched and the Mascot MS/MS search form displayed. The search title and data file path are filled in automatically. The other settings are the search form defaults, (which can be customised by following a link on the search form selection page).

Searches can be automated by selecting TS2Mascot from within Mascot Daemon

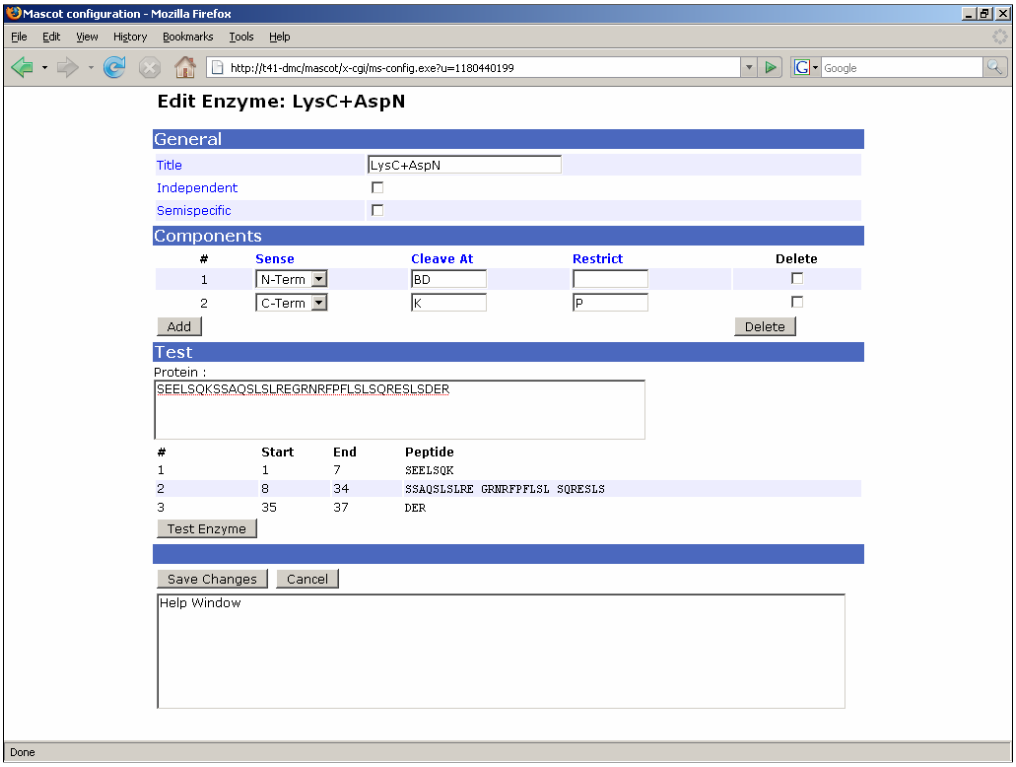
Utilities: New Configuration Editor



John has already shown us the configuration editor in the quantitation talk earlier. The same editor can be used for editing most of the other configuration files. Adding a new modification or enzyme for example, is now relatively straightforward.



For example, to modify the LysC + AspN enzyme, we can just click on the link here



We can see that this is a special case of where a mixture of two enzymes are used. LysC is an C terminus cutter - cutting after the lysine and AspN is in N terminus enzyme, cutting before the Asparagine. The default setup is for the enzymes to be applied serially to the same sample. However, if you apply these enzymes independently to separate aliquots which you then mix then the enzyme needs to be specified differently in Mascot; we can change this ever so easily by checking the check box.

There is also a test sequence to see the effect of the enzymes.

Nearly all of the editing of the configuration files can now be performed remotely from the server using these browser based tools. This is a real advantage if your server is administered by your IT group and you would otherwise have to wait for them to change the files.

Utilities: Mascot Daemon improvements

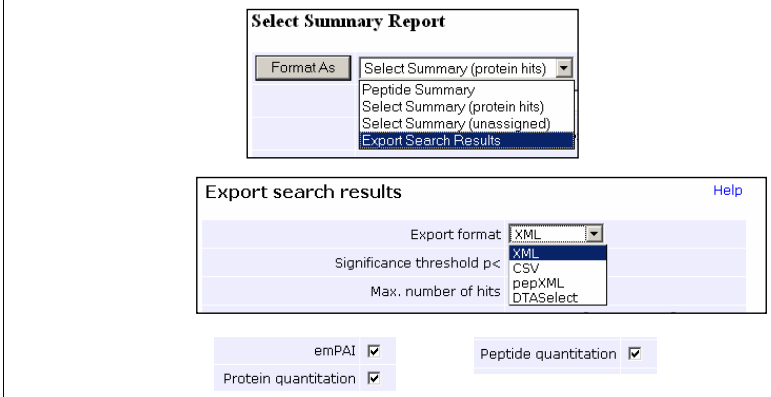
- **Previous limit of 2Gb has been removed**
  - Can be the only way to submit huge searches due to browser limitations.

In earlier versions of Mascot, it wasn't possible to submit searches from Daemon with more than 2Gb of peak list data. This tends to only happen when combining data from multiple MudPIT fractions into a single search, but could happen for a large single data file.

In Mascot Daemon 2.2, this restriction has been removed, although some web servers may still have limits. Many browsers can't be used to submit files larger than 2Gb, so if you have very large data files, using Mascot Daemon may be your easiest solution.

Utilities: Export utility improvements

- **Select from the main results page:**



We've also made a number of improvements to the export utility. In case you've not used it before, select 'Export Search Results' from the main results page and click the Format As button.

On the export search results page, you can select from a number of formats. The CSV format is useful if you want to load the results into Excel

We've added a number of extra options, including of course the new features such as the decoy results and the quantitation results. We'll be adding support for the PSI AnalysisXML format as soon as it is ready, but please let us know if there is anything missing or any other new export format you'd like to see.

In general, it is better and safer to use the export utility rather than trying to process the .dat files as these have become significantly more complex with the addition af decoy and error tolerant searches. Also, the results files don't (and can't) for example include iTRAQ quantitation results, but the export file does include this information in an easy to use format.

MASCOT 2.2

- **New search functionality**
  - Automatic decoy, Automatic error tolerant, ^{13}C peak
- **Utilities**
  - New: TS2Mascot, Configuration editor
  - Improved: Mascot Daemon,export utility
- **Other improvements**
  - Improved ETD support, PMF improvements, 64 bit support

Onto the last section where I've picked out three items to describe.

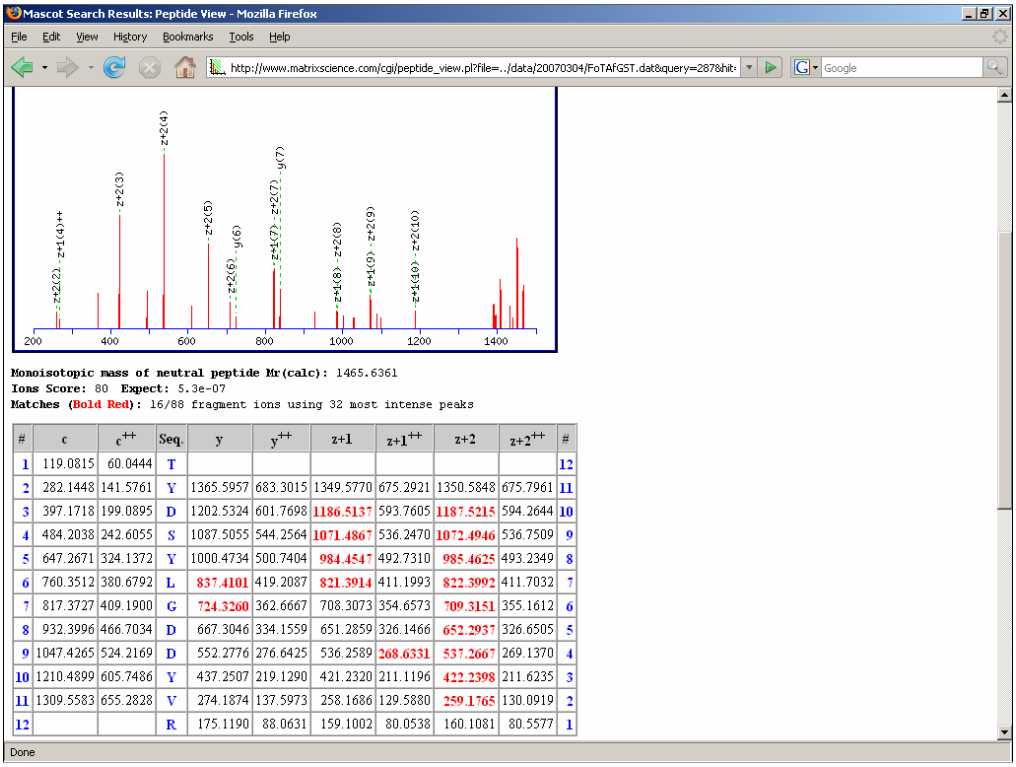
Improved support for ETD

- Support for ECD (and hence ETD) since 2002
- Added z+2h
- New instrument is called ETD-TRAP but has same series as FTMS-ECD:
  c, y, z+h and z+2h

We have had support for ECD and hence also ETD in Mascot since 2002.

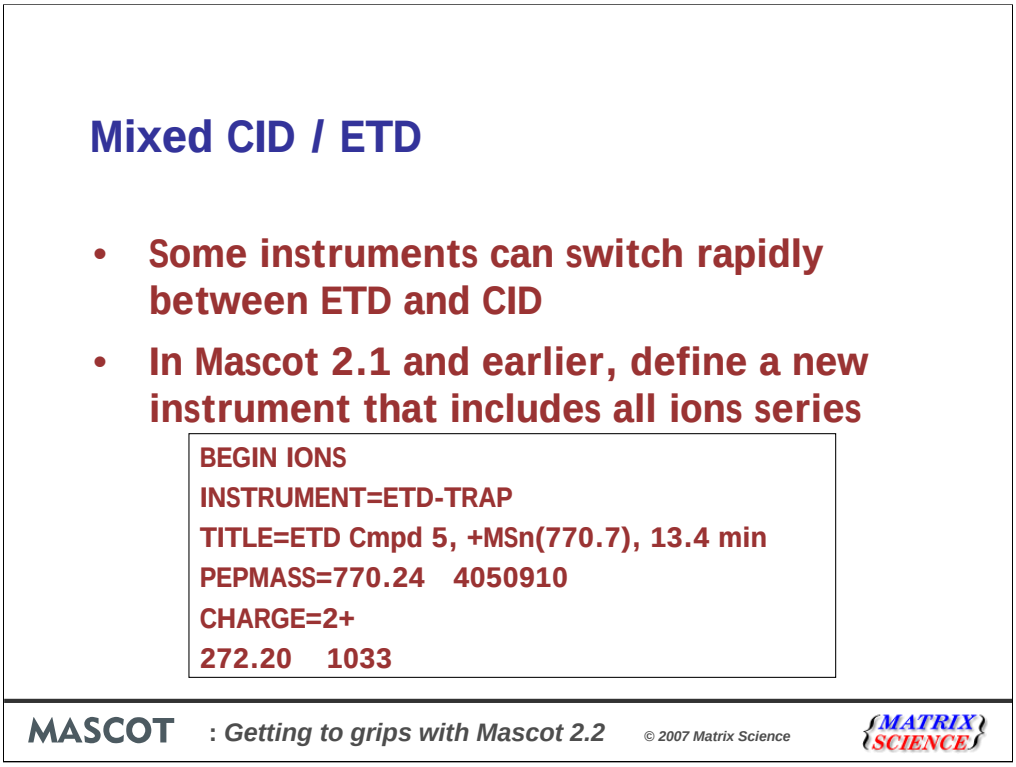
However, we have improved the support for these types of instrument in Mascot 2.2 in two different ways.

Firstly, we've added a new ions series, the Z+2 series, which are seen in some instruments. For convenience, we've also added a new instrument called ETD-TRAP, but this includes the exact same series as the original FTMS-ECD instrument, that is c, y. z+h and z+2h



#	c	c^{++}	Seq.	y	y^{++}	z+1	$z+1^{++}$	z+2	$z+2^{++}$	#
1	119.0815	60.0444	T							12
2	282.1448	141.5761	Y	1365.5957	683.3015	1349.5770	675.2921	1350.5848	675.7961	11
3	397.1718	199.0895	D	1202.5324	601.7698	**1186.5137**	593.7605	**1187.5215**	594.2644	10
4	484.2038	242.6055	S	1087.5055	544.2564	**1071.4867**	536.2470	**1072.4946**	536.7509	9
5	647.2671	324.1372	Y	1000.4734	500.7404	**984.4547**	492.7310	**985.4625**	493.2349	8
6	760.3512	380.6792	L	**837.4101**	419.2087	**821.3914**	411.1993	**822.3992**	411.7032	7
7	817.3727	409.1900	G	**724.3260**	362.6667	708.3073	354.6573	**709.3151**	355.1612	6
8	932.3996	466.7034	D	667.3046	334.1559	651.2859	326.1466	**652.2937**	326.6505	5
9	1047.4265	524.2169	D	552.2776	276.6425	536.2589	**268.6331**	**537.2667**	269.1370	4
10	1210.4899	605.7486	Y	437.2507	219.1290	421.2320	211.1196	**422.2398**	211.6235	3
11	1309.5583	655.2828	V	274.1874	137.5973	258.1686	129.5880	**259.1765**	130.0919	2
12			R	175.1190	88.0631	159.1002	80.0538	160.1081	80.5577	1

Here we have a case where an ECD spectrum with an excellent z+2h series which would have only got a weak match in previous releases of Mascot.



The other improvement is to provide support for the new instruments that can switch rapidly between ETD and CID mode. In Mascot 2.2, the best way to search this data is to create a new instrument definition that includes the ions series of both types of data. This obviously reduces specificity somewhat.

In Mascot 2.2, it is possible to specify the instrument type for each ms-ms spectrum as shown here. It's also possible to specify variable modifications at this level too.

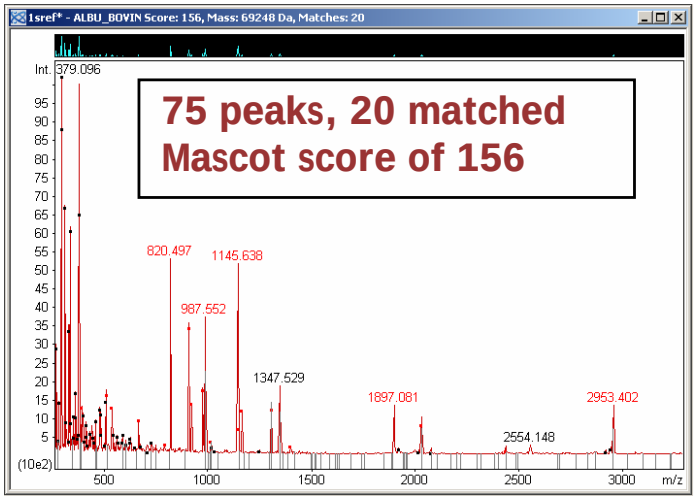
PMF Improvements

- Peak detection for MS searches still an issue for some systems
- Mass values are now selected, iteratively, by peak intensity
- This will give better matches where there are a lot of weak, noise peaks.

When we look at why PMF searches are failing, one of the reasons is still that peak detection can be poor. We see cases with peak lists consisting of many hundreds of peaks. Since a tryptic digest of an “average” protein (30 kDa) should produce of the order of 50 peptide peaks, these are either complex mixtures or most of the peaks must be noise.

In Mascot 2.2, we now select peaks iteratively by peak intensity which will give better matches where there are a lot of weak noise peaks.

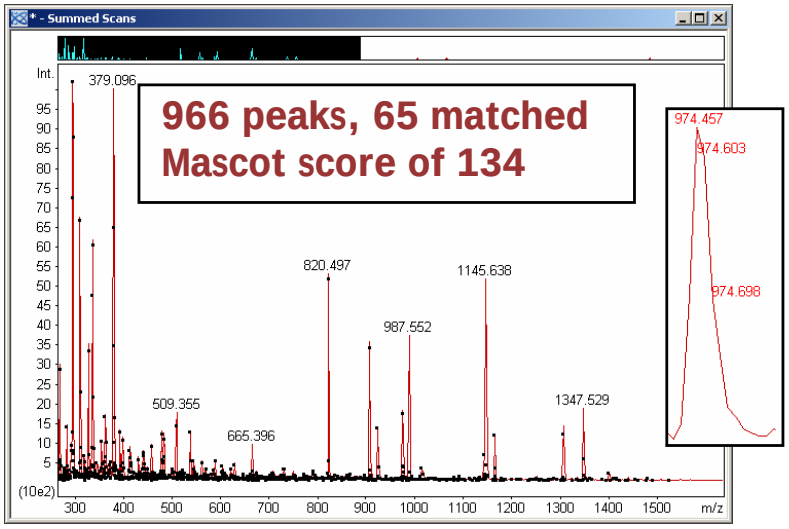
PMF Improvements



MASCOT : *Getting to grips with Mascot 2.2* MATRIX SCIENCE

As an example, I've taken a high quality MALDI-TOF spectrum and performed high quality peak detection. This produces a peak list of 75 peaks, which is a little less than we would expect for this 60k Da protein. 20 of the peaks get a match and it gets a score of 156

PMF Improvements



I've now performed some really bad peak detection - each of the little black dots is a peak. We now have 966 peaks, but most of these are low intensity. With Mascot 2.1, you would only ever get random matches with this number of peaks. With Mascot 2.2, we get a score of 134 which is slightly lower than 156 when we have better peak detection - but at least we found the correct match with a significant score.

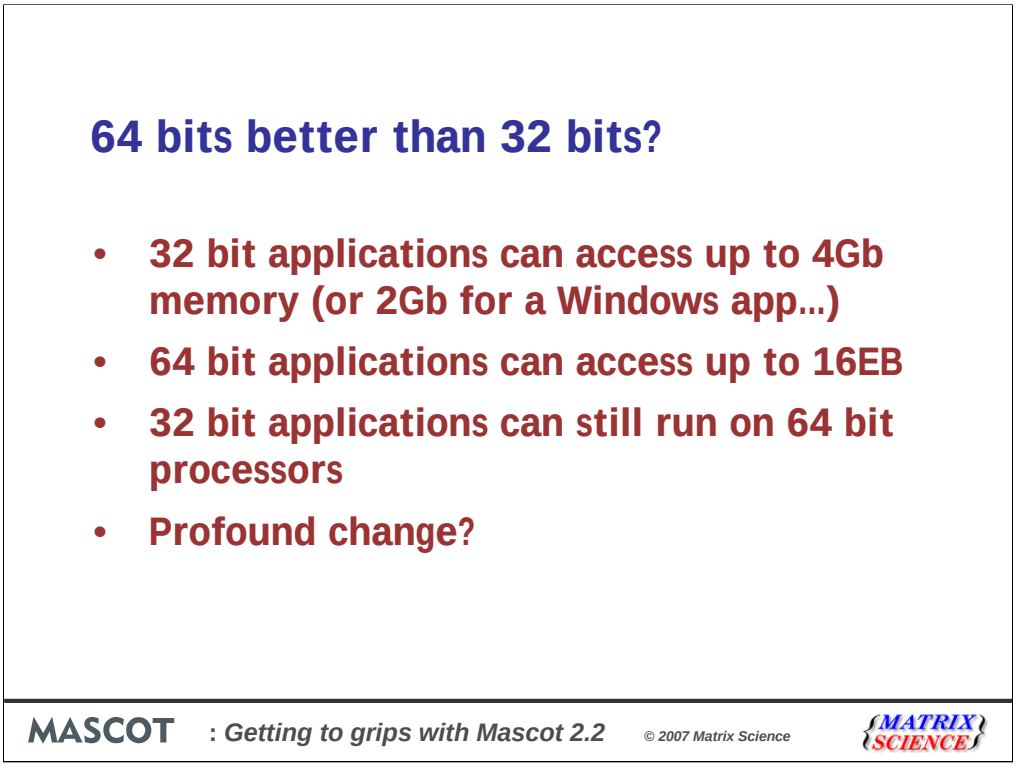
You may have noticed that more peaks appeared to match. Did we get better coverage? Yes, but only slightly. In this case, many single peaks actually had several mass values that matched to the same peptide, so this accounts for most of the increase in number of matches.

This is not a substitute for good peak detection. If your sample gives a borderline match with good peak detection, then it probably won't give a significant match with poor peak detection, even in Mascot 2.2

64 bits better than 32 bits?

"This is a time of profound change in our industry, the move from 32-bit to 64-bit computing offers a dramatic improvement in performance and reliability." - *Bill Gates, April 2005*

There has been considerable 'hype' in the press in the last couple of years about the benefit of 64 bit computing. Statements like this from Bill Gates have certainly fuelled this - but what is the reality for running Mascot searches?



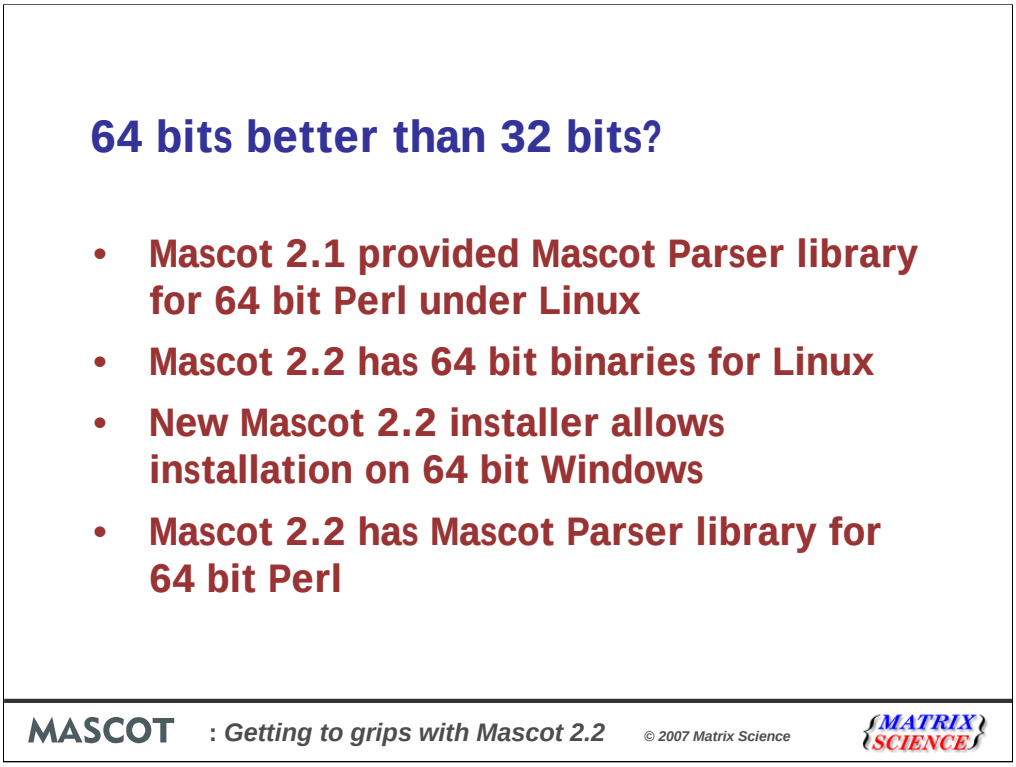
First and foremost, the amount of memory that can be accessed by an application is dramatically increased. A 32 bit application can in theory access up to 4GB of memory - 4Gb is 2 to the power 32. However, it turns out that the limit is 2GB for most Windows application and 3.5GB for Linux applications.

A 64 bit application can access a massive 16 Exabytes - or more correctly Exbibytes. That's about 10^18 bytes. Of course (and please don't quote me on this) nobody will ever want to access that much memory.

For all the operating systems that Matrix Science currently support, it is possible to still run 32 bit applications on the 64 bit version of the operating system.

Bill Gate's stated that we would expect massive improvements in reliability. What he was describing was potential improvements in operating system reliability due to features that are enabled when Intel processors run in 64 bit mode.

He also described the move to 64 bits as a profound change? I don't think so...



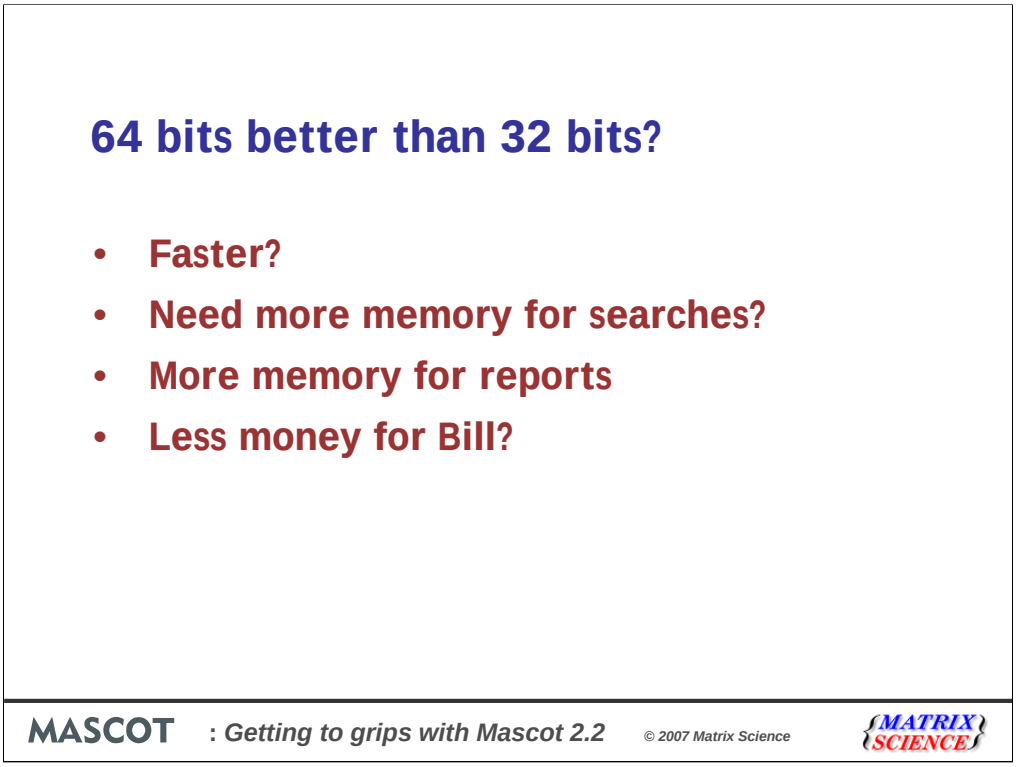
So, what have we done in Mascot to embrace these profound changes in the computing industry that I seem to be rather cynical about.

Firstly, for AIX, Solaris and SGI IRIX, we have provided 64 bit binaries for many years.

In Mascot 2.1, 64 bit versions of the Mascot Parser library were available for Linux, but all the other binaries were 32 bit

In Mascot 2.2, we provide 32 and 64 bit binaries

For Windows, it wasn't possible to install Mascot 2.1 and earlier on 64 bit Windows due to some limitations in the Installer software that we were using. With Mascot 2.2, we have a brand new installer which means that Mascot can be installed on 64 bit Windows. We've also included full support for 64 bit Perl. However, the other binaries are still all 32 bit.



What are the benefits of 64 bits for Mascot?

Is it faster? The 64 bit binaries on Linux can run some searches up to 18% faster. Other searches run at the same speed.

Do we need more memory for searches? Not really because the searches get split into chunks and therefore there is no advantage here.

But yes, it is possible for the scripts to run out of memory when generating huge reports, and this is why we need to support 64 bit Perl and why we made this the first priority for 64 bit Windows. If you are installing Mascot on a 64 bit system (any platform) and running very large searches, it is best to try and install 64 bit Perl which will in turn cause Mascot to use the 64 bit Mascot Parser.

Another reason for choosing 64 bit Windows is that it involves giving less money to Bill.

64 bits better than 32 bits?

Operating System	Max CPU	Max RAM (GB)	Cost
2000 Professional	2	4	
2000 Server	4	4	
2000 Advanced Server	8	8	
2000 Data Center	32	32	
XP Professional	2	4	£227.00
2003 Web Edition	2	2	
2003 Standard Edition	4	4	
2003 Enterprise Edition	8	32	£2974.00
2003 Data Center Edition	32	64	?
XP Professional - 64 bit edition	2	16	£227.00
2003 Enterprise Edition - 64 bit	8	2048	£2974.00
2003 Data Center Edition - 64 bit	64	2048	?

For marketing, rather than technical reasons Microsoft decided to limit the amount of memory that recent operating systems could access. For example, if you put 8Gb or RAM in a computer with XP, then you will only be able to use 4Gb of that, and you might as well throw the rest away. If you wanted to have more than 4Gb with Windows 2003, you needed to invest in the Enterprise Edition which costs about £3000 in UK money.

I guess they felt a little uncomfortable with these limits on 64 bit systems, so they increased them to more sensible values. If you want to put 8Gb of RAM in a PC, then 64 bit XP might be a suitable option.

MASCOT 2.2

- **New search functionality**
  - Automatic decoy, Automatic error tolerant, ^{13}C peak
- **Utilities**
  - New TS2Mascot, Configuration editor
  - Improved Mascot Daemon,export utility
- **Other improvements**
  - Improved ETD support, PMF improvements, 64 bit support

As you've seen, Mascot 2.2 included a lot of extra functionality apart from the quantitation support. I've shown the automatic decoy and error tolerant searches.

We've looked at a couple of new utilities and finally I've just highlighted 3 other changes to Mascot 2.2

If you are using an earlier version of Mascot in house, I'd strongly recommend that you try Mascot 2.2 on our public web site - and if you like it, please don't hesitate to ask us about getting an upgrade for your in house copy.