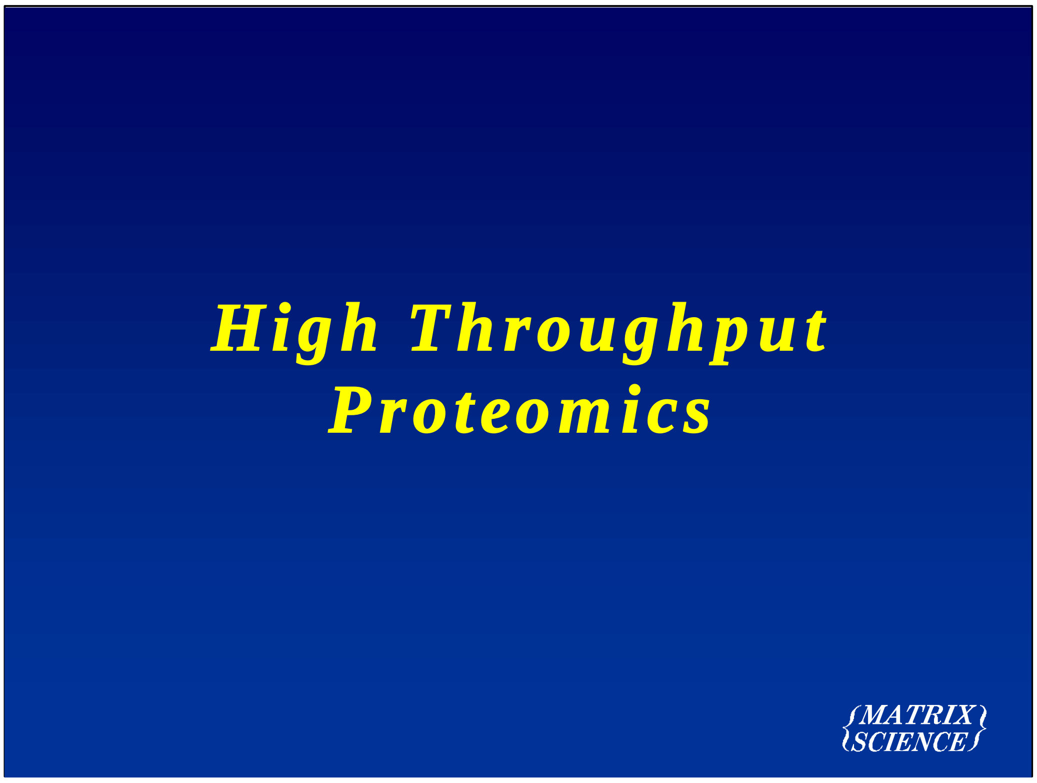
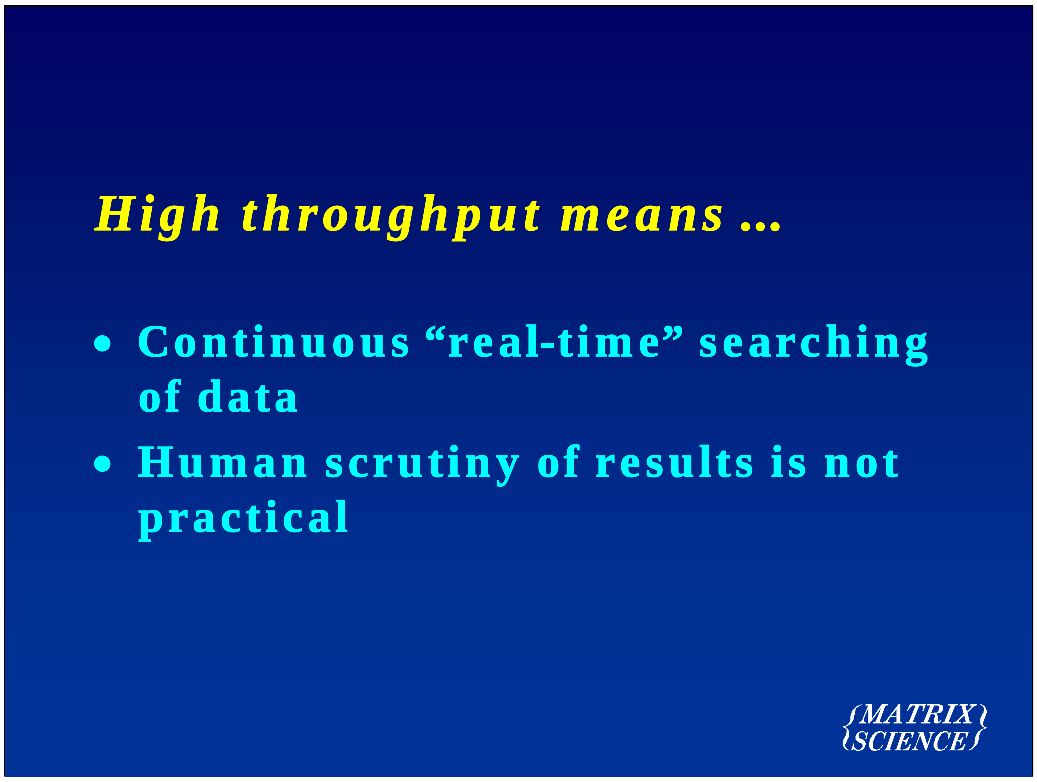


High Throughput Proteomics

MATRIX SCIENCE



High throughput means …

- Continuous “real-time” searching of data
- Human scrutiny of results is not practical

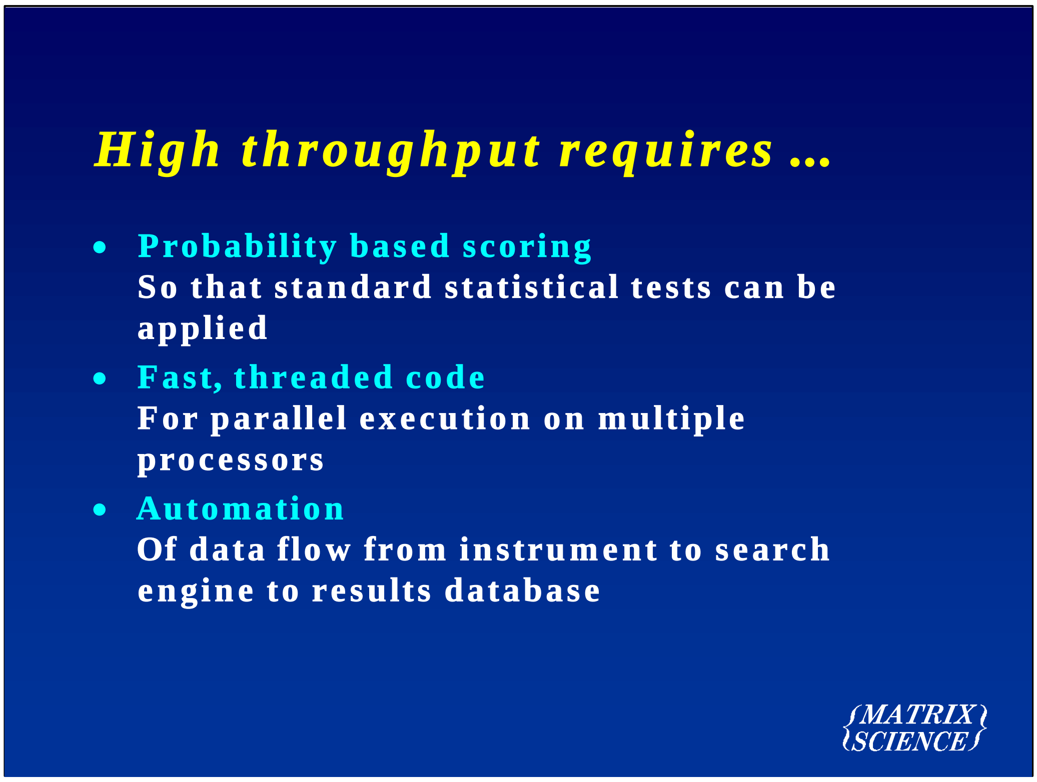
MATRIX SCIENCE

High throughput doesn’t necessarily mean large scale. We would characterise high throughput by two conditions:

First, that the data flow is continuous and must be searched at the same rate as it is produced … you can’t allow a backlog to build up because the backlog will never be cleared.

Second, the data rate is such that it is not practical for every result to be verified by a human operator … you have to trust the search software.

Even a single instrument could meet these conditions. For example, an LC-MS/MS system doing a a separation every 30 minutes, each of which generates a couple of hundred MS/MS spectra, would be classed as high throughput and require the same fundamental approach as a very large scale project.



High throughput requires …

- **Probability based scoring**
  So that standard statistical tests can be applied
- **Fast, threaded code**
  For parallel execution on multiple processors
- **Automation**
  Of data flow from instrument to search engine to results database

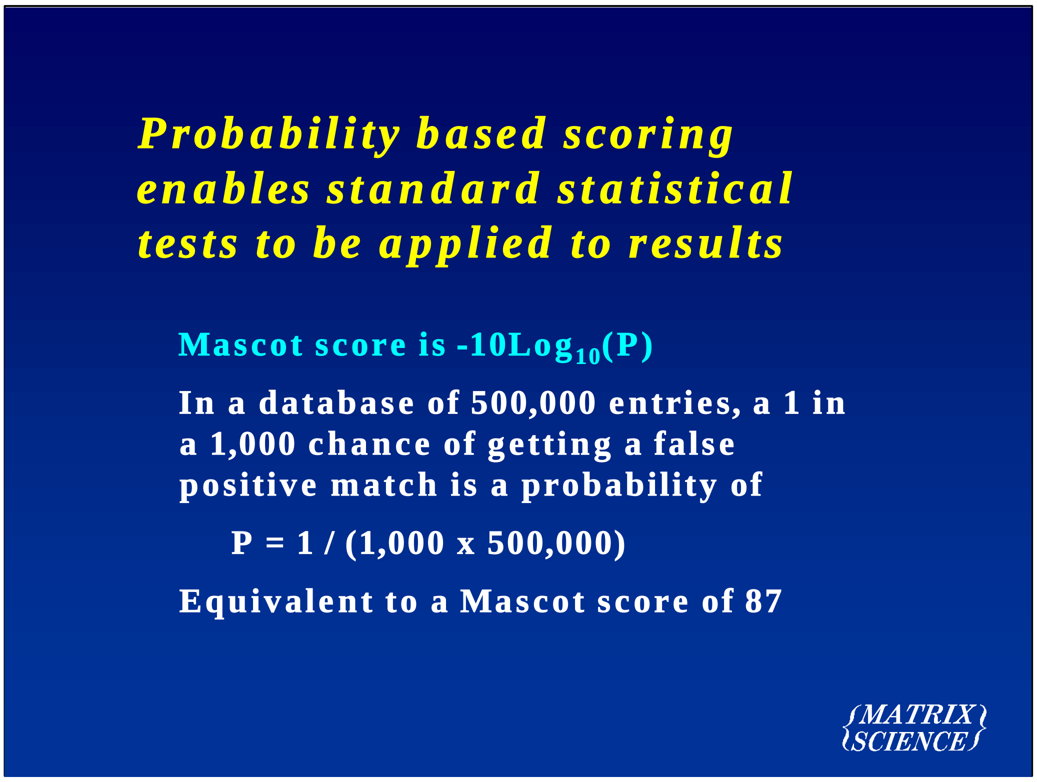
In order to implement a high throughput system, we require:

A scoring scheme which allows simple rule-based decision making. If a human being is not going to study the search results, then the mechanism for deciding whether an identification is safe or not has to be kept as simple as possible. Our approach is to use strict probability based scoring, so that standard statistical confidence tests can be applied.

Systematic protein identification will involve searching large databases, such as a comprehensive non-redundant protein database or dbEST or a complete genome. It also means complex searches, involving multiple variable (that is, non-quantitative) modifications, relaxed enzyme specificity or maybe no enzyme specificity at all. In general, such large searches cannot be performed in real-time using a single processor. It becomes essential to parallelise the process by using multiple processors.

Finally, the flow of data from instrument to search engine to results database has to be automated. Even if human operators are willing to perform endless and tedious data transfer operations manually, their error rate would not be acceptable.

So the three requirements are: scoring, speed and automation.



Probability based scoring enables standard statistical tests to be applied to results

Mascot score is $-10\mathrm{Log}_{10}(P)$

In a database of 500,000 entries, a 1 in a 1,000 chance of getting a false positive match is a probability of

$$P = 1 / (1{,}000 \times 500{,}000)$$

Equivalent to a Mascot score of 87

MATRIX SCIENCE

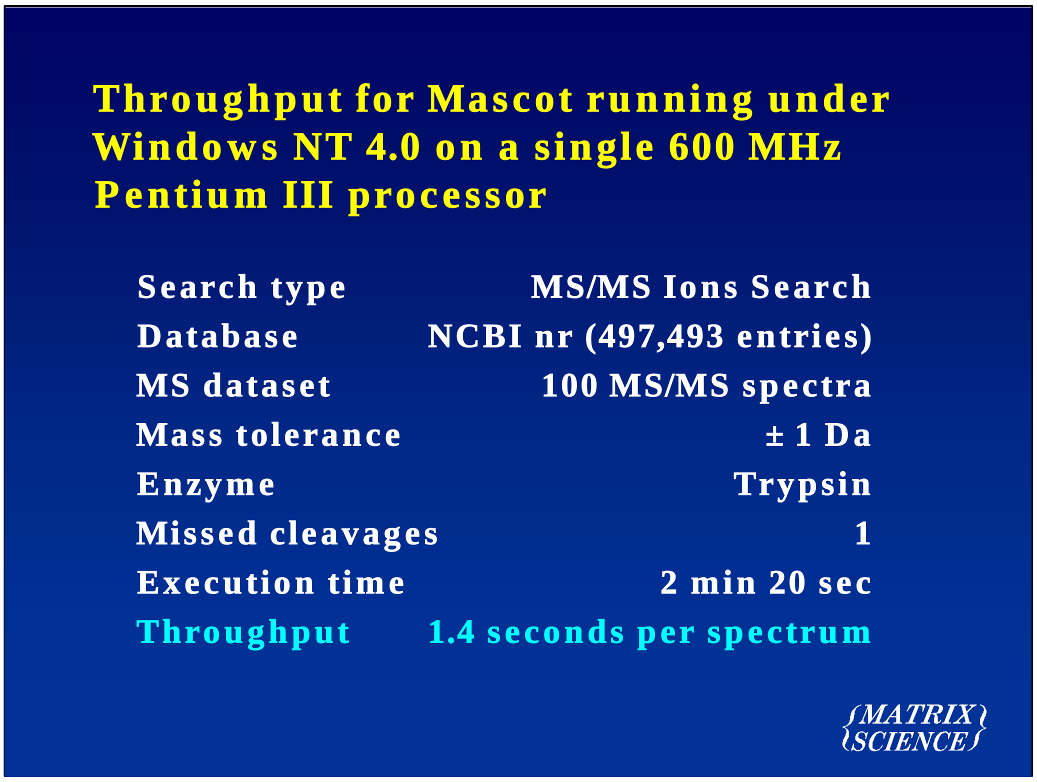
In Mascot, we calculate the probability that the observed match is a random event. The real match, which is not a random event, then has a very low probability.

The calculated probabilities are converted into scores by taking logs, so that a good match has a high score.

Assigning a significance threshold or confidence level to a match is then very simple. Assume we are running a fully automated system and prefer to repeat an experiment rather than get a false positive. We might choose a significance threshold of 1 in 1,000. That is, we are only interested in results which have less than a 1 in 1,000 chance of being random events.

If the database being searched has 500,000 protein entries, a 1 in 1,000 chance of finding a match is simply 1 over 1,000 times 500,000. Which converts into a Mascot score of 87.

So, we can have a simple rule in software which looks for matches with scores greater than 87.



Throughput for Mascot running under Windows NT 4.0 on a single 600 MHz Pentium III processor

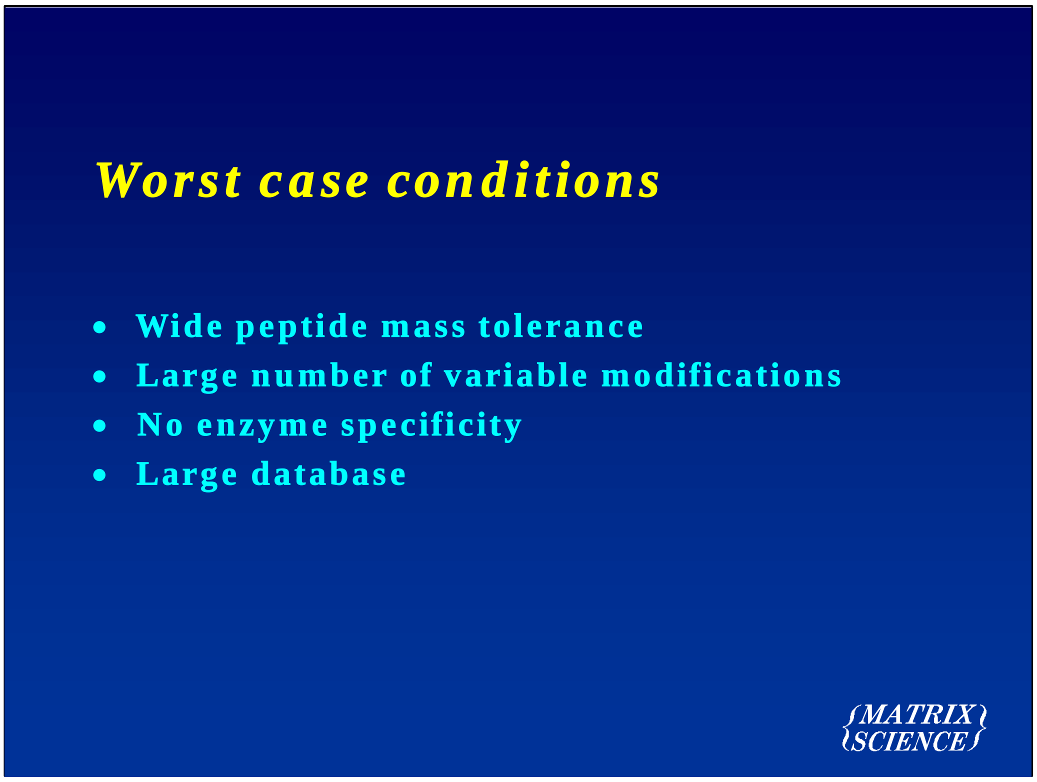
Search type	MS/MS Ions Search
Database	NCBI nr (497,493 entries)
MS dataset	100 MS/MS spectra
Mass tolerance	± 1 Da
Enzyme	Trypsin
Missed cleavages	1
Execution time	2 min 20 sec
Throughput	1.4 seconds per spectrum

{MATRIX}
{SCIENCE}

As mentioned earlier, to be useful in practice, searches need to be fast. We've worked hard to make Mascot as fast as possible, and here is a typical benchmark.

Note that this is a pretty average processor, not the latest and fastest. Even with a single 600 MHz processor, we can search MS/MS data against a database of half a million entries at a rate of one spectrum every 1.4 seconds using the parameters shown here.

Note that Mascot does not use pre-calculated indexes of mass values. All calculations are done on the fly while streaming through the FASTA sequence data. This is important for flexibility. It makes it possible to specify variable modifications. That is, modifications which may or may not be present or which may not be quantitative. Also, MS/MS data can be searched with no enzyme specificity, so as to find non-specific cleavage products or peptides which are not the result of enzyme cleavage.



Worst case conditions

- **Wide peptide mass tolerance**
- **Large number of variable modifications**
- **No enzyme specificity**
- **Large database**

{MATRIX}
{SCIENCE}

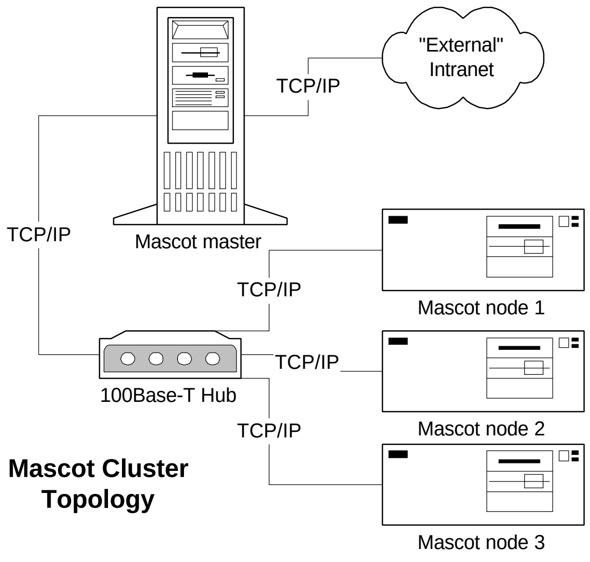
While this benchmark represents a fairly typical example, it is far from a worst case. Worst case search times are a result of pushing these four factors:

The search space is roughly proportional to the peptide mass tolerance.

Each additional variable or non-quantitative modification can cause a geometric increase in search time.

No enzyme specificity requires orders of magnitude more peptides to be tested than (say) trypsin.

Finally, and most obviously, the size of the database.



{MATRIX} {SCIENCE}

So, to do comprehensive searches of large databases in real-time, it becomes essential to farm out the problem to a large number of processors. This can, of course, be done on a large, multi-processor server.

But, there is increasing interest in using clusters of standard systems, such as Alphas or PC's.

Support for execution on networked clusters is part of the standard Mascot package. Many people choose to assemble their own cluster.

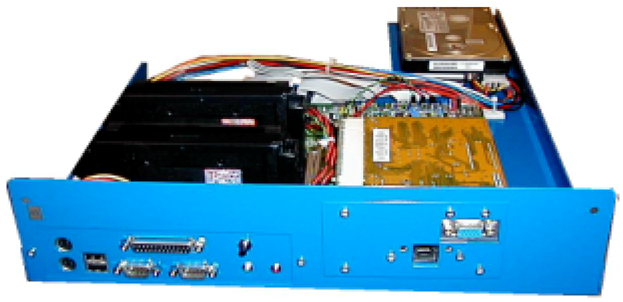
Such clusters can be very large. At GeneProt, in Switzerland, Mascot is being run on more than one thousand Compaq Alpha processors, so as to perform exhaustive searches in real time on data flowing from 50 high performance mass spectrometers.

Mascot Cluster

- **19” rack mount chassis containing up to 5 dual processor nodes**

On a slightly more modest scale, we can supply a complete turn-key system based on a 19” rackmount chassis containing up to 5 dual processor nodes. The nodes are hot-swappable and the chassis contains redundant power supplies feeding a common power distribution bus.

Mascot Cluster Node

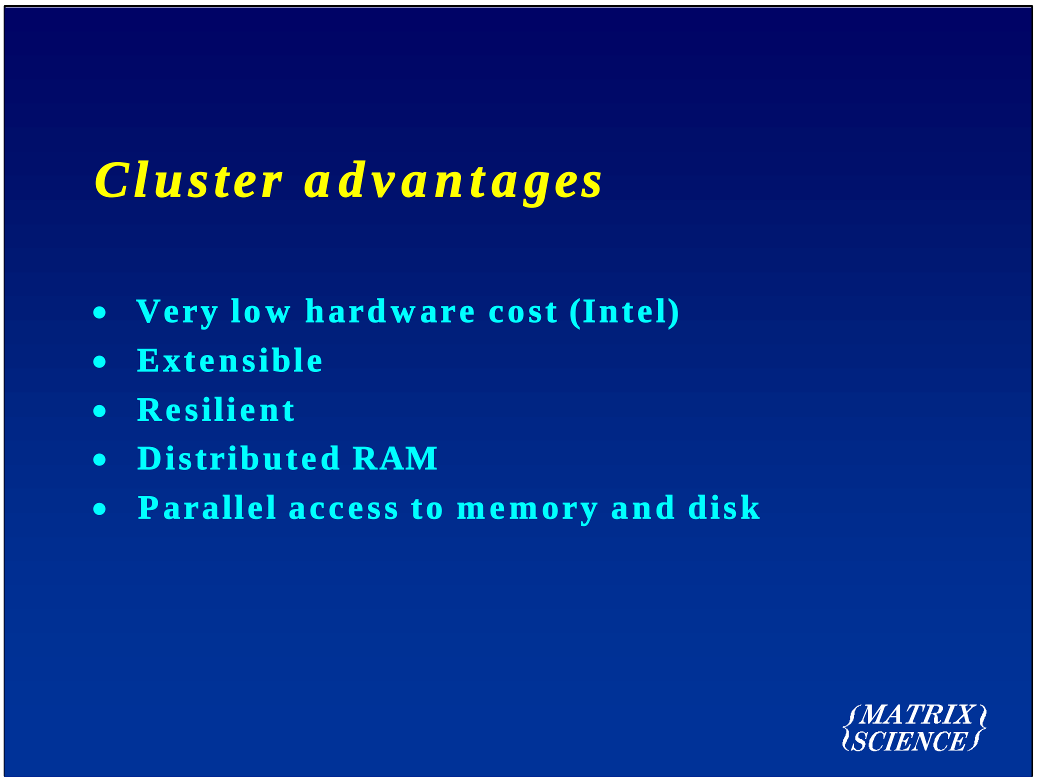


- **Tyan Thunder LE S2510 incorporating ServerWorks LE chipset and dual Intel ethernet ports**
- **Dual Pentium III 933 MHz processors with on-chip L2 cache**
- **1 Gb PC133 RAM**
- **75 Gb IBM hard drive**
- **Windows 2000 Professional or Linux**

{MATRIX} {SCIENCE}

Each node contains a high performance motherboard with fast Pentium processors and a gigabyte of RAM.

Since Mascot is almost perfectly parallelisable, running the earlier benchmark on a 10 processor cluster would take the throughput from 1.4 seconds per spectrum to 7 spectra per second.



Cluster advantages

- Very low hardware cost (Intel)
- Extensible
- Resilient
- Distributed RAM
- Parallel access to memory and disk

{MATRIX}
{SCIENCE}

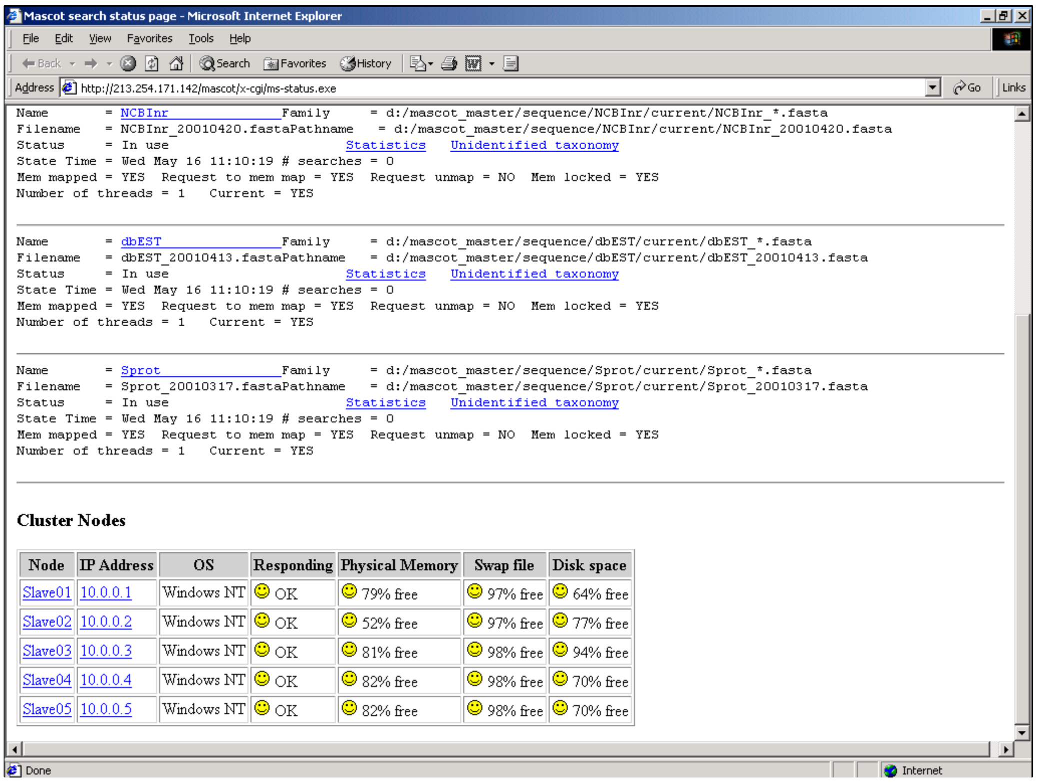
The primary attraction of a cluster has to be the low hardware cost. Using PC's. the cost is around an order of magnitude lower than a multiprocessor server of equivalent throughput. (e.g. $35k versus $350k for 10 CPU's)

Another attraction is versatility. It is very difficult to predict exactly how much processing power will be needed for a particular project. With a cluster, you can start conservatively and just add boxes as needed. If you buy a multi-processor server with a maximum capacity of 8 CPU's, and then find you need 12, the consequences can be costly and disruptive.

A cluster is resilient because a node can fail without bringing down the whole system. If Mascot detects that a node has stopped responding, it will reconfigure itself, either without the node or with a replacement node, and continue operation. Meanwhile, the faulty node can be repaired.

One of the limitations of standard PC's is that they usually only have 4 slots for RAM, and DIMM bigger than 256 Mb are very expensive. So, 1 Gb RAM per PC is often a practical upper limit. With a cluster, we can have 1 Gb per system.

Compared with supercomputers, standard PC's have relatively slow bandwidth to disk and memory. By spreading the load across many parallel systems, this bottleneck is alleviated.



```
Name      = NCBInr                Family     = d:/mascot_master/sequence/NCBInr/current/NCBInr_*.fasta
Filename  = NCBInr_20010420.fastaPathname   = d:/mascot_master/sequence/NCBInr/current/NCBInr_20010420.fasta
Status    = In use                        Statistics   Unidentified taxonomy
State Time = Wed May 16 11:10:19 # searches = 0
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = YES
Number of threads = 1   Current = YES

Name      = dbEST                 Family     = d:/mascot_master/sequence/dbEST/current/dbEST_*.fasta
Filename  = dbEST_20010413.fastaPathname   = d:/mascot_master/sequence/dbEST/current/dbEST_20010413.fasta
Status    = In use                        Statistics   Unidentified taxonomy
State Time = Wed May 16 11:10:19 # searches = 0
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = YES
Number of threads = 1   Current = YES

Name      = Sprot                 Family     = d:/mascot_master/sequence/Sprot/current/Sprot_*.fasta
Filename  = Sprot_20010317.fastaPathname   = d:/mascot_master/sequence/Sprot/current/Sprot_20010317.fasta
Status    = In use                        Statistics   Unidentified taxonomy
State Time = Wed May 16 11:10:19 # searches = 0
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = YES
Number of threads = 1   Current = YES
```

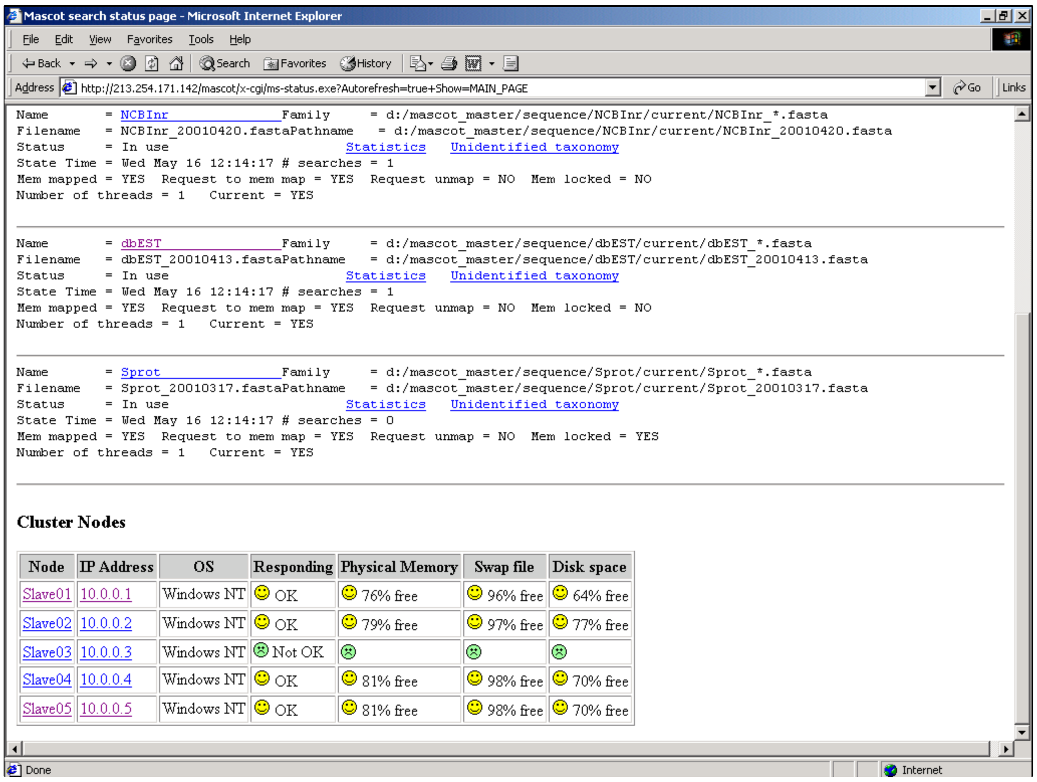
Cluster Nodes

Node	IP Address	OS	Responding	Physical Memory	Swap file	Disk space
Slave01	10.0.0.1	Windows NT	OK	79% free	97% free	64% free
Slave02	10.0.0.2	Windows NT	OK	52% free	97% free	77% free
Slave03	10.0.0.3	Windows NT	OK	81% free	98% free	94% free
Slave04	10.0.0.4	Windows NT	OK	82% free	98% free	70% free
Slave05	10.0.0.5	Windows NT	OK	82% free	98% free	70% free

A Mascot cluster is like a Beowulf cluster. One node acts as the master and distributes the executables and databases to all of the slave nodes. Similarly, each search is divided up by the master and distributed to the slaves. At the end of the search, the master collates the results back into a single file.

There are tools to allow a system administrator to monitor the status of each of the nodes from a web browser. This table, for example, shows at a glance if one of the nodes is running low on resources or failing to respond.

Mascot search status page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://213.254.171.142/mascot/x-cgi/ms-status.exe
Go Links
Done
Internet



Mascot search status page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://213.254.171.142/mascot/x-cgi/ms-status.exe?Autorefresh=true+Show=MAIN_PAGE
Go Links

```
Name       = NCBInr                 Family     = d:/mascot_master/sequence/NCBInr/current/NCBInr_*.fasta
Filename   = NCBInr_20010420.fastaPathname   = d:/mascot_master/sequence/NCBInr/current/NCBInr_20010420.fasta
Status     = In use                          Statistics   Unidentified taxonomy
State Time = Wed May 16 12:14:17 # searches = 1
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = NO
Number of threads = 1   Current = YES
```

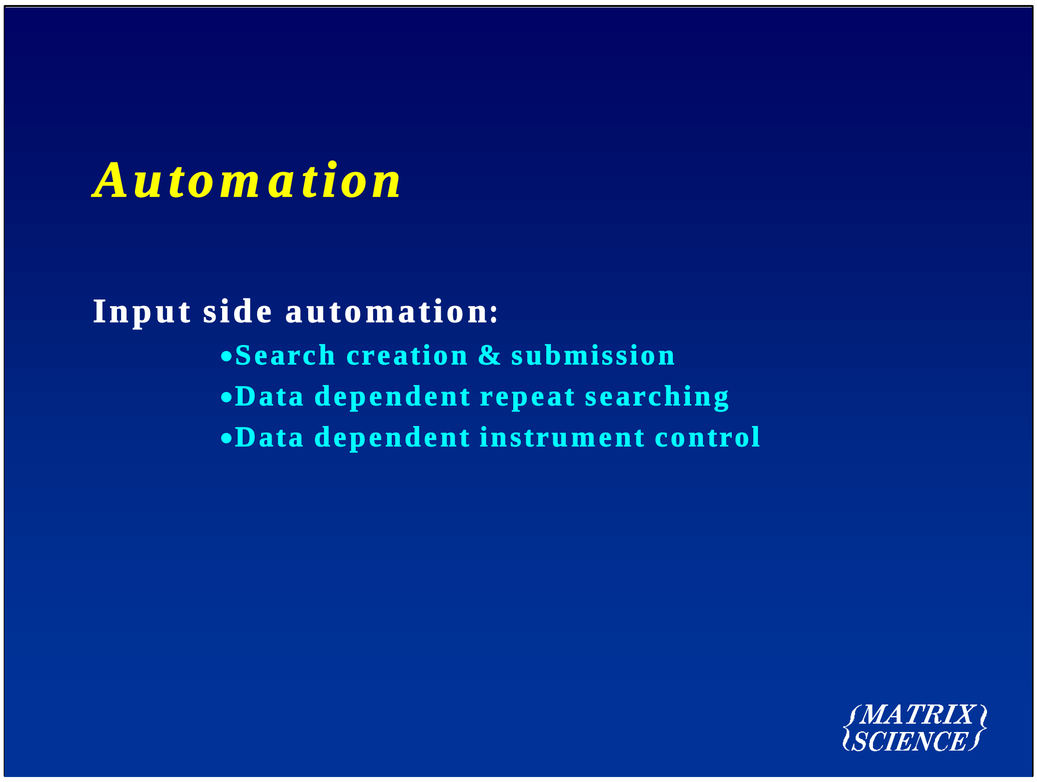
```
Name       = dbEST                  Family     = d:/mascot_master/sequence/dbEST/current/dbEST_*.fasta
Filename   = dbEST_20010413.fastaPathname   = d:/mascot_master/sequence/dbEST/current/dbEST_20010413.fasta
Status     = In use                          Statistics   Unidentified taxonomy
State Time = Wed May 16 12:14:17 # searches = 1
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = NO
Number of threads = 1   Current = YES
```

```
Name       = Sprot                  Family     = d:/mascot_master/sequence/Sprot/current/Sprot_*.fasta
Filename   = Sprot_20010317.fastaPathname   = d:/mascot_master/sequence/Sprot/current/Sprot_20010317.fasta
Status     = In use                          Statistics   Unidentified taxonomy
State Time = Wed May 16 12:14:17 # searches = 0
Mem mapped = YES  Request to mem map = YES  Request unmap = NO  Mem locked = YES
Number of threads = 1   Current = YES
```

Cluster Nodes

Node	IP Address	OS	Responding	Physical Memory	Swap file	Disk space
Slave01	10.0.0.1	Windows NT	☺ OK	☺ 76% free	☺ 96% free	☺ 64% free
Slave02	10.0.0.2	Windows NT	☺ OK	☺ 79% free	☺ 97% free	☺ 77% free
Slave03	10.0.0.3	Windows NT	☹ Not OK	☹	☹	☹
Slave04	10.0.0.4	Windows NT	☺ OK	☺ 81% free	☺ 98% free	☺ 70% free
Slave05	10.0.0.5	Windows NT	☺ OK	☺ 81% free	☺ 98% free	☺ 70% free

This is an example.



Automation

Input side automation:

- **Search creation & submission**
- **Data dependent repeat searching**
- **Data dependent instrument control**

MATRIX SCIENCE

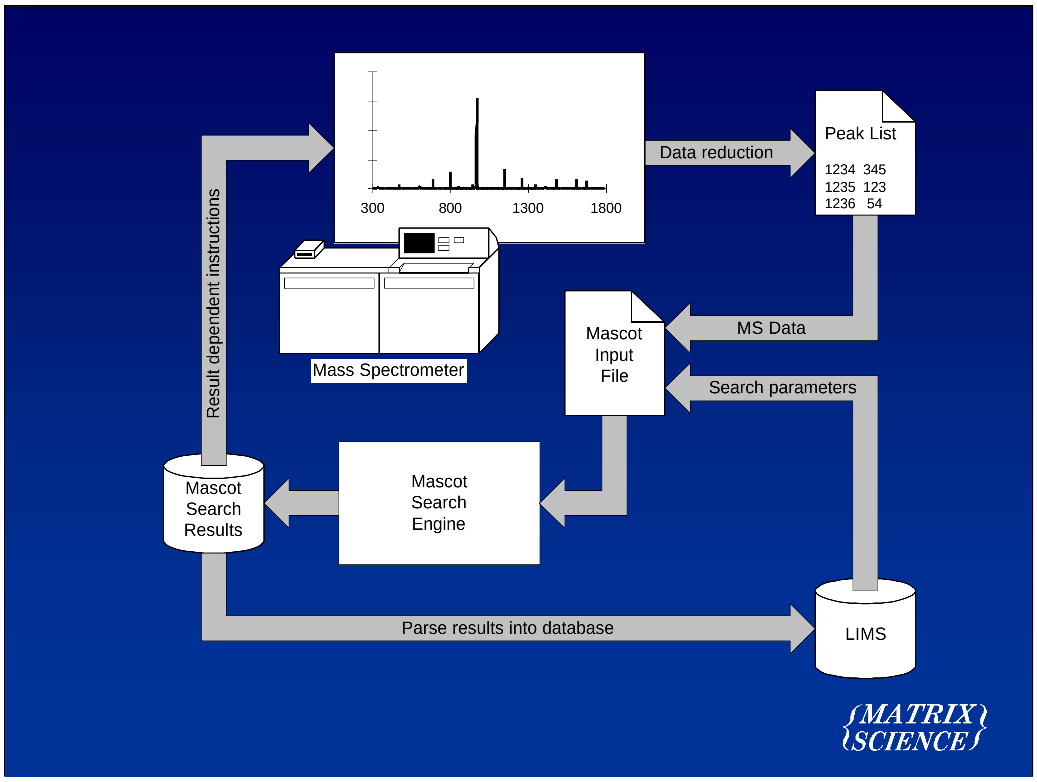
The final, crucial requirement for high throughput work is automation.

Automation can be divided into two. Input side or front-end automation covers getting the searches into Mascot. Output side automation covers doing something with the results.

On the input side, we need to have mechanisms for creating searches by assembling mass spectrometry data and search parameters, then submitting the searches to Mascot.

Second, we need to be able to chain searches together so as to implement complex search strategies.

Third, in an ideal world we will have closed loop feedback to the instrument, so that we can use the search results to make best use of available instrument time. For example, curtailing acquisition once a goal has been achieved, or repeating an acquisition if the result is unclear.



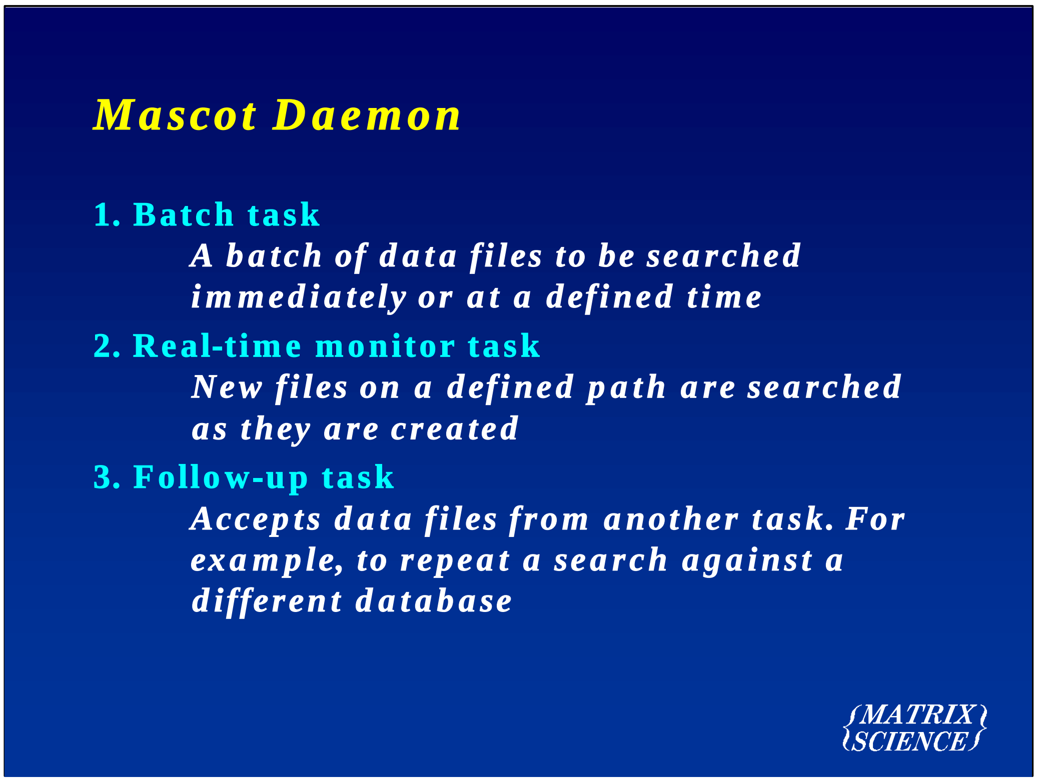
This diagram illustrates the data flow for high throughput protein identification.

Starting at the top, the raw data from the mass spectrometer must be reduced to high quality peak lists. That is, pairs of mass and intensity values rather than profile data. At present, we depend on the instrument control data system to perform this task. This data reduction is often the weak link in the chain …real-time peak detection, de-isotoping, and de-charging is not handled well by all data systems.

The peak lists must then be combined in some fashion with a number of search parameters. For example, which database is to be searched?, what enzyme was used for digestion?, etc.

We provide an automation client, called Mascot Daemon. Daemon can implement the front end automation without any custom programming by using predefined sets of parameters.

A more logical source for the search parameters is the LIMS which manages sample tracking. There is a mechanism to achieve this with Mascot Daemon, but it requires some programming on the LIMS side. Likewise, parsing of result files into the LIMS and data dependent instrument control require some custom programming at present.



Mascot Daemon

1. **Batch task**

   A batch of data files to be searched immediately or at a defined time

2. **Real-time monitor task**

   New files on a defined path are searched as they are created

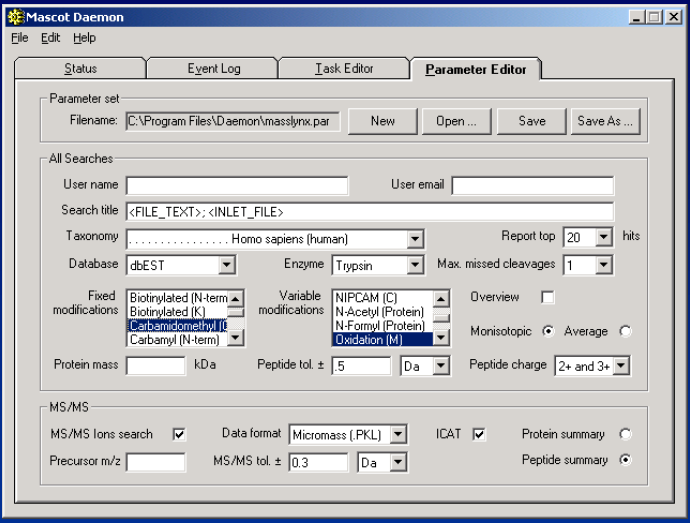
3. **Follow-up task**

   Accepts data files from another task. For example, to repeat a search against a different database

{MATRIX}
{SCIENCE}

Mascot Daemon runs on any Win32 platform and supports three kinds of tasks.

The follow-up task is very powerful because it allows searches to be chained together to implement complex decision paths. For example, as batch of data files might be screened against a contaminants database containing entries for keratins, BSA, trypsin, etc. Those data files which fail to find a match can then be automatically searched against a non-redundant protein database. Spectra which are still unmatched can then be searched against a large EST database, etc., etc.



Mascot Daemon

File Edit Help

Status | Event Log | Task Editor | **Parameter Editor**

Parameter set

Filename: C:\Program Files\Daemon\masslynx.par

New | Open ... | Save | Save As ...

All Searches

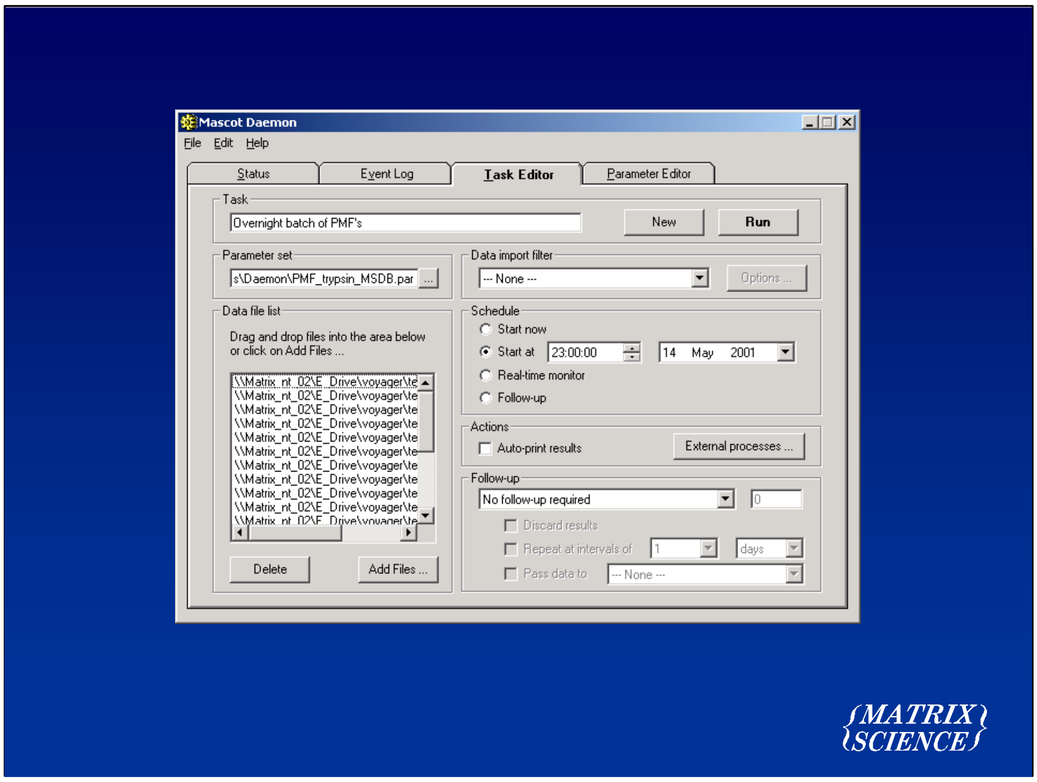
User name		User email	
Search title	<FILE_TEXT>; <INLET_FILE>		
Taxonomy	. . . . Homo sapiens (human)	Report top	20 hits
Database	dbEST	Enzyme	Trypsin
Max. missed cleavages	1		
Fixed modifications	Biotinylated (N-term; Biotinylated (K); Carbamidomethyl (C; Carbamyl (N-term)	Variable modifications	NIPCAM (C); N-Acetyl (Protein); N-Formyl (Protein); Oxidation (M)
Overview	☐	Monisotopic [✓]	Average
Protein mass	kDa	Peptide tol. ±	.5 Da
Peptide charge	2+ and 3+		

MS/MS

MS/MS Ions search	✓	Data format	Micromass (.PKL)
ICAT	✓	Protein summary	
Precursor m/z		MS/MS tol. ±	0.3 Da
Peptide summary	[✓]		

{MATRIX SCIENCE}

The parameter editor allows sets of search parameters to be defined and saved to disk, so that they can be used over and over again. The search parameters define *how* the data will be searched.



Mascot Daemon

File Edit Help

Status | Event Log | **Task Editor** | Parameter Editor

Task: Overnight batch of PMF's — New — **Run**

Parameter set: s\Daemon\PMF_trypsin_MSDB.par

Data import filter: --- None --- — Options ...

Data file list

Drag and drop files into the area below or click on Add Files ...

```
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
\\Matrix_nt_02\E_Drive\voyager\te
```

Delete — Add Files ...

Schedule

- Start now
- [✓ Start at] 23:00:00 — 14 May 2001
- Real-time monitor
- Follow-up

Actions

- Auto-print results — External processes ...

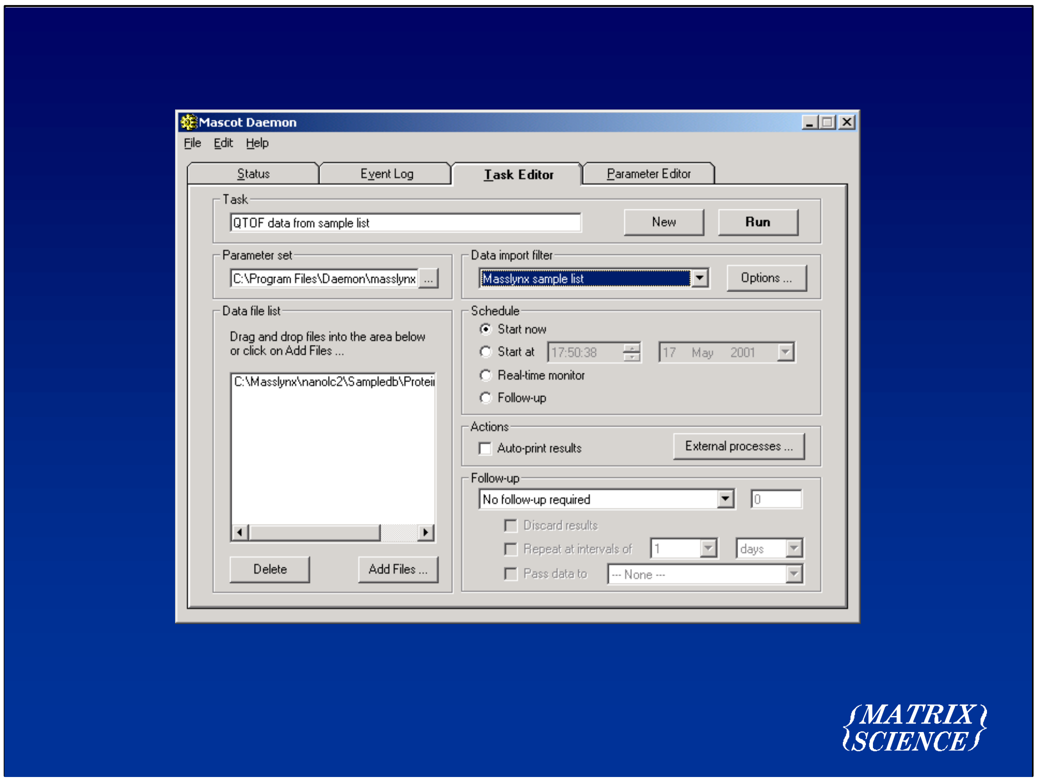
Follow-up: No follow-up required — 0

- Discard results
- Repeat at intervals of 1 days
- Pass data to --- None ---

{MATRIX SCIENCE}

The Task Editor tab is used to define each task. A task defines *what* data will be searched and and *when* the search will take place.

Here we have a simple batch task. A set of data files has been created, a parameter set has been chosen, and the task will run all the searches as a batch at a predefined time.



Task: QTOF data from sample list

Parameter set: C:\Program Files\Daemon\masslynx

Data import filter: Masslynx sample list

Data file list: Drag and drop files into the area below or click on Add Files ...

C:\Masslynx\nanolc2\Sampledb\Protei

Schedule: [✓ Start now] Start at 17:50:38 17 May 2001 Real-time monitor Follow-up

Actions: Auto-print results

Follow-up: No follow-up required 0

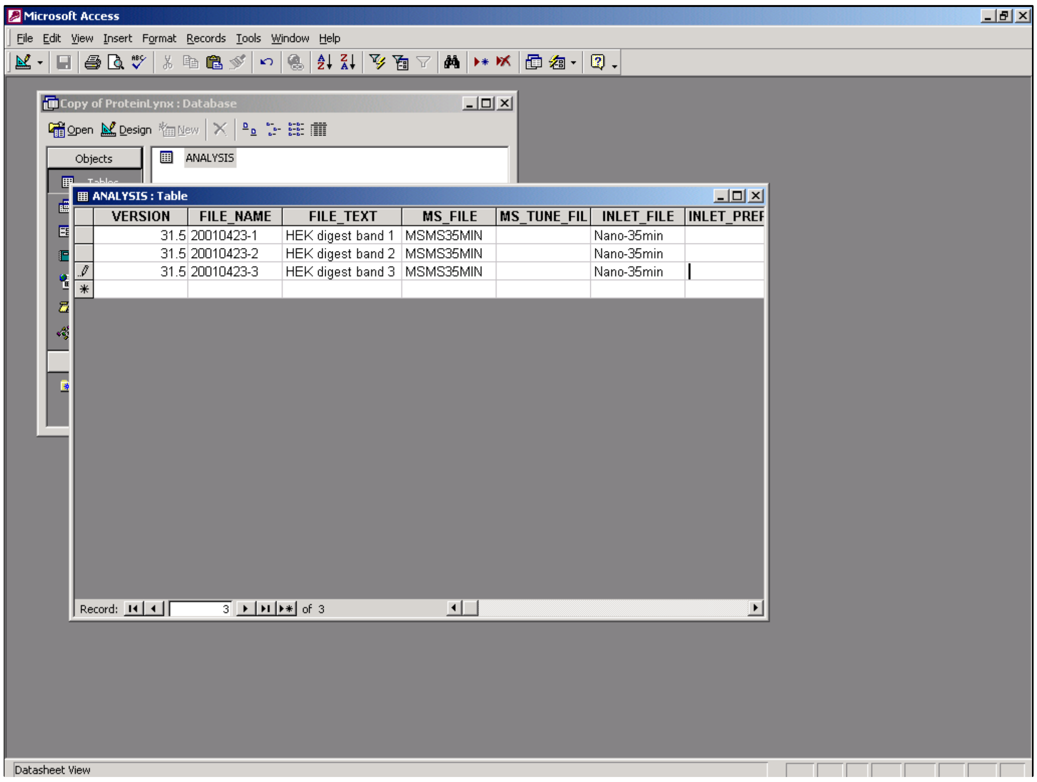
Discard results

Repeat at intervals of 1 days

Pass data to --- None ---

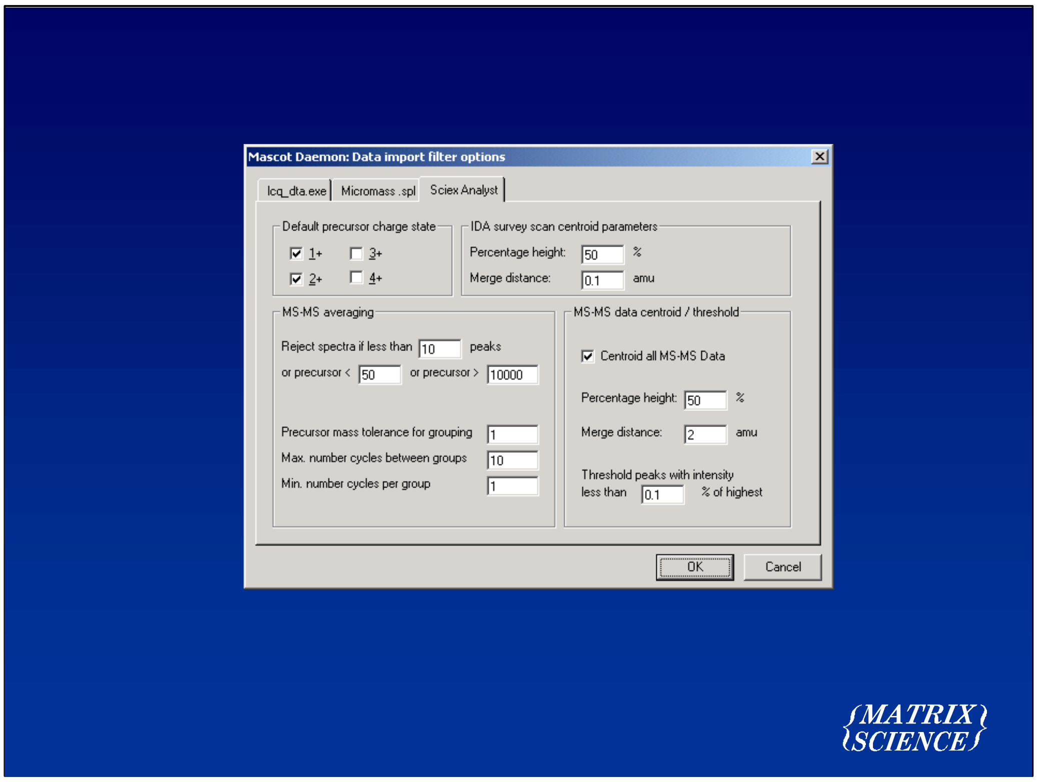
When you search Mascot using the web browser forms, you're limited to uploading one file of peak lists at a time. Daemon not only allows you to create batches of searches, it also provides some data import filters.

For example, we can open up a MassLynx sample list and fish out the data file names.



VERSION	FILE_NAME	FILE_TEXT	MS_FILE	MS_TUNE_FIL	INLET_FILE	INLET_PREF
31.5	20010423-1	HEK digest band 1	MSMS35MIN		Nano-35min	
31.5	20010423-2	HEK digest band 2	MSMS35MIN		Nano-35min	
31.5	20010423-3	HEK digest band 3	MSMS35MIN		Nano-35min	

We can use tags in title field of the Mascot search which get substituted at run time by fields from the MassLynx sample list. For example, “HEK digest band 1”, or any text from any of the columns, could appear at the top of the Mascot search report.


Mascot Daemon: Data import filter options

lcq_dta.exe | Micromass .spl | Sciex Analyst

Default precursor charge state: [✓ 1+] 3+ [✓ 2+] 4+

IDA survey scan centroid parameters
Percentage height: 50 %
Merge distance: 0.1 amu

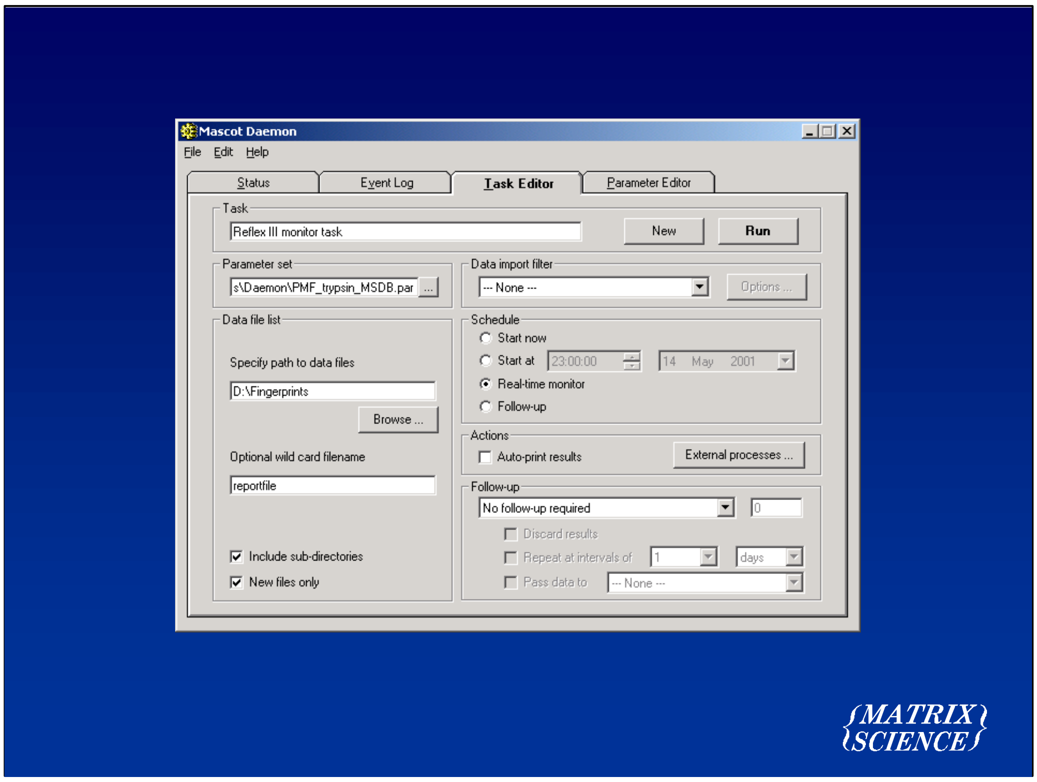
MS-MS averaging
Reject spectra if less than 10 peaks
or precursor < 50 or precursor > 10000
Precursor mass tolerance for grouping 1
Max. number cycles between groups 10
Min. number cycles per group 1

MS-MS data centroid / threshold
[✓ Centroid all MS-MS Data]
Percentage height: 50 %
Merge distance: 2 amu
Threshold peaks with intensity less than 0.1 % of highest

We also have an import filter which uses the lcq_dta.exe utility from ThermoFinnigan to convert Xcalibur .RAW files to .DTA files.

And, recently, we've added an import filter for Sciex Analyst .WIFF files.

Both the Xcalibur and the Analyst import filters depend on libraries supplied with the data systems. Daemon must be running on a system which has the relevant software installed to take advantage of these features.


Mascot Daemon

File Edit Help

Status | Event Log | **Task Editor** | Parameter Editor

Task: Reflex III monitor task — New — Run

Parameter set: s\Daemon\PMF_trypsin_MSDB.par

Data import filter: --- None --- — Options ...

Data file list

Specify path to data files: D:\Fingerprints — Browse ...

Optional wild card filename: reportfile

[✓] Include sub-directories

[✓] New files only

Schedule: ○ Start now ○ Start at 23:00:00 14 May 2001 [● Real-time monitor] ○ Follow-up

Actions: [ ] Auto-print results — External processes ...

Follow-up: No follow-up required — 0

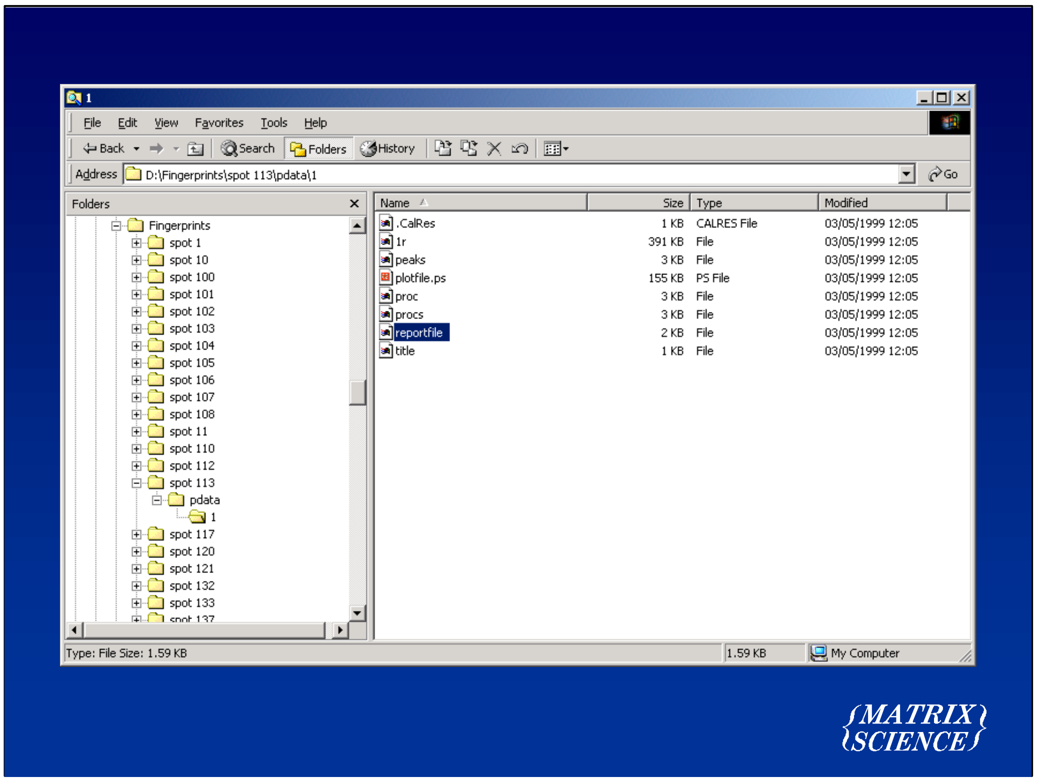
[ ] Discard results

[ ] Repeat at intervals of 1 days

[ ] Pass data to --- None ---

Batch searches are a great timesaver. But, the most commonly used task for true automation is the real-time monitor task, sometimes described as a sniffer.

A real-time monitor task watches a particular path, that is a directory, and looks for new files which match a wild card pattern. In this example, we are looking in the directory d:\fingerprints, and all its subdirectories, for any file called reportfile.

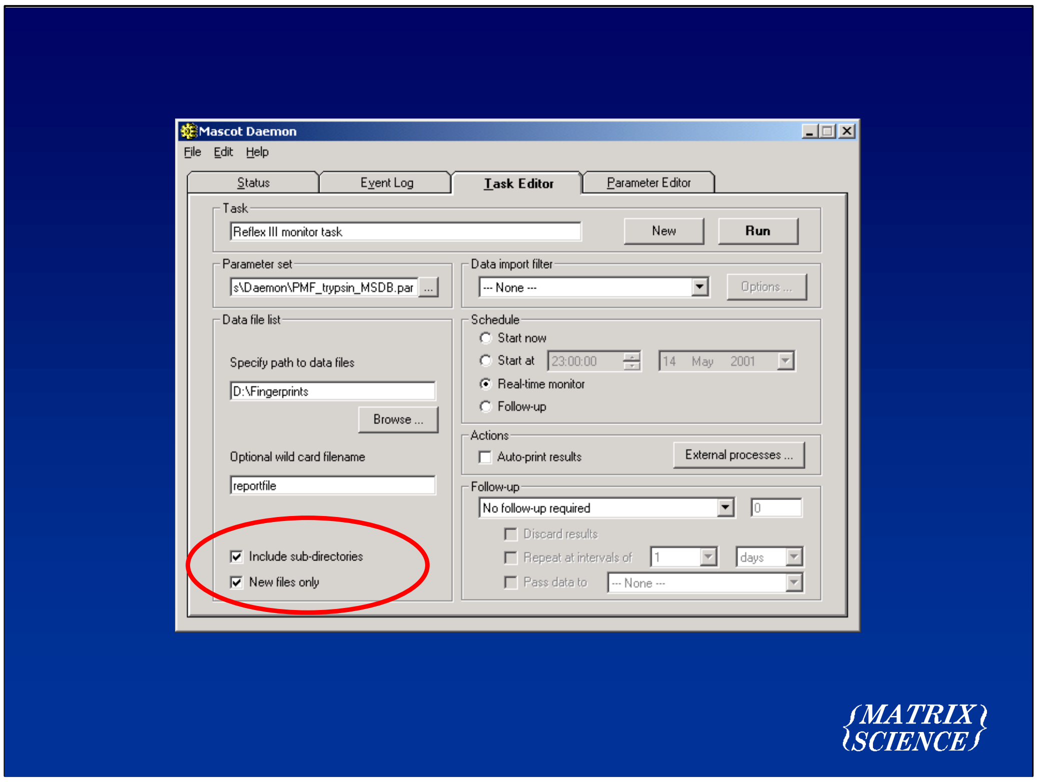

1
File Edit View Favorites Tools Help
Back Search Folders History
Address D:\Fingerprints\spot 113\pdata\1
Go
Folders

Fingerprints
- spot 1
- spot 10
- spot 100
- spot 101
- spot 102
- spot 103
- spot 104
- spot 105
- spot 106
- spot 107
- spot 108
- spot 11
- spot 110
- spot 112
- spot 113
  - pdata
    - 1
- spot 117
- spot 120
- spot 121
- spot 132
- spot 133
- spot 137

Name	Size	Type	Modified
.CalRes	1 KB	CALRES File	03/05/1999 12:05
1r	391 KB	File	03/05/1999 12:05
peaks	3 KB	File	03/05/1999 12:05
plotfile.ps	155 KB	PS File	03/05/1999 12:05
proc	3 KB	File	03/05/1999 12:05
procs	3 KB	File	03/05/1999 12:05
reportfile	2 KB	File	03/05/1999 12:05
title	1 KB	File	03/05/1999 12:05

Type: File Size: 1.59 KB 1.59 KB My Computer

This could be used to watch for Bruker XMASS data files, where each sample generates a new subdirectory structure. Other data systems put the peak lists into a common directory with different filenames, so we would use a wildcard filename like *.PKL



Mascot Daemon

File Edit Help

Status | Event Log | Task Editor | Parameter Editor

Task: Reflex III monitor task — New — Run

Parameter set: s\Daemon\PMF_trypsin_MSDB.par

Data import filter: --- None --- — Options ...

Data file list

Specify path to data files: D:\Fingerprints — Browse ...

Optional wild card filename: reportfile

[✓] Include sub-directories

[✓] New files only

Schedule

○ Start now

○ Start at 23:00:00 14 May 2001

● Real-time monitor

○ Follow-up

Actions

☐ Auto-print results — External processes ...

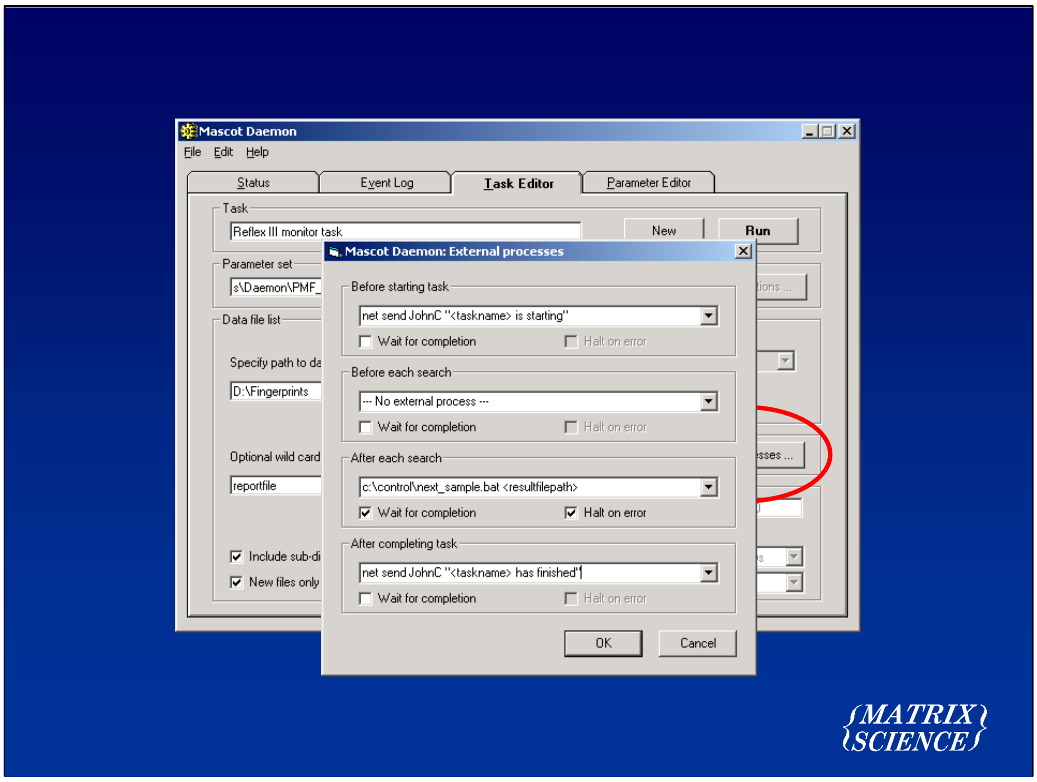
Follow-up: No follow-up required — 0

☐ Discard results

☐ Repeat at intervals of 1 days

☐ Pass data to --- None ---

The new files only checkbox determines whether Daemon searches files which existed when the task is started, or just looks for new files. If you put all of your data files into one humungous directory, starting a task with this checkbox cleared can be a dangerous thing to do.



Task: Reflex III monitor task

Parameter set: s\Daemon\PMF_

Data file list

Specify path to da: D:\Fingerprints

Optional wild card: reportfile

[✓] Include sub-di

[✓] New files only

Mascot Daemon: External processes

Before starting task: net send JohnC "<taskname> is starting"
[ ] Wait for completion [ ] Halt on error

Before each search: --- No external process ---
[ ] Wait for completion [ ] Halt on error

After each search: c:\control\next_sample.bat <resultfilepath>
[✓] Wait for completion [✓] Halt on error

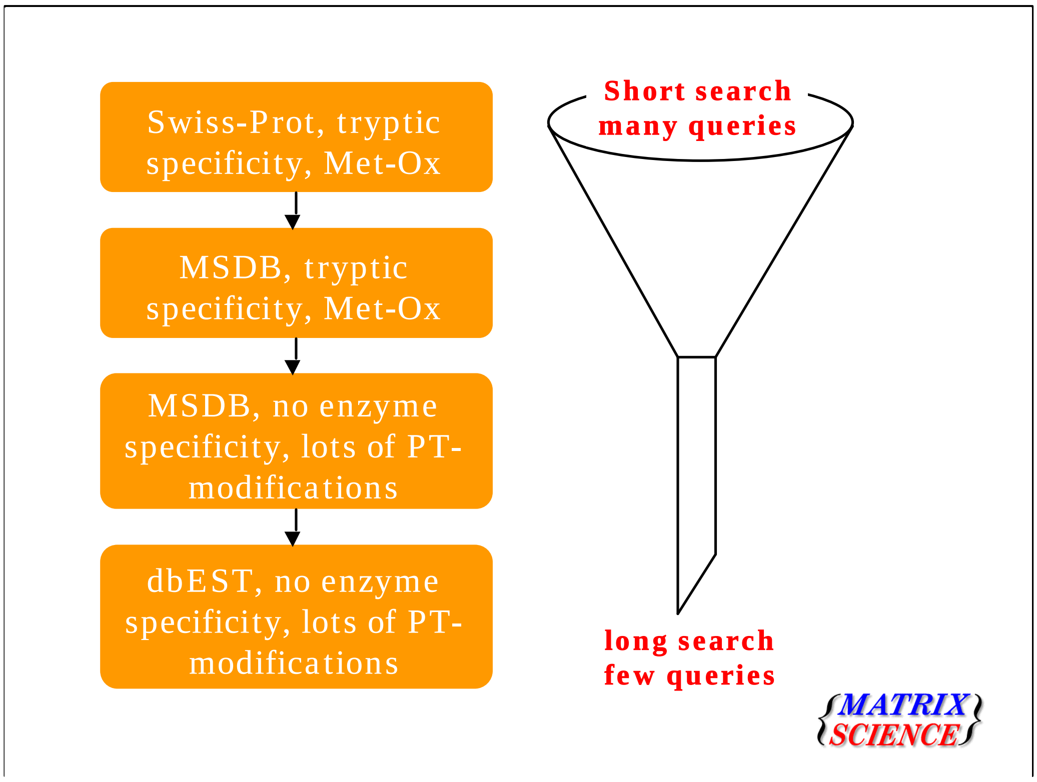
After completing task: net send JohnC "<taskname> has finished"
[ ] Wait for completion [ ] Halt on error

External processes allows us to hook in all sorts of functionality, such as data dependent instrument control.

You can define external processes at any or all of these time points: the start of a task, the start of a search, search completion, or task completion.

The external process could be a simple utility, such as net send to send a message to the task owner. Or, it could be a batch file which will result in a series of actions, such as opening up a result file, fish out one or more scores, and tell the instrument whether to move on to the next sample.

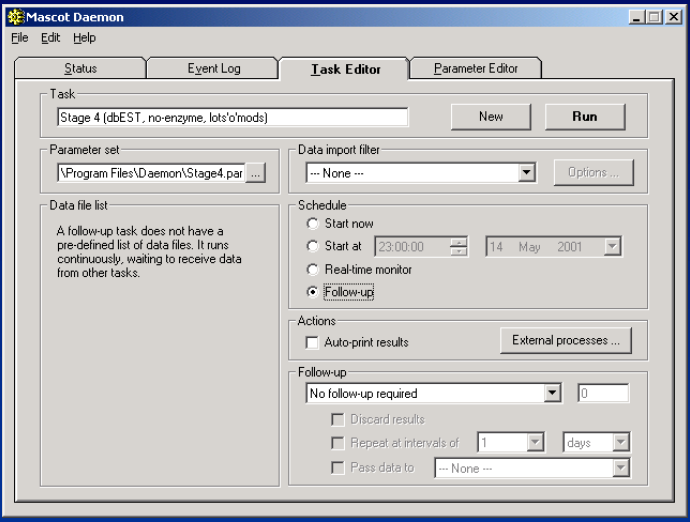
In order to achieve this, tags can be specified as arguments, such as <resultfilepath>.


Swiss-Prot, tryptic specificity, Met-Ox
MSDB, tryptic specificity, Met-Ox
MSDB, no enzyme specificity, lots of PT-modifications
dbEST, no enzyme specificity, lots of PT-modifications
Short search many queries
long search few queries
MATRIX SCIENCE

Maximum throughput cannot be achieved by searching every datafile against the biggest database with the widest possible range of modifications and no enzyme specificity. This would take forever.

For efficiency, we want to identify as many spectra as possible using fast, simple searches against a small, high-quality database such as Swiss-prot.

Spectra which cannot be identified easily, can then be searched against progressively larger databases, using progressively more exhaustive search parameters.



Task: Stage 4 (dbEST, no-enzyme, lots'o'mods)

Parameter set: \Program Files\Daemon\Stage4.par

Data import filter: --- None ---

Data file list

A follow-up task does not have a pre-defined list of data files. It runs continuously, waiting to receive data from other tasks.

Schedule
- ○ Start now
- ○ Start at 23:00:00 14 May 2001
- ○ Real-time monitor
- ● Follow-up

Actions
- ☐ Auto-print results

Follow-up: No follow-up required 0
- ☐ Discard results
- ☐ Repeat at intervals of 1 days
- ☐ Pass data to --- None ---

{MATRIX}
{SCIENCE}

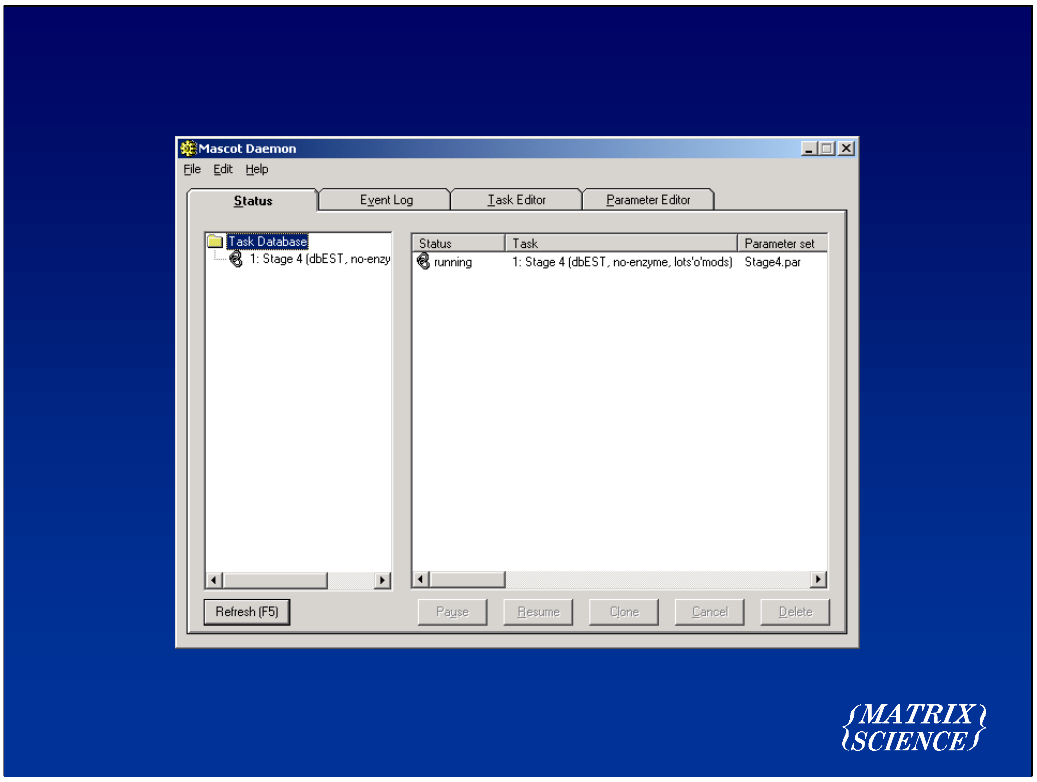
The way we achieve this in Daemon is with a follow-up task. This is a task which doesn't have a predefined list of data files. It runs continuously, waiting to receive data files passed from other tasks.

The clever trick is that a follow-up task can pass data to another follow-up task. This allows us to create chains of tasks to implement complex search strategies.

The thing which is conceptually difficult about follow-up tasks is that you have to define the chain starting from the end.

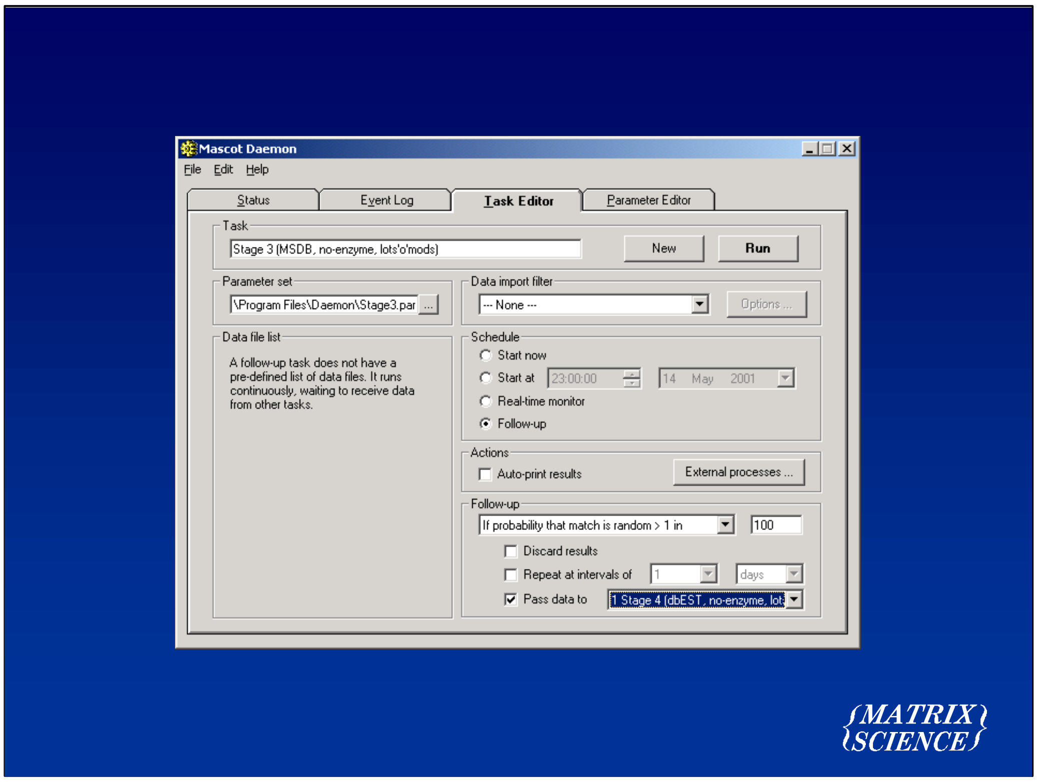
This is obvious if you think about it. The task we are going to pass the data to has to exist so that it can be identified in the preceding task.

So, to implement the task chain we saw in the earlier slide, we start by defining the stage 4 task, the wide open dbEST search.



Status	Task	Parameter set
running	1: Stage 4 (dbEST, no-enzyme, lots'o'mods)	Stage4.par

We press run, and it appears on the status tree, waiting for something to happen.



Task: Stage 3 (MSDB, no-enzyme, lots'o'mods) — New — Run

Parameter set: \Program Files\Daemon\Stage3.par ...

Data import filter: --- None --- — Options ...

Data file list: A follow-up task does not have a pre-defined list of data files. It runs continuously, waiting to receive data from other tasks.

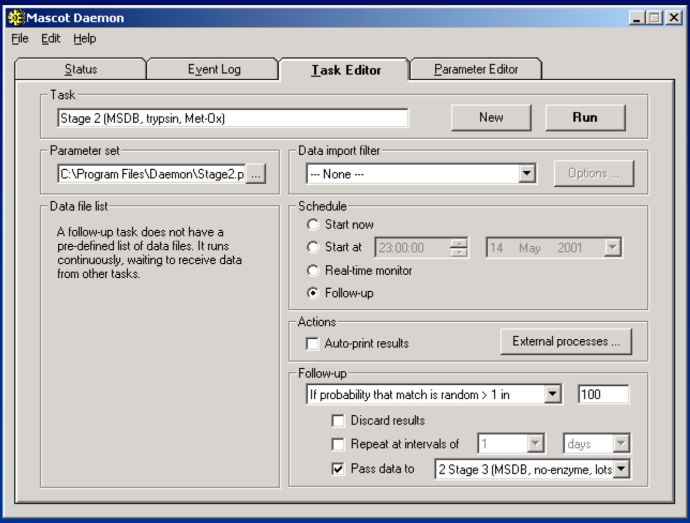
Schedule:
- ( ) Start now
- ( ) Start at 23:00:00 14 May 2001
- ( ) Real-time monitor
- [✓ Follow-up]

Actions:
- [ ] Auto-print results — External processes ...

Follow-up: If probability that match is random > 1 in 100
- [ ] Discard results
- [ ] Repeat at intervals of 1 days
- [✓] Pass data to 1 Stage 4 (dbEST, no-enzyme, lots

We then define the stage 3 task, a wide open nr search. Notice that we have set-up a follow-up chain. If the probability that the match is random is greater than 1 in 100, that is, the match is not a good one, then pass the data to the stage 4 task.

For a peptide mass fingerprint, this score threshold applies to the protein score for each complete search. For an MS/MS search, the score threshold applies independently to the peptide score for each MS/MS spectrum. Each task sieves out the spectra which can be matched and passes the ones which can't be matched on to the next task.



Mascot Daemon

File Edit Help

Status | Event Log | Task Editor | Parameter Editor

Task: Stage 2 (MSDB, trypsin, Met-Ox) New Run

Parameter set: C:\Program Files\Daemon\Stage2.p ...

Data import filter: --- None --- Options ...

Data file list: A follow-up task does not have a pre-defined list of data files. It runs continuously, waiting to receive data from other tasks.

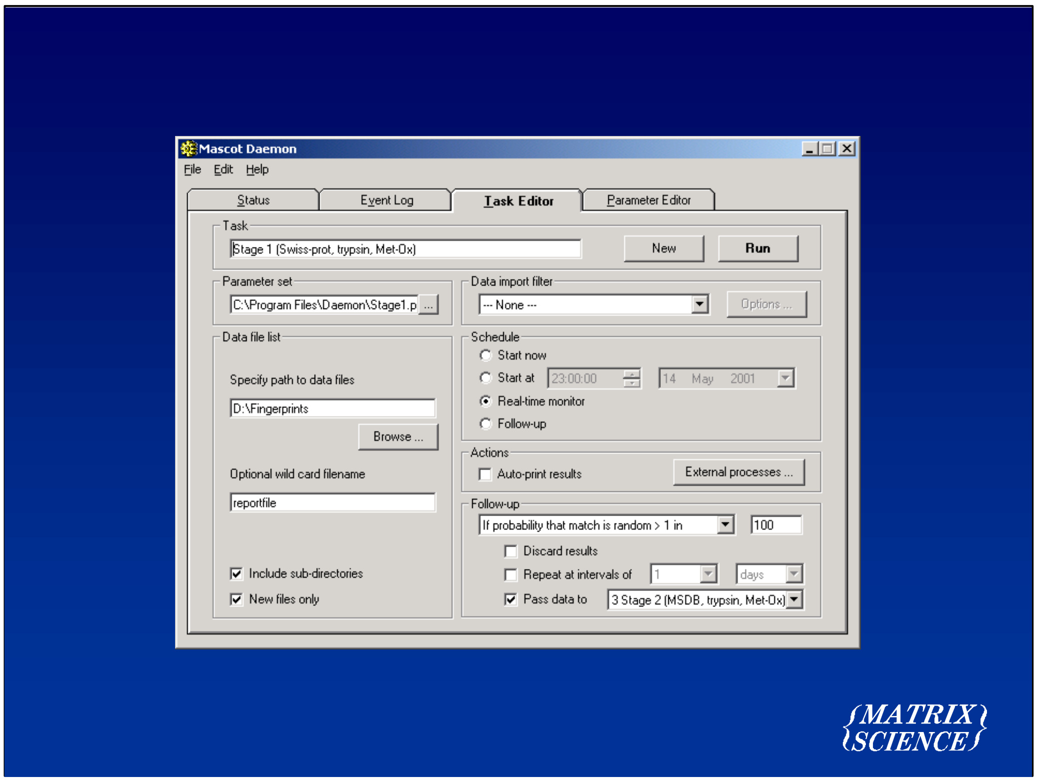
Schedule:
- ( ) Start now
- ( ) Start at 23:00:00 14 May 2001
- ( ) Real-time monitor
- (•) Follow-up

Actions:
- [ ] Auto-print results External processes ...

Follow-up:
- If probability that match is random > 1 in 100
- [ ] Discard results
- [ ] Repeat at intervals of 1 days
- [✓] Pass data to 2 Stage 3 (MSDB, no-enzyme, lots

{MATRIX SCIENCE}

And then the stage 2 task, a fast nr search, which passes its failures to stage 3.



Mascot Daemon

File Edit Help

Status | Event Log | Task Editor | Parameter Editor

Task: Stage 1 (Swiss-prot, trypsin, Met-Ox) — New — Run

Parameter set: C:\Program Files\Daemon\Stage1.p ...

Data import filter: --- None --- — Options ...

Data file list

Specify path to data files: D:\Fingerprints — Browse ...

Optional wild card filename: reportfile

[✓] Include sub-directories

[✓] New files only

Schedule

- ( ) Start now
- ( ) Start at 23:00:00 14 May 2001
- (●) Real-time monitor
- ( ) Follow-up

Actions

[ ] Auto-print results — External processes ...

Follow-up

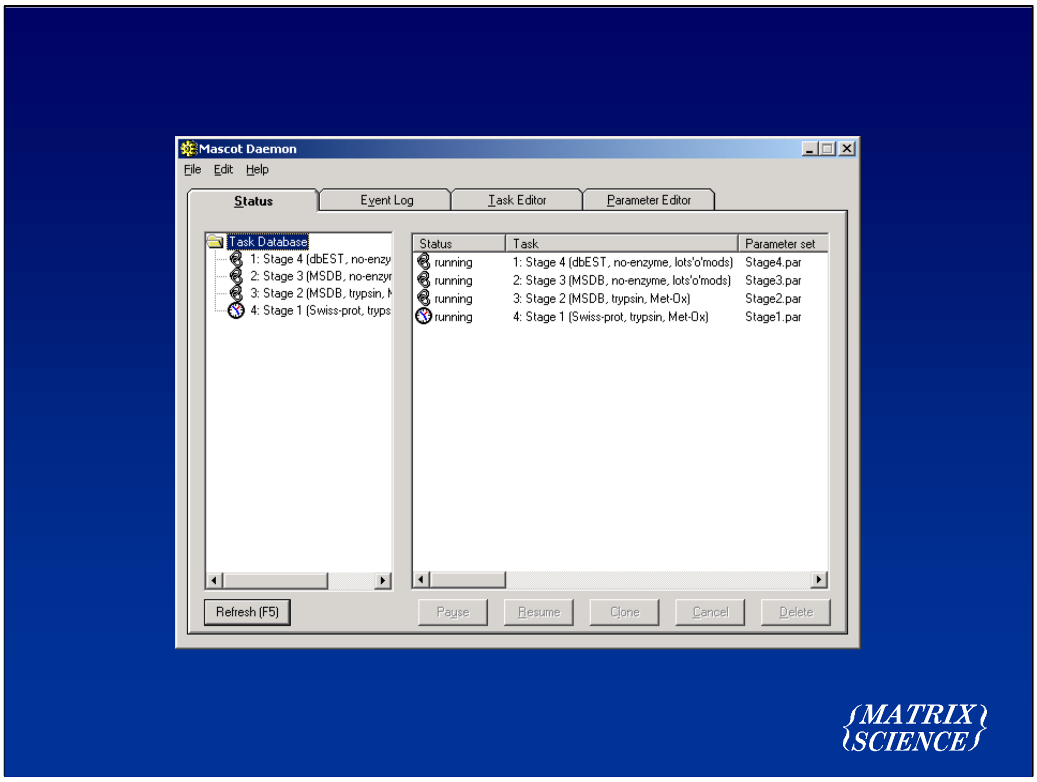
If probability that match is random > 1 in 100

[ ] Discard results

[ ] Repeat at intervals of 1 days

[✓] Pass data to 3 Stage 2 (MSDB, trypsin, Met-Ox)

Finally, we get to the start of the chain, stage 1. This is the fast Swiss-prot search. We are showing this as a real-time monitor task, but it could equally well be a batch task.

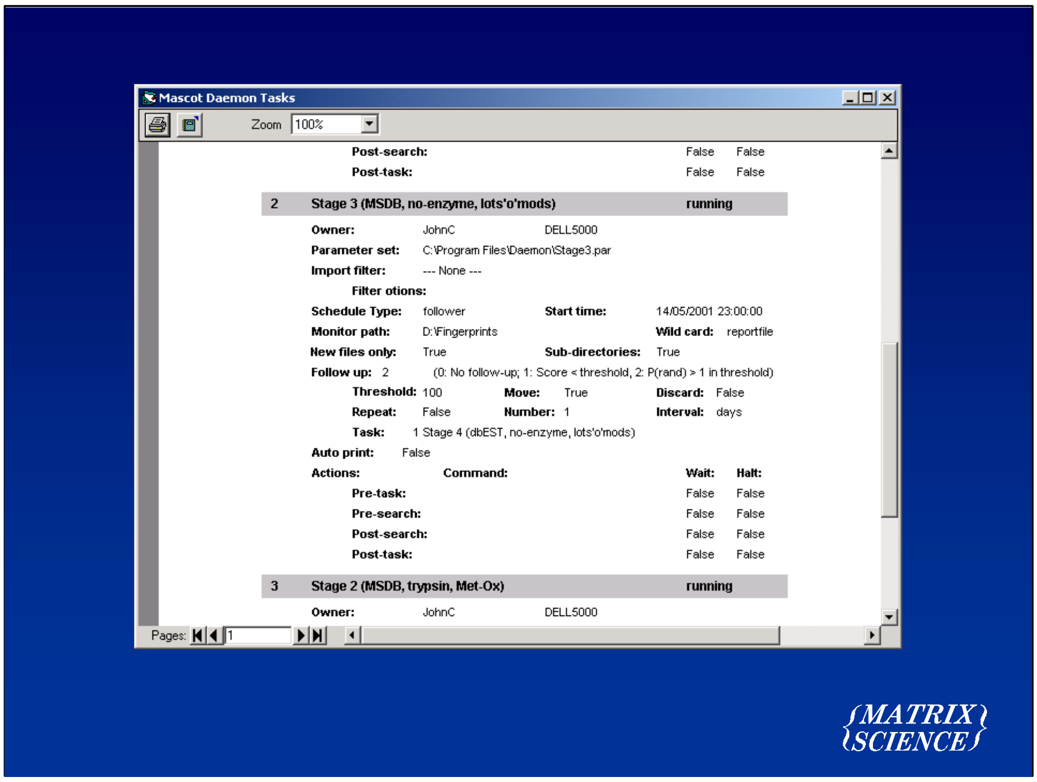


Status	Task	Parameter set
running	1: Stage 4 (dbEST, no-enzyme, lots'o'mods)	Stage4.par
running	2: Stage 3 (MSDB, no-enzyme, lots'o'mods)	Stage3.par
running	3: Stage 2 (MSDB, trypsin, Met-Ox)	Stage2.par
running	4: Stage 1 (Swiss-prot, trypsin, Met-Ox)	Stage1.par

When we finally press run, the whole chain swings into action, and data files will be passed along the chain until we get a match.

Although this may have sounded complicated, it really isn't.

Also, please realise that you don't have to set the chain up every time. Once a chain is set up, an unlimited number of real-time monitor or batch searches can feed into it. We could have 10 more tasks here, maybe one for each instrument in a lab, all feeding into the same task chain by specifying that unmatched data files should be passed to the stage 2 search.

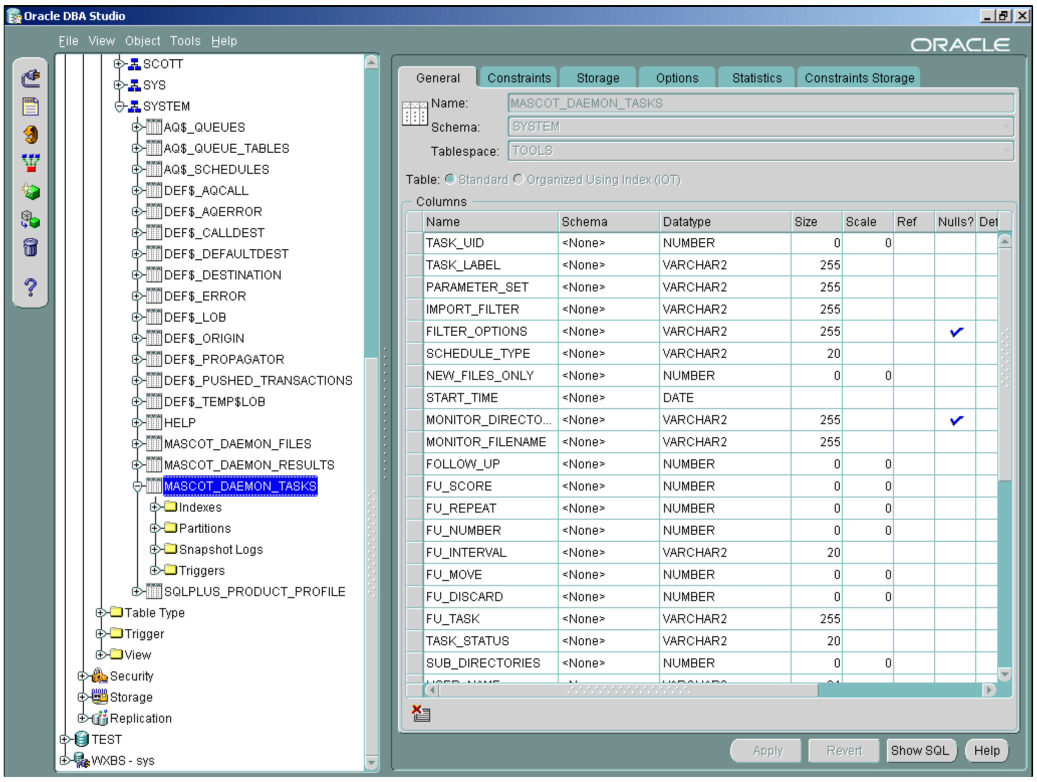


```
        Post-search:                                         False   False
        Post-task:                                           False   False

2   Stage 3 (MSDB, no-enzyme, lots'o'mods)                   running
    Owner:           JohnC           DELL5000
    Parameter set:   C:\Program Files\Daemon\Stage3.par
    Import filter:   --- None ---
        Filter otions:
    Schedule Type:   follower        Start time:      14/05/2001 23:00:00
    Monitor path:    D:\Fingerprints                  Wild card:   reportfile
    New files only:  True            Sub-directories: True
    Follow up:  2      (0: No follow-up; 1: Score < threshold, 2: P(rand) > 1 in threshold)
        Threshold: 100         Move:    True          Discard:   False
        Repeat:    False       Number:  1             Interval:  days
        Task:     1 Stage 4 (dbEST, no-enzyme, lots'o'mods)
    Auto print:    False
    Actions:            Command:                             Wait:   Halt:
        Pre-task:                                            False   False
        Pre-search:                                          False   False
        Post-search:                                         False   False
        Post-task:                                           False   False

3   Stage 2 (MSDB, trypsin, Met-Ox)                          running
    Owner:           JohnC           DELL5000
```

If you lose track of all the different tasks, it is now possible to view or print a report showing details for all the tasks on the tree. This was a widely requested feature, and is new in version 1.7



- SCOTT
- SYS
- SYSTEM
  - AQ$_QUEUES
  - AQ$_QUEUE_TABLES
  - AQ$_SCHEDULES
  - DEF$_AQCALL
  - DEF$_AQERROR
  - DEF$_CALLDEST
  - DEF$_DEFAULTDEST
  - DEF$_DESTINATION
  - DEF$_ERROR
  - DEF$_LOB
  - DEF$_ORIGIN
  - DEF$_PROPAGATOR
  - DEF$_PUSHED_TRANSACTIONS
  - DEF$_TEMP$LOB
  - HELP
  - MASCOT_DAEMON_FILES
  - MASCOT_DAEMON_RESULTS
  - MASCOT_DAEMON_TASKS
    - Indexes
    - Partitions
    - Snapshot Logs
    - Triggers
  - SQLPLUS_PRODUCT_PROFILE
- Table Type
- Trigger
- View
- Security
- Storage
- Replication
- TEST
- WXBS - sys

Name: MASCOT_DAEMON_TASKS
Schema: SYSTEM
Tablespace: TOOLS
Table: [✓ Standard] Organized Using Index (IOT)

Columns

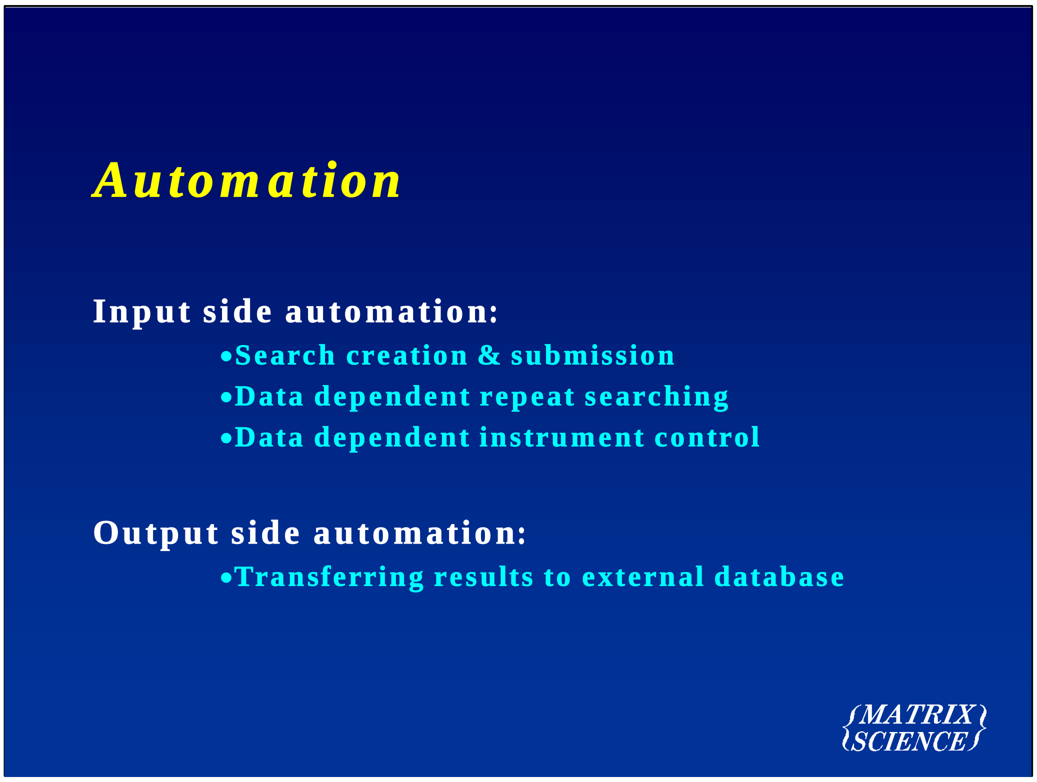
Name	Schema	Datatype	Size	Scale	Ref	Nulls?	Det
TASK_UID	<None>	NUMBER	0	0			
TASK_LABEL	<None>	VARCHAR2	255				
PARAMETER_SET	<None>	VARCHAR2	255				
IMPORT_FILTER	<None>	VARCHAR2	255				
FILTER_OPTIONS	<None>	VARCHAR2	255			✓	
SCHEDULE_TYPE	<None>	VARCHAR2	20				
NEW_FILES_ONLY	<None>	NUMBER	0	0			
START_TIME	<None>	DATE					
MONITOR_DIRECTO...	<None>	VARCHAR2	255			✓	
MONITOR_FILENAME	<None>	VARCHAR2	255				
FOLLOW_UP	<None>	NUMBER	0	0			
FU_SCORE	<None>	NUMBER	0	0			
FU_REPEAT	<None>	NUMBER	0	0			
FU_NUMBER	<None>	NUMBER	0	0			
FU_INTERVAL	<None>	VARCHAR2	20				
FU_MOVE	<None>	NUMBER	0	0			
FU_DISCARD	<None>	NUMBER	0	0			
FU_TASK	<None>	VARCHAR2	255				
TASK_STATUS	<None>	VARCHAR2	20				
SUB_DIRECTORIES	<None>	NUMBER	0	0			

Of course, the task master in a fully automated laboratory should really be the LIMS.

Underlying Daemon is a database containing three tables. One containing all the task information, one listing all the data files it knows about, and one listing all the result files.

By default, these tables live in a Microsoft Access database called TaskDB.mdb.

However, they can live in any ODBC compliant database engine, such as Oracle or SQL Server. So, if you have an Oracle based LIMS, the Daemon tables can be inside the LIMS. Instead of using the Daemon user interface, the LIMS can manipulate the fields in the tables directly. Daemon then becomes a specialised agent, executing searches at the direction of the LIMS.



Automation

Input side automation:

- **Search creation & submission**
- **Data dependent repeat searching**
- **Data dependent instrument control**

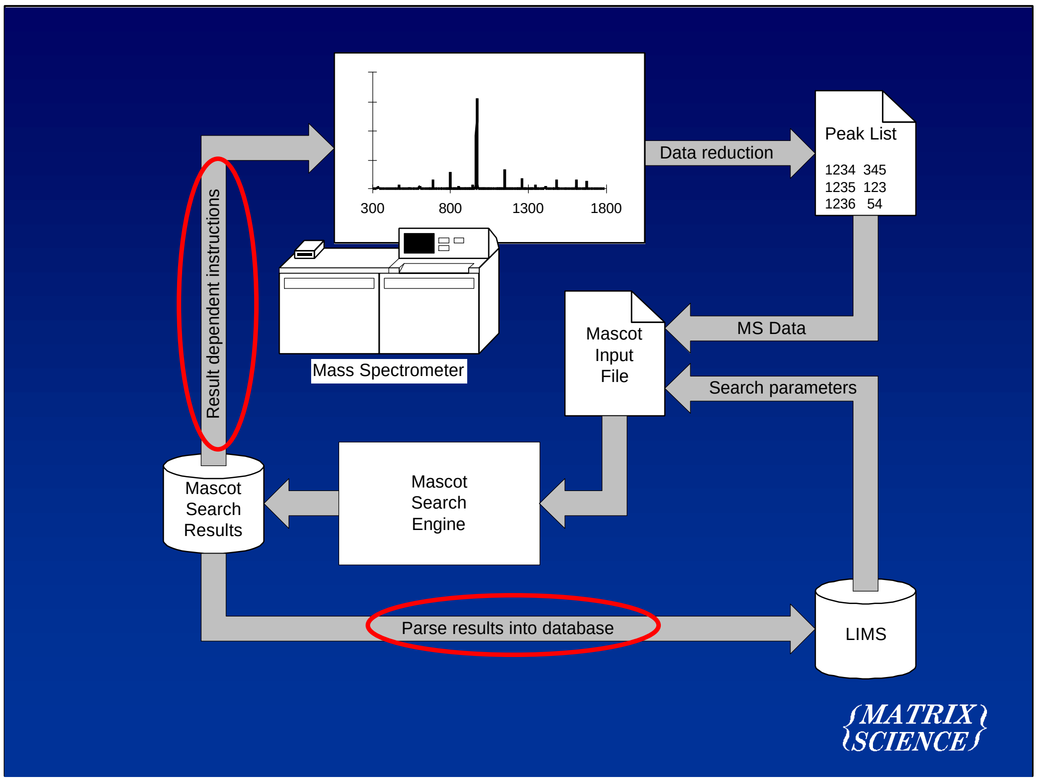
Output side automation:

- **Transferring results to external database**

MATRIX SCIENCE

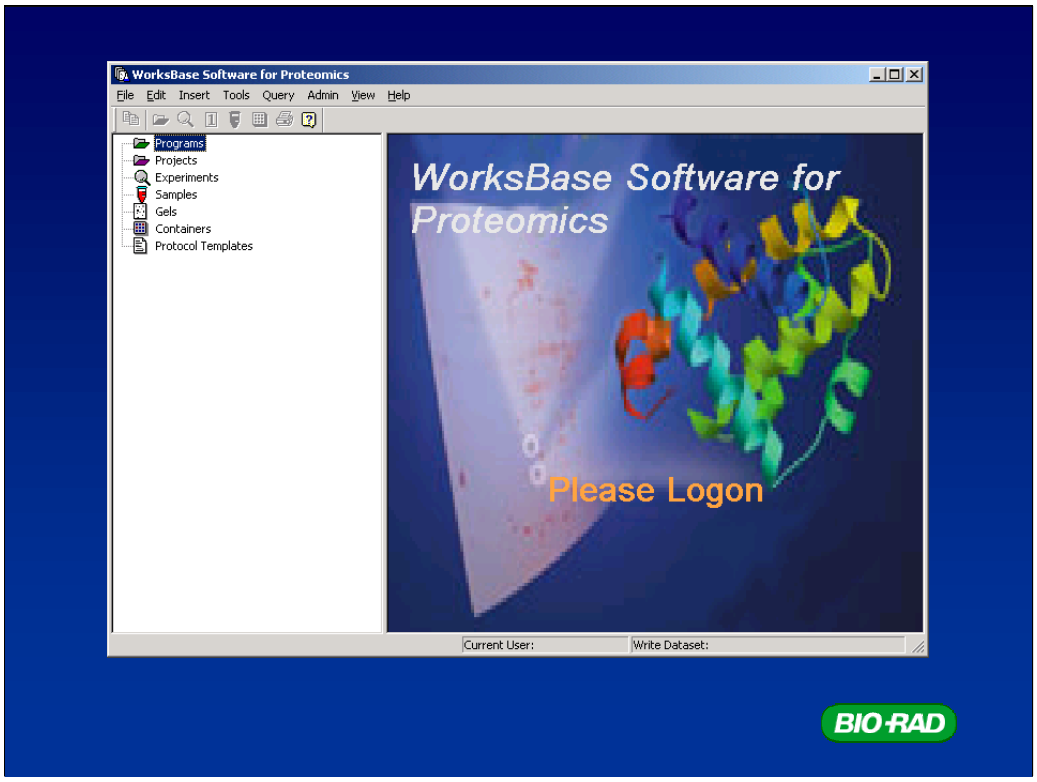
Let us focus on output side automation.

That is, doing something with the results other than just printing them out.

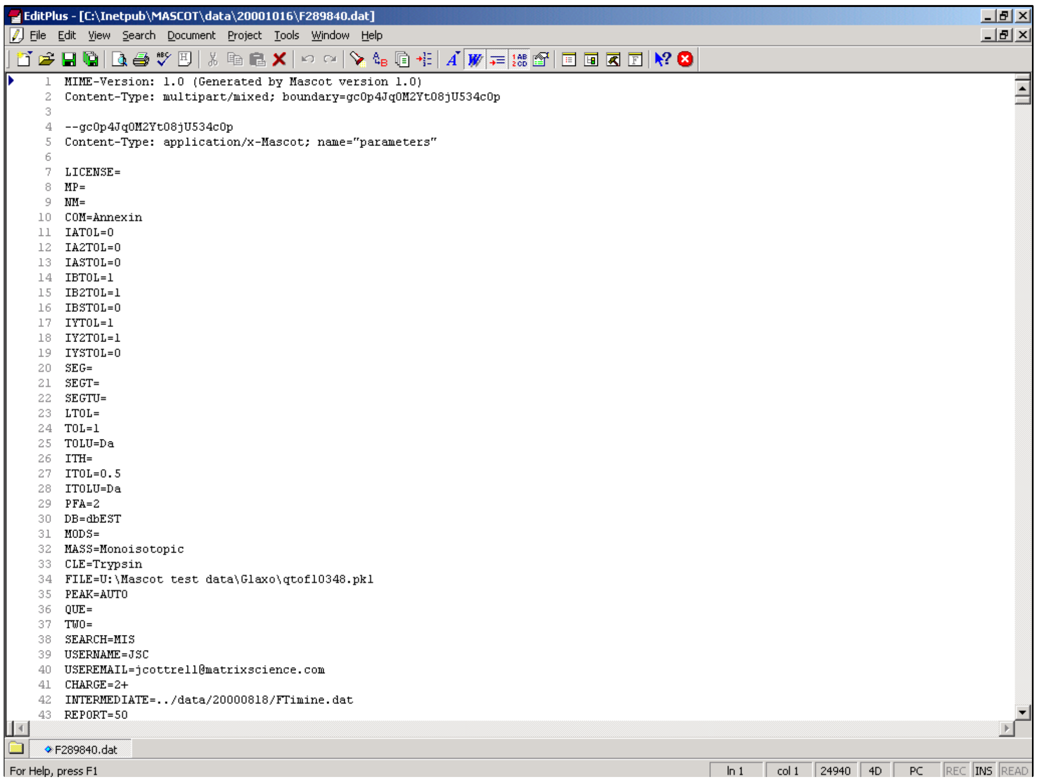


Now, we are focusing on these two connections.

This can be more difficult than input side automation, because everyone has different requirements. What do we mean by results and what do we mean by LIMS?



An example of a sophisticated proteomics oriented LIMS would be the WorksBase package from Bio-Rad. At Matrix Science, we are working with Bio-Rad to ensure that Mascot is fully integrated into WorksBase.


EditPlus - [C:\Inetpub\MASCOT\data\20001016\F289840.dat]
File Edit View Search Document Project Tools Window Help
F289840.dat
For Help, press F1
ln 1
col 1
24940
4D
PC
REC
INS
READ

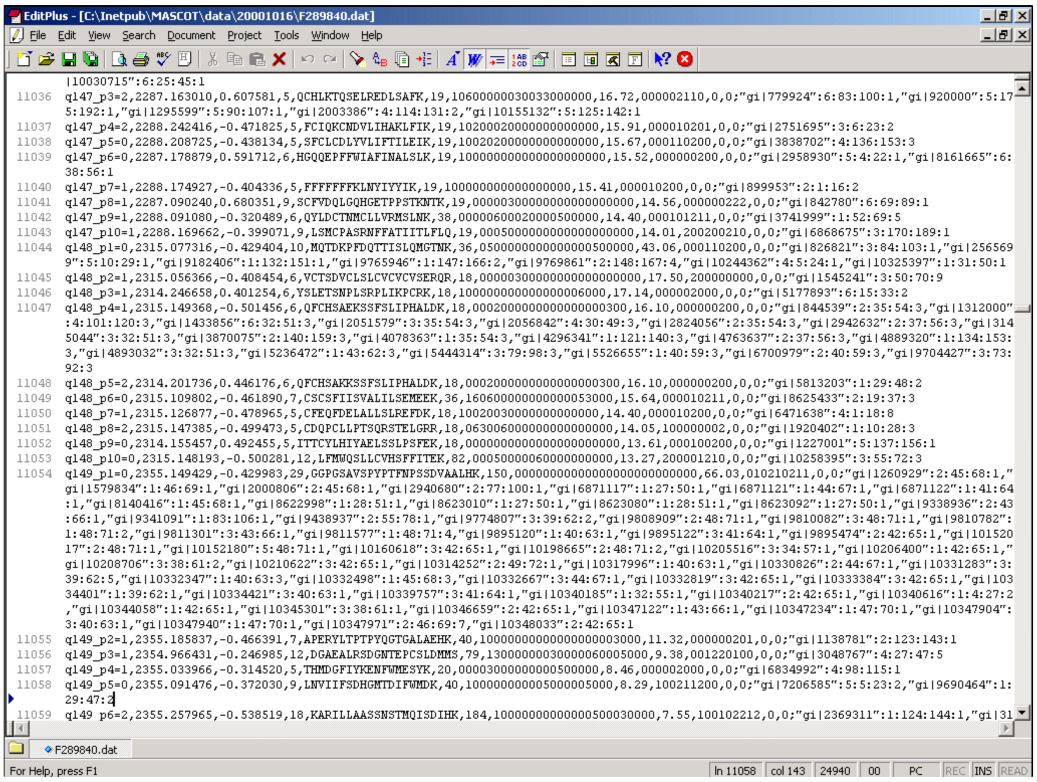
```
MIME-Version: 1.0 (Generated by Mascot version 1.0)
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08jU534c0p

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

LICENSE=
MP=
NM=
COM=Annexin
IATOL=0
IA2TOL=0
IASTOL=0
IBTOL=1
IB2TOL=1
IBSTOL=0
IYTOL=1
IY2TOL=1
IYSTOL=0
SEG=
SEGT=
SEGTU=
LTOL=
TOL=1
TOLU=Da
ITH=
ITOL=0.5
ITOLU=Da
PFA=2
DB=dbEST
MODS=
MASS=Monoisotopic
CLE=Trypsin
FILE=U:\Mascot test data\Glaxo\qtof10348.pkl
PEAK=AUTO
QUE=
TWO=
SEARCH=MIS
USERNAME=JSC
USEREMAIL=jcottrell@matrixscience.com
CHARGE=2+
INTERMEDIATE=../data/20000818/FTimine.dat
REPORT=50
```

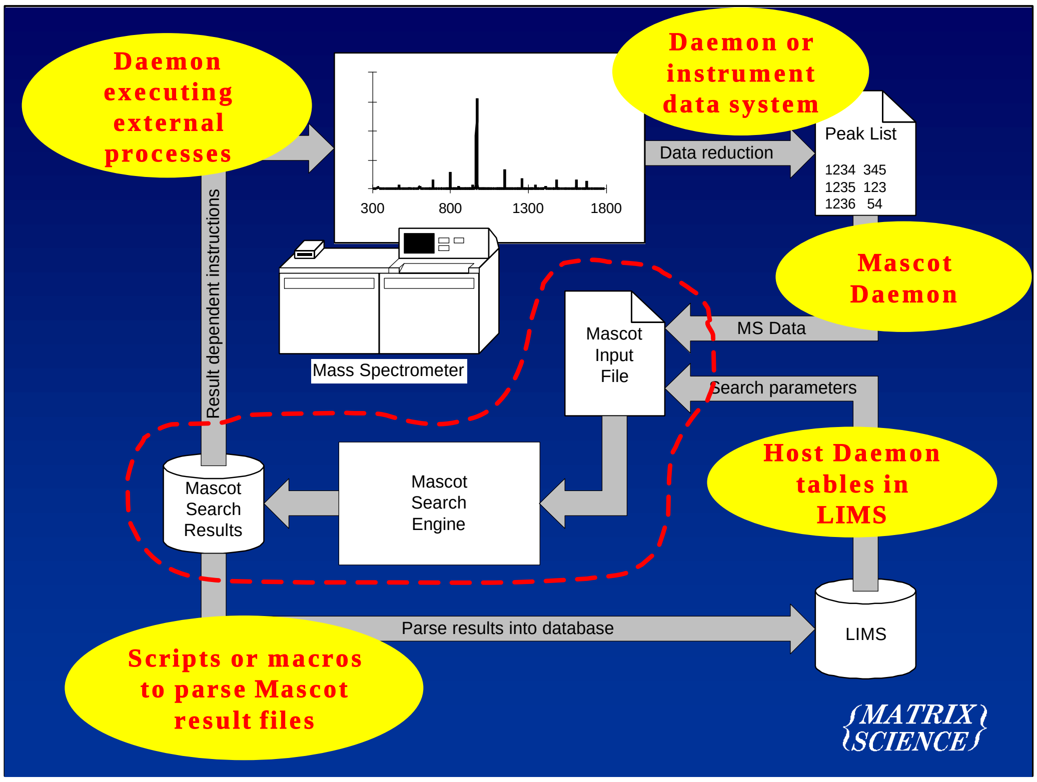
The Mascot architecture is very simple. All the search input, both mass spec data and parameters, is input to the search engine in a single file. At the completion of a search, all of the search results are written to disk as a single file. In both cases, the file is a MIME format, structured text file.

Here, you can see the simple label = value structure used for the search parameters.


EditPlus - [C:\Inetpub\MASCOT\data\20001016\F289840.dat]
File Edit View Search Document Project Tools Window Help

```
|10030715":6:25:45:1
q147_p3=2,2287.163010,0.607581,5,QCHLKTQSELREDLSAFK,19,10600000030033000000,16.72,000002110,0,0;"gi|779924":6:83:100:1,"gi|920000":5:17
5:192:1,"gi|1295599":5:90:107:1,"gi|2003386":4:114:131:2,"gi|10155132":5:125:142:1
q147_p4=2,2288.242416,-0.471825,5,FCIQKCNDVLIHAKLFIK,19,10200020000000000000,15.91,000010201,0,0;"gi|2751695":3:6:23:2
q147_p5=0,2288.208725,-0.438134,5,SFCLCDLYVLIFTILEIK,19,10020200000000000000,15.67,000110200,0,0;"gi|3838702":4:136:153:3
q147_p6=0,2287.178879,0.591712,6,HGQQEPFFWIAFINALSLK,19,10000000000000000000,15.52,000000200,0,0;"gi|2958930":5:4:22:1,"gi|8161665":6:
38:56:1
q147_p7=1,2288.174927,-0.404336,5,FFFFFFFKLNYIYYIK,19,100000000000000000,15.41,000010200,0,0;"gi|899953":2:1:16:2
q147_p8=1,2287.090240,0.680351,9,SCFVDQLGQHGETPPSTKNTK,19,000003000000000000000,14.56,000000222,0,0;"gi|842780":6:69:89:1
q147_p9=1,2288.091080,-0.320489,6,QYLDCTNMCLLVRMSLNK,38,000006000200005000000,14.40,000101211,0,0;"gi|3741999":1:52:69:5
q147_p10=1,2288.169662,-0.399071,9,LSMCPASRNFFATIITLFLQ,19,00050000000000000000,14.01,200200210,0,0;"gi|6868675":3:170:189:1
q148_p1=0,2315.077316,-0.429404,10,MQTDKPFDQTTISLQMGTNK,36,05000000000000000500000,43.06,000110200,0,0;"gi|826821":3:84:103:1,"gi|256569
9":5:10:29:1,"gi|9182406":1:132:151:1,"gi|9765946":1:147:166:2,"gi|9769861":2:148:167:4,"gi|10244362":4:5:24:1,"gi|10325397":1:31:50:1
q148_p2=1,2315.056366,-0.408454,6,VCTSDVCLSLCVCVCVSERQR,18,00000300000000000000000,17.50,200000000,0,0;"gi|1545241":3:50:70:9
q148_p3=1,2314.246658,0.401254,6,YSLETSNPLSRPLIKPCRK,18,10000000000000000006000,17.14,000002000,0,0;"gi|5177893":6:15:33:2
q148_p4=1,2315.149368,-0.501456,6,QFCHSAEKSSFSLIPHALDK,18,00020000000000000000300,16.10,000000200,0,0;"gi|844539":2:35:54:3,"gi|1312000"
:4:101:120:3,"gi|1433856":6:32:51:3,"gi|2051579":3:35:54:3,"gi|2056842":4:30:49:3,"gi|2824056":2:35:54:3,"gi|2942632":2:37:56:3,"gi|314
5044":3:32:51:3,"gi|3870075":2:140:159:3,"gi|4078363":1:35:54:3,"gi|4296341":1:121:140:3,"gi|4763637":2:37:56:3,"gi|4889320":1:134:153:
3,"gi|4893032":3:32:51:3,"gi|5236472":1:43:62:3,"gi|5444314":3:79:98:3,"gi|5526655":1:40:59:3,"gi|6700979":2:40:59:3,"gi|9704427":3:73:
92:3
q148_p5=2,2314.201736,0.446176,6,QFCHSAKKSSFSLIPHALDK,18,00020000000000000000300,16.10,000000200,0,0;"gi|5813203":1:29:48:2
q148_p6=0,2315.109802,-0.461890,7,CSCSFIISVALILSEMEEK,36,16060000000000000053000,15.64,000010211,0,0;"gi|8625433":2:19:37:3
q148_p7=1,2315.126877,-0.478965,5,CFEQFDELALLSLREFDK,18,10020030000000000000,14.40,000010200,0,0;"gi|6471638":4:1:18:8
q148_p8=2,2315.147385,-0.499473,5,CDQPCLLPTSQRSTELGRR,18,06300600000000000000,14.05,100000002,0,0;"gi|1920402":1:10:28:3
q148_p9=0,2314.155457,0.492455,5,ITTCYLHIYAELSSLPSFEK,18,000000000000000000000,13.61,000100200,0,0;"gi|1227001":5:137:156:1
q148_p10=0,2315.148193,-0.500281,12,LFMWQSLLCVHSFFITEK,82,00050000060000000000,13.27,200001210,0,0;"gi|10258395":3:55:72:3
q149_p1=0,2355.149429,-0.429983,29,GGPGSAVSPYPTFNPSSDVAALHK,150,000000000000000000000000000,66.03,010210211,0,0;"gi|1260929":2:45:68:1,"
gi|1579834":1:46:69:1,"gi|2000806":2:45:68:1,"gi|2940680":2:77:100:1,"gi|6871117":1:27:50:1,"gi|6871121":1:44:67:1,"gi|6871122":1:41:64
:1,"gi|8140416":1:45:68:1,"gi|8622998":1:28:51:1,"gi|8623010":1:27:50:1,"gi|8623080":1:28:51:1,"gi|8623092":1:27:50:1,"gi|9338936":2:43
:66:1,"gi|9341091":1:83:106:1,"gi|9438937":2:55:78:1,"gi|9774807":3:39:62:2,"gi|9808909":2:48:71:1,"gi|9810082":3:48:71:1,"gi|9810782":
1:48:71:2,"gi|9811301":3:43:66:1,"gi|9811577":1:48:71:4,"gi|9895120":1:40:63:1,"gi|9895122":3:41:64:1,"gi|9895474":2:42:65:1,"gi|101520
17":2:48:71:1,"gi|10152180":5:48:71:1,"gi|10160618":3:42:65:1,"gi|10198665":2:48:71:2,"gi|10205516":3:34:57:1,"gi|10206400":1:42:65:1,"
gi|10208706":3:38:61:2,"gi|10210622":3:42:65:1,"gi|10314252":2:49:72:1,"gi|10317996":1:40:63:1,"gi|10330826":2:44:67:1,"gi|10331283":3:
39:62:5,"gi|10332347":1:40:63:3,"gi|10332498":1:45:68:3,"gi|10332667":3:44:67:1,"gi|10332819":3:42:65:1,"gi|10333384":3:42:65:1,"gi|103
34401":1:39:62:1,"gi|10334421":3:40:63:1,"gi|10339757":3:41:64:1,"gi|10340185":1:32:55:1,"gi|10340217":2:42:65:1,"gi|10340616":1:4:27:2
,"gi|10344058":1:42:65:1,"gi|10345301":3:38:61:1,"gi|10346659":2:42:65:1,"gi|10347122":1:43:66:1,"gi|10347234":1:47:70:1,"gi|10347904":
3:40:63:1,"gi|10347940":1:47:70:1,"gi|10347971":2:46:69:7,"gi|10348033":2:42:65:1
q149_p2=1,2355.185837,-0.466391,7,APERYLTPTPYQGTGALAEHK,40,10000000000000000003000,11.32,000000201,0,0;"gi|1138781":2:123:143:1
q149_p3=1,2354.966431,-0.246985,12,DGAEALRSDGNTEPCSLDMMS,79,13000000030000060005000,9.38,001220100,0,0;"gi|3048767":4:27:47:5
q149_p4=1,2355.033966,-0.314520,5,THMDGFIYKENFWMESYK,20,00003000000000500000,8.46,000002000,0,0;"gi|6834992":4:98:115:1
q149_p5=0,2355.091476,-0.372030,9,LNVIIFSDHGMTDIFWMDK,40,10000000000500000005000,8.29,100211200,0,0;"gi|7206585":5:5:23:2,"gi|9690464":1:
29:47:2
q149 p6=2,2355.257965,-0.538519,18,KARILLAASSNSTMQISDIHK,184,100000000000000500030000,7.55,100102212,0,0;"gi|2369311":1:124:144:1,"gi|31
```

Here is part of the block which contains the peptide match results. It looks a little busy, but this kind of thing is dead easy to parse into a database. The tricky bit is deciding what information you want to extract.


Daemon executing external processes
Daemon or instrument data system
300
800
1300
1800
Data reduction
Peak List
1234 345
1235 123
1236 54
Mascot Daemon
MS Data
Mascot Input File
Search parameters
Mass Spectrometer
Result dependent instructions
Mascot Search Results
Mascot Search Engine
Host Daemon tables in LIMS
LIMS
Parse results into database
Scripts or macros to parse Mascot result files
MATRIX SCIENCE

Returning to our block diagram to summarise: We have mechanisms in place to implement a fully automated protein identification pipeline. Either on a small scale, with one instrument, or on a large scale, with many.

The core is the Mascot server.

Generating peak lists currently relies on the instrument data system, but this will change in the not too distant future.

Mascot Daemon does a pretty good job of automating search submission.

We can even integrate Daemon with a LIMS by hosting its tables in the LIMS database.

The facility to execute external processes provides a crude mechanism for data dependent instrument control, but will generally involve scripting or programming by the user and depends on the instrument control software having the appropriate hooks in place.

Finally, parsing of result information into a LIMS or project database can be easy or can be difficult. It all depends on the questions you are asking.

{MATRIX SCIENCE}

http://www.matrixscience.com