Sequence Queries
Three ways to use mass spectrometry data for protein identification

1. **Peptide Mass Fingerprint**
   A set of peptide molecular masses from an enzyme digest of a protein

2. **Sequence Query**
   Mass values combined with amino acid sequence or composition data

3. **MS/MS Ions Search**
   Uninterpreted MS/MS data from a single peptide or from a complete LC-MS/MS run

You will remember from the introduction, that sequence queries are searches where mass information is combined with amino acid sequence or composition information.
The best known example is a sequence tag search, where a few residues of amino acid sequence are interpreted from the MS/MS spectrum.
You can enter sequence tags, and other types of query, into the sequence query form. Remember that all the search parameters, including enzyme specificity, modifications, and precursor charge, still apply to this type of search.

Mascot will look for a match between the tag and the ion series specified by the instrument type. Note that Mascot will only try to match the tag against ion series formed by a single backbone cleavage, and maybe a neutral loss, like $y$ or $b^*$ or $y^{++}$. It won’t try to match against side chain cleavage fragments, like $d$, $v$, $w$ or internal fragments.
Standard sequence tag

Keyword is tag

What’s (probably) wrong with this tag?

1890.2 tag(1004.1, LSADTG, 1548.5)

Very unlikely that you would be able to call L from a spectrum. Should be

1890.2 tag(1004.1, [I|L]SADTG, 1548.5)

Ambiguity is OK as long as it is explicitly represented

877.4 tag(376.2, [I|L][Q|K][I|L], 730.2)
(VG = R, F = MetOx)

Unless you have high energy fragmentation, and are able to distinguish L from I by side chain cleavage fragments, then this tag is wrong. It should be I or L.

Ambiguity in a tag is fine as long as it is recognised and spelt out. Most times, you won’t know whether a residue is Q or K. F is almost identical to oxidised M. If the peaks are weak, are you sure you have a mass difference of R, or could it be VG and the intermediate peak is missing?
Error tolerant sequence tag

Keyword is etag
Peptide in database is
GVQVETISP GDGR, MH+ = 1314.7
b ion series tag called from TISP should be
1314.7 tag (614.3, T[I|L]SP, 911.5)
But, if unknown modification or SNP increases mass
by 100 Da, mass values would become
N-term side: 1414.7 etag (714.3, T[I|L]SP, 1011.5)
C-term side: 1414.7 etag (614.3, T[I|L]SP, 911.5)

If the sequence is in the database, it is easier and safer to perform an MS/MS search of the peak list. In this sense, the standard sequence tag is obsolete.

The error tolerant tag, which can find a match when there is an unsuspected modification or a small difference in the sequence, is very powerful and very useful.

Imagine we had an unmodified peptide of MH+ 1314.7 and we interpreted a tag of TISP in the b+ series between peaks at 614.3 and 911.5.

What happens if there is a modification or SNP that increases mass by 100 Da?
If the mod is on the N-term side of the tag, all the masses shift up by 100. However, if it is on the C-term side, only the peptide mass changes.
If the tag was in the y ion series, the reverse would be observed
Error tolerant sequence tag

Peptide mass is allowed to change by $\Delta m$
- EITHER both fragment ion masses unchanged
- OR both fragment ion masses shift by $\Delta m$

etags have low specificity
- Use reasonable peptide mass tolerance
- Must select an enzyme

The error tolerant tag allows for this. In effect, it allows the peptide mass to vary and allows the tag to float. However, the tag must stay attached to one end or the other. Either both fragment ion masses are unchanged or both fragment ion masses shift by the same amount as the precursor.

This causes a huge loss of specificity, so we cannot allow etag searches with very wide peptide mass tolerance (> 1% or > 10 Da) or with no enzyme specificity. The enzyme specificity in an etag search is never fully specific, in any case, because one end of the peptide can just extend until it finds a cleavage point.
Tags can be entered with the high mass fragment on the left or the right. These two tags are identical.

Mascot allows multiple tags in a single query. That is, you can call multiple tags from a single MS/MS spectrum. Tags are scored probabilistically. If one tag is wrong, you can still get a good match from the tags that are correct.

If one tag in a query is an etag than all the tags for that query are treated as etags, (not all tags in the search, just in the query)

Finally, you cannot mix ions qualifiers with tag or etag qualifiers. It would just be too complicated.
A lot of people call tags using a calculator and a table of mass values. An alternative is to use Mascot Distiller. Here is a short movie to illustrate:

- Maximise the window
- Choose a likely looking peak, such as 987.384
- Right click to start a tag
- Click on any arrow to extend the tag
- In general, I’ll just go for the biggest peak
- Stop when it starts to look tricky
- Here’s the tag
- Do a Mascot search of the peak list to see what the answer should have been. Here’s one I prepared earlier.
- Whoops! Got it wrong, should have been GE, not W. I’ll stick to the day job.
Alternatively, you can automate the process entirely by using the de novo algorithm. Here’s a nice spectrum where the Mascot database search has failed to find a match. If we right click the peak list and choose de novo …
We get a reasonably high scoring solution, but with a fair amount of uncertainty
Right click the solution and choose Mascot search from the context menu. Note that we have already toggled the tag type to error tolerant.
Distiller populates the query field with the tags taken from the non-ambiguous parts of the de novo solution. We submit the search …
And back comes the result. Note that the results from this most recent search have replaced the original database search. You can switch back to the previous results by selecting them on the searches tab.

This match looks promising. If we right click and choose to view the full Mascot report in a browser …
We can see a good illustration of my earlier point about the enzyme being almost semi-specific in an etag search. The peptide can just extend until it finds a cleavage point and then hypothesise a modification that causes a loss of mass to bring the peptide mass back into line.

If we click on the hyperlink to see the peptide view …
The match was obtained by placing a modification delta of +57 Da on the N-term residue. This is almost certainly carbamidomethylation, which can derivatise amino groups if the conditions aren’t right. This was why the original database search failed to get a match and this is why the error tolerant tag is so useful.
## Search strategy

1. **Standard Mascot search returns the easy matches**
2. **Error tolerant search returns additional matches, but only for proteins already identified**
3. **De novo occasionally returns additional full-length peptide sequences that were not in the database**
4. **More often, de novo returns partial / ambiguous peptide sequences**
   - No real reason to expect additional matches from a tag search
   - Use etag search to find matches to isolated peptides that have a SNP or unsuspected modification
   - Blast or MS-Blast if there is a good stretch of clean sequence

If you want to get as many identifications as possible, as efficiently as possible, you might come up with a strategy similar to this.
Besides tag and etag, Mascot supports a number of other sequence qualifiers. One of these is seq()

Note that seq() is a filter. It must be correct or there will be no match
The other important one is comp(). This would be useful in an ICAT search. Note that comp() is a filter. It must be correct or there will be no match.
As always, there is more information in the Mascot help pages. These references are a good starting point if you are interested in learning more about the potential of combining mass and sequence information.