

DTASelect and Contrast Users' Manual

David Tabb*

December 16, 2002

Abstract

DTASelect provides a means by which complex SEQUEST results can be filtered, organized, and viewed. A single sample may produce tens of thousands of tandem mass spectra. Manually perusing and selecting SEQUEST matches among such a mass of data risks inconsistency. DTASelect allows the user to set complex criteria for acceptance or rejection of individual spectrum results. It also features rules for dealing with multiple, identical peptide matches and for removing proteins that are insufficiently evidenced. It provides its sorted and filtered summary as HTML and text documents for easy review and also offers several auxiliary reports. DTASelect is a powerful tool for automatic analysis of complex mixture tandem mass spectrometry.

Contrast is a tool for differentiating multiple runs of data. A user may run a control sample and experimental sample through the SEQUEST process and then use Contrast to highlight the differences in the protein compositions of the two samples. An alternate use allows the user to determine the reliability of individual identifications in a sample by stepping through multiple criteria sets. The tool can also be used with multiple data sets and multiple criteria sets simultaneously. Results are stored in a unified web browseable summary and in HTML files representing each set under each set of criteria. The Contrast program leverages the power of DTASelect's criteria across multiple data sets and criteria sets.

1 Introduction

As tandem mass spectrometry emerges as a primary tool for proteomics, the complexity of the data produced through this technique has increased dramatically. While SEQUEST aids enormously in the identification of individual spectra, complementary tools to summarize SEQUEST results have lagged behind. When faced with multidimensional chromatography results in particular, the researcher needs assistance in grouping and selecting significantly matching spectra from more ambiguous results.

DTASelect and Contrast fill this niche. DTASelect acts as an organizer and filter, grouping together related spectra by protein and removing those that do not pass basic data quality criteria. Contrast provides meta-analytical capability, helping the user to tune criteria to particular data

*dtabb@u.washington.edu

sets and to compare results from multiple samples. Together, these tools dramatically increase the scope of analyses possible through tandem mass spectrometry and SEQUEST.

This manual will assume that the reader is familiar with most features of SEQUEST operation. A brief discussion of SEQUEST-related files, however, is warranted. SEQUEST is configured by modification of the `sequest.params` file. The information in this file includes the path of the FASTA sequence database and the differential modifications specified for the search. SEQUEST's analysis of an individual spectrum (`.dta`) file is recorded in file with the extension `".out."` These files include loci and sequences which match to this spectrum as well as the scores characterizing each match, such as `XCorr`, `deltCN`, and `Sp`. A single MudPIT (LC/LC/MS/MS) experiment may yield thousands of `.out` files in several subdirectories. DTASelect groups together these SEQUEST results to produce a unified report.

Comparing the protein content of two samples has been a daunting task. Because DTASelect automates the process of filtering the protein lists, however, the challenge is greatly reduced. Contrast uses DTASelect's filtering capacity to create protein lists for multiple samples and then compares the proteins found in each. What could otherwise be a nightmarish session with a large spreadsheet becomes as simple as specifying the directories to be compared and waiting a minute.

2 Requirements

DTASelect and Contrast can be run on any Java 1.1-compliant virtual machine (VM). These VMs are widely available for many platforms. The most widespread VM installs with Microsoft's Internet Explorer 4.0 and later. If you use Microsoft Windows and want to determine whether or not you have this VM installed, type `"jview"` at a command line. The IBM Java 2 Developer's Kit for Linux and Windows v1.3 was used for compilation and execution during development.

DTASelect draws information from three sources while running: SEQUEST's output files, the `sequest.params` file, and the sequence database. SEQUEST output (`.out`) files are typically present in subdirectories off of the directory holding the `sequest.params` file. The database used for the SEQUEST search is identified from the `sequest.params` file, and it is expected to be in the same location as indicated in `sequest.params`. If a precompiled `.bin` database was used, the program will assume that a standard FASTA database with the same name except for the `.bin` extension is present. DTASelect has been tested with many versions of SEQUEST and will work with the `.out` files produced by the SEQUEST in Bioworks 3.

If a version of SEQUEST producing unified output files has been employed, the directory may contain `.sqt` files rather than `.out` files. DTASelect will check the current directory for `.sqt` files and read these in preference to `.out` files if they are detected.

DTASelect offers support for Matrix Science's Mascot algorithm output. Identifications and configuration are read from MIME-formatted `.dat` files. For information on this support, see Appendix A.

Contrast requires that DTASelect has been run in each directory to be included in its analysis. When DTASelect is first run in a directory, it creates a file entitled `DTASelect.txt`. This file must be present in each directory included in a Contrast analysis. Contrast reads from the `Contrast.params` file in the current directory to determine the type of analysis to be performed.

The programs will expect the SEQUEST output file names to follow this pattern:

`[DATfilename].[firstscannumber].[secondscannumber].[chargestate].out`

Extraneous periods or different file name organization will likely confuse the program.

3 DTASelect

Once the .dta files are extracted and SEQUEST has been run on the spectra, DTASelect can be used. The first run of DTASelect on any data set takes the most time. During this first pass, the program must read thousands of individual .out files from the disk as well as sift through the entire selected sequence database. Subsequent runs of DTASelect are far faster than its initial execution in a directory.

3.1 Installation

DTASelect installation begins when its files are copied to a directory on the computer where it will be run. The DTASelect.ini file (see Figure 1) should be modified to configure which servers will be used for viewing spectra and sequence coverages, which CGIs are available (for example, is Flicka installed rather than Web_retrieve?), and how long the locus names given by SEQUEST are (different versions of the program crop the name to different numbers of characters). A batch file or script to start the program should be created (example batch files for Windows are included in the distribution). On Windows machines, this batch file should be called "DTASelect.bat" and be stored in C:\WINNT or C:\WINDOWS. On a Linux box, the script could be stored in /usr/bin or another location located on the system path. The script should look something like this:

```
C:\IBMJDK\java -cp "C:\DTASelect" DTASelect %*
```

In this line, C:\IBMJDK\java is the filename of IBM's Java Virtual Machine (in this case, a Windows version is used). The cp option ensures that the DTASelect installation directory, in this case C:\DTASelect, is on the java classpath. DTASelect specifies that this program is run (Contrast will specify that Contrast runs instead). The %* ensures that options a user adds to the command line when starting the script are passed to the program (for Windows NT and derived operating systems). Some virtual machines allow the user to specify the memory available to DTASelect. It is helpful to create one script to start DTASelect, one for Contrast, and a third to start MoreFASTA (if used). All work from the same classpath but start from different classes within the directory.

3.2 Execution

Once the program is installed, users can start it by running the created script or batch files. The program should be run from the directory holding the sequest.params file. Typically, the subdirectories of spectra and SEQUEST output files are located here. *If the .out files themselves are located in the directory where DTASelect is run (rather than in a subdirectory), the --here or - . options should be used.*

The options selected by the user should be appended at the end of the line. The order of the options does not matter, so long as the numerical arguments, when required, follow the appropriate option. If an option is left off the command line, its default value is used. Users may specify alternate default options in a file entitled DTASelect.params in the directory where the .dta files are located. This file's format is a single line of options entered as they would be on the command

```

# DTASelect configuration file
# List the SEQUEST CGI server's name or IP address below
server-name                localhost
# Which CGI will display spectra?
spectrum-display           /cgi-shl/displayions.exe
# Which CGI will display SEQUEST output files?
output-display             /cgi-shl/showout.pl
# Which CGI will show sequence coverage?
sequence-coverage          /cgi-shl/flicka.pl
# Which CGI will handle protein validation?
protein-validation         /cgi-bin/EvalocusA
# Which server and CGI should handle BLAST searching?
BLAST-handler              www.ncbi.nlm.nih.gov:80/blast/Blast.cgi
# What additional options should be sent to BLAST?
BLAST-args
# Is this a Windows or UNIX computer?
server-type                 Windows
# When newer CGIs are used, which server should be used?
cgi-server-name            localhost
# What is the length at which my SEQUEST truncates locus names?
locus-length-cutoff        21
# Where is Mascot installed?
mascot-path                 c:/inetpub/mascot

```

Figure 1: Sample DTASelect.ini file

line. It is possible to run DTASelect without any specified options; all of the default values are used in this case. Valid options are listed in Tables 1, 2, 3, and 4.

Option	Default	INDIVIDUAL SPECTRUM FILTERS:
-1 #	1.8	Set lowest +1 XCorr
-2 #	2.5	Set lowest +2 XCorr
-3 #	3.5	Set lowest +3 XCorr
-d #	0.08	Set lowest DeltCN
-c #	1	Set lowest charge state
-C #	3	Set highest charge state
--mz #		Set minimum precursor m/z
--MZ #		Set maximum precursor m/z
-i #	0.0	Set lowest proportion of fragment ions observed
-s #	1000	Set maximum Sp ranking
-m 0		Require peptides to be modified
-m 1	X	Include peptides regardless of modification
-m 2		Exclude modified peptides
-y 0	X	Include peptides regardless of tryptic status
-y 1		Include only half- or fully tryptic peptides
-y 2		Include only fully tryptic peptides
-v -1		Keep "N" peptides, discard all others
-v 0	X	Ignore manual validation info
-v 1		Keep "Y" peptides, discard "N" peptides
-v 2		Keep "Y" and "M" peptides, discard "N" peptides
-v 3		Keep "Y" and "M" peptides, discard "N" and "U" peptides

Table 1: DTASelect Spectrum Filters: # symbols indicate a numerical parameter follows while \$ symbols indicate character or string parameters follow. Changes can be made at the command line or in a DTASelect.params file. Each spectrum must pass all filters.

The default criteria for DTASelect are designed to retain only the proteins for which ample evidence is present and to include results only from those .out files that stand on their own merits. The default criteria for DTASelect are conservative; if a protein is listed under the defaults, there is strong evidence that it is present in the sample. The default criteria are listed in the second column of Tables 1 and 3.

Many of the filters bear further investigation. The individual spectrum filters are fairly straightforward, setting minimum or maximum limits on the basic parameters of each identification. The extended filters, however, may not seem terribly useful at first glance. If one is attempting to find spectra that SEQUEST does not identify well, the maximum XCorr settings can come in handy. The sequence-based filters are helpful for special uses. For example, if one wants to exclude peptides that result from incomplete tryptic digestions, using `-Sec RK` will manage this task. Identifying peptides that carry certain motifs or which constitute affinity tags can be accomplished with the `-Sip` option. Creating customized cleavage filters can be handled with `-Stn` and `-Stc`. Filtering to include only peptides that would result from a combined CNBr / Trypsin digest can be

Option	EXTENDED FILTERS (off by default):
-Sic \$	Sequences must contain all of these characters (excludes C terminal residue)
-Sip \$	Sequences must contain this pattern
-Sec \$	Sequences must not contain any of these characters (excludes C terminal residue)
-Stn \$	Preceding residue must be one of these
-Stc \$	C terminal residue must be one of these
-Smn #	Sequence must be at least this length
-Smx #	Sequence must be no longer than this
-X1 #	Set highest +1 XCorr
-X2 #	Set highest +2 XCorr
-X3 #	Set highest +3 XCorr

Table 2: Extended Spectrum Filters: # symbols indicate a numerical parameter follows while \$ symbols indicate character or string parameters follow. Extended filters only apply if they are specified).

accomplished with `-Stn M-RK -Stc MRK`, where the `M-RK` requires that the residue preceding this peptide in the database must be either the start of the protein sequence or a met, arg, or lys residue. The `-Smn` and `-Smx` filters can be used to specify the minimum and maximum lengths of permissible sequences, respectively. The extended filters are not always useful, but they can be helpful for seeking out particular sets of peptides.

3.3 Program Flow

DTASelect proceeds in three phases: summarizing, filtering, and reporting.

Summarization is carried out by examining each `.out` file in every subdirectory of the current directory. The identifications are stored in memory. Any sequence match with a `DeltCN` of 0.0 (that is, any identification that ties the best identification in `XCorr`) will be stored for each `.out` file. Since multiple proteins may contain the same peptide, some spectra will be associated with several proteins. When all files have been read, the `.out` file information is sorted by locus. This list is used to produce a list of genetic loci or proteins present, with a list of evidentiary `.out` files associated with each locus. The summary of `.out` files is written to the `DTASelect.txt` file. Subsequent runs of DTASelect will not read the `.out` files but rather will import the `DTASelect.txt` file.

The next stage of DTASelect removes low-scoring and redundant information. The three-step process removes spectra that do not meet the basic criteria, ejects redundant spectrum copies, and then removes loci that do not pass the necessary criteria:

First, the individual spectrum filters are applied to each peptide. If a peptide has an insufficient `XCorr` score, for example, it will be removed. The majority of criteria available in DTASelect are parameters describing the removal of individual spectrum results at this stage. A spectrum must pass all of these filters to be included.

If peptide-level validation has been selected (via the `-v` option), individual spectrum filters may be bypassed for some or all peptides. If a peptide has not been validated, then its spectrum is

Option	Default	LOCUS FILTERS:
-t 0		Do not purge duplicate spectra for each sequence
-t 1		Purge duplicate spectra on basis of total intensity
-t 2	X	Purge duplicate spectra on basis of XCorr
-V -1		Keep "N" peptides, discard all others
-V 0	X	Ignore manual validation info
-V 1		Keep "Y" proteins, discard "N" proteins
-V 2		Keep "Y" and "M" proteins, discard "N" proteins
-V 3		Keep "Y" and "M" proteins, discard "N" and "U" proteins
-u	false	Include only loci with uniquely matching peptides
-o	false	Remove proteins that are subsets of others
--mw #		Set minimum protein molecular weight
--MW #		Set maximum protein molecular weight
--pi #		Set minimum protein isoelectric point
--PI #		Set maximum protein isoelectric point
-e \$		Remove proteins with IDs matching this string
-E \$		Include only proteins with IDs matching this string
-l \$		Remove proteins with descriptions including this word
-L \$		Include only proteins with descriptions including this word
-M #	0	Set minimum modified peptides per locus criterion
-r #	10	Show all loci with peptides that appear this many times
-p #	2	Set minimum peptides per locus criterion

Table 3: DTASelect Locus Filters: # symbols indicate a numerical parameter follows while \$ symbols indicate character or string parameters follow. Filter changes can be made at the command line or in a DTASelect.params file. Each locus must pass either -r or -p and then any -e, -E, -l, -L, -u, and -o options specified.

Option	UTILITIES:
--nofilter, -n	Do not apply any criteria
--copy	Create script to copy selected spectra and IDs (or subset SQT and MS2 files)
--GUI	Report through GUI instead of output files
--compress	Create .IDX and .SPM files from spectra
--CGI	Use the replacement CGIs instead of the classic ones
--Mascot	Draw peptide IDs from Mascot .dat files instead
--BE	Produce Bird's Eye view of proteins found
--class	Classify proteins according to Classifications.txt
--aux	Incorporate auxiliary protein information from AuxInfo.txt
--XML	Save XML report of filtered results
--DB	Save in format for database import
--chroma	Save chromatography report
--similar	Save protein similarity table
--align	Save sequence alignment report
--mods	Save modification report
--help, -h	Print this list of options
--here, -.	Include only IDs in current directory

Table 4: DTASelect Utilities

evaluated by the above rules unless `-v 3` has been specified. A quick summary of the command-line rules appears in Table 5.

Command Line	Symbolic Description	Longer Description
<code>-v -1</code>	<code>+N -YMU</code>	Keep "N" peptides, discard all others
<code>-v 0</code>	Ignore	Ignore manual validation info
<code>-v 1</code>	<code>+Y -N</code>	Keep "Y" peptides, discard "N" peptides
<code>-v 2</code>	<code>+YM -N</code>	Keep "Y" and "M" peptides, discard "N" peptides
<code>-v 3</code>	<code>+YM -NU</code>	Keep "Y" and "M" peptides, discard "N" and "U" peptides

Table 5: Manual Validation Handling in DTASelect

Secondly, if multiple .dta files are identified to match precisely the same sequence and charge state, a spectrum redundancy exists. If the user has chosen to do so, only the best representative of these redundant spectra will be retained on the basis of highest XCorr or total intensity. Either all (`-t 0`) or one (`-t 1` or `-t 2`) of each set of identical spectra will be kept.

Finally, the loci in memory are purged. Each protein must first pass either the `-r` or `-p` filter. Next, the `-e`, `-E`, `-l`, `-L`, and `-u` options are used to trim the list. Finally, the `-o` option goes into effect. A full description of these options follows.

Loci are retained if they pass either of two rules. The first (`-p #`) keeps loci that have a sufficient number of distinct peptides remaining in evidence (these peptides will all have different

sequences or charge states unless the `-t 0` option has been used). The other rule (`-r #`) keeps loci which are supported by at least one peptide which has been identified redundantly enough to be deemed reliable. For example, a locus represented by two copies of the same spectrum will be retained under `-p 2 -r 2` settings by passing the redundancy rule (`-r`) but not the number of distinct spectra rule (`-p`). The more typical way to pass this requirement is that a protein has three peptides when only two are required (`-p 2`; default setting). The redundancy rule is set strictly enough by default (10 copies of a spectrum required) that few proteins should pass it. The `-M` rule is quite similar to the `-p` rule except that only peptides with dynamic modifications are counted, and 0 are required by default. These quota rules filter out large numbers of proteins for which scant evidence is present.

Two more filters remove proteins by simple criteria. If the user has chosen the `-u` option, each locus will be examined to see whether any of the representative peptides are unique to that locus. If at least one remains, the locus is retained. Otherwise, the locus is removed. The exclude option (`-e $`) may be used to remove groups of proteins with names containing the specified key phrase, while the include name option (`-E $`) requires that all proteins match a particular string. These options are most useful when the sequence database has been created by merging together two disparate ones and the database IDs from each component database use different patterns. The exclude option could be used, for example, to remove all proteins with IDs that start with “gi” by specifying `-e gi` on the command line or in `Contrast.params`. The `-l` and `-L` options are similar except that they are applied to the locus descriptions rather than the locus names themselves and are case insensitive. Not all databases feature extensive descriptions, but someone searching for proteins with “kinase” in the description could use these options to list just these proteins. All of these filters are only in effect if explicitly specified.

The final locus filter is more complex to describe. Proteins are then grouped by combined sequence coverage. If two different proteins have identical sequence coverage, the proteins will be grouped together for DTASelect’s output. If the user has selected the parsimony (`-o` for William of Occam) option, subset proteins are removed. A subset protein is one for which all peptides are found in another protein. For example, if protein A has peptides 1, 2, and 3, protein B with only peptides 1 and 2 is a subset protein. This option is likely to be most useful under `-p 1` (single peptide per locus) criteria or under very large sequence databases. If the user does not indicate that parsimony should be used, it is off by default. The only proteins that will remain in the list are those that pass either `-r` or `-p`. Next loci must pass the `-M` rule followed by `-e`, `-E`, `-l`, `-L`, `-u`, and finally `-o` if specified.

Again, if manual validation criteria are put in place (with the `-V` option), the locus filtering may be handled differently. The summary of the peptide-level `-v` option in Table 5 is an accurate assessment of the locus-level `-V` option if one substitutes the word “protein” for “peptide.”

After filtering is complete, the percentage of residues in each protein sequence that are represented by at least one peptide is calculated. The output files are created, and if the user has specified the `--GUI` option, the graphical user interface is displayed (see Figure 2). If the user has specified the `--CGI` option, DTASelect will link to improved CGIs rather than those that ship with SEQUEST (see Table 6 for more information).

Often the proteins in database have been categorized by biological function (such as the MIPS annotation of the *Saccharomyces* protein database). DTASelect can group proteins that are found in a sample by such classes when the `--class` option is employed. The program will examine the contents of the `Classifications.txt` file to determine how to group proteins (see Figure 3 for a brief

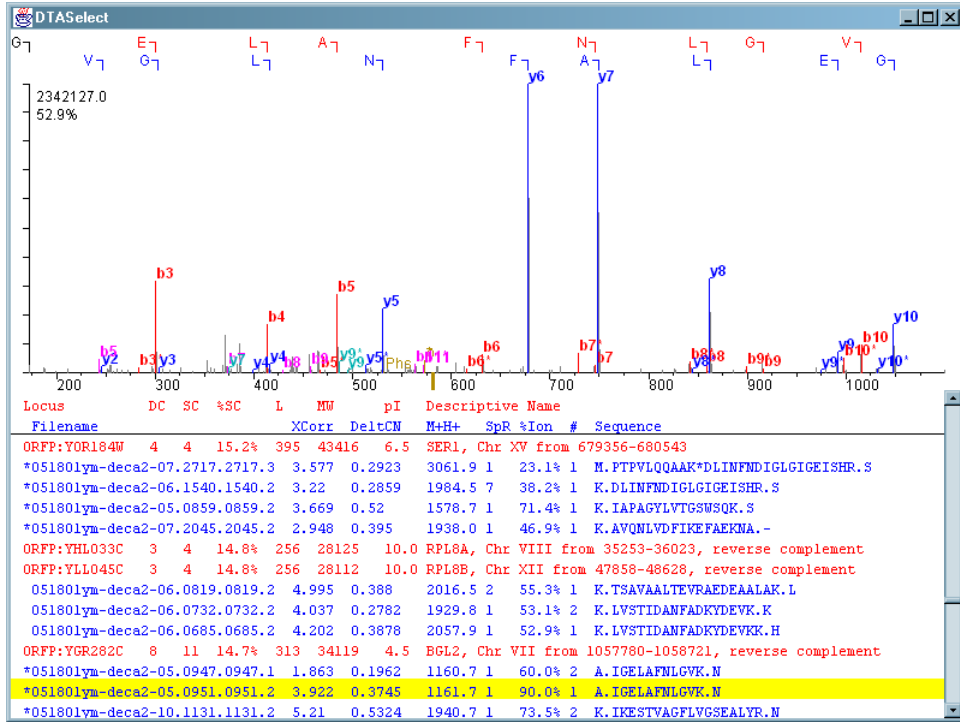


Figure 2: Sample of DTASelect’s GUI displaying a spectrum: Spectra can be viewed directly from the compressed files or from .dta or .ms2 files. Clicking on the name of a spectrum in the list will display it in the viewing window. For a description of the spectrum view, see Appendix B.

Name	Description	Author
Show	Displays spectra via SpectrumApplet	Hayes McDonald
SeqCov	Displays depth of sequence coverage	Johannes Graumann
Evalocus	Allows changes in locus validation states	Johannes Graumann
Evalpeptide	Allows validation of peptide identifications	Hayes McDonald

Table 6: Auxiliary CGIs designed for use with DTASelect. Note that these programs are not included in the base distribution of DTASelect and may require separate licenses for use. For more information about the SpectrumApplet spectrum viewer, see Appendix B.

example). If `--class` is used with the `--DB` option, a file entitled “DB-Classes.txt” is created along with the others.

class	1	Huge Proteins
class	2	Big Proteins
class	3	Sizeable Proteins
class	4	Small Proteins
class	5	Tiny Proteins

```
YLR106C 1  
YKR054C 1
```

...

Figure 3: Sample Classifications.txt file. Six classes are described, numbered 1-6 (up to 126 are possible). Individual proteins are assigned to a group by having their locus names followed by the number corresponding to the appropriate group. Here, YLR106C and YKR054C are assigned to group 1, “Huge Proteins.” More loci would follow in the full file. Proteins which are not associated with a group are assigned to group 127, the unclassified proteins.

These classifications are particularly useful when the option to produce a Bird’s Eye view of the proteins is employed (see Figure 4). The proteins found in the sample are visualized as small blocks, with the colors representing the sequence coverages found for each protein. This display can be configured by means of the BirdsEye.ini file, read from the directory where DTASelect is run (see Figure 5 for an example).

DTASelect can also incorporate data from external programs into its output. The `--aux` option will cause DTASelect to check the file AuxInfo.txt for information pertaining to each locus observed in a sample. In this way, algorithms for quantifying protein content or providing other data can be integrated with DTASelect. A heading for these additional columns will be added to the DTASelect.html file, and the values entered will appear before the description of each protein. For an example of the AuxInfo.txt format, see Figure 6.

Users interested in storing DTASelect’s results in a database environment may use its `--DB` option to produce the DB-Proteins.txt, DB-Peptides.txt, and DB-Prot2Pep.txt files. The first two files list the proteins and peptides remaining after DTASelect’s filtering (with duplicate proteins, those that have been grouped to others with identical sequence coverage, flagged in the final column of the DB-Proteins.txt file). The third shows how which proteins link to which peptides. The files are tab-delimited and can be opened in most spreadsheets and databases.

To compare the proteins found in a run against the full set of proteins in a database, one can use the “MoreFASTA” tool. The program takes as an argument the name of a FASTA database. If given the database yeast_orfs, it produces a file called yeast_orfs.DB. Each line of the DB file is similar, but not identical to, the format of the DB-Proteins file and contains the locus name, sequence length, molecular weight, calculated pI, description, and residue composition of each protein in the database.

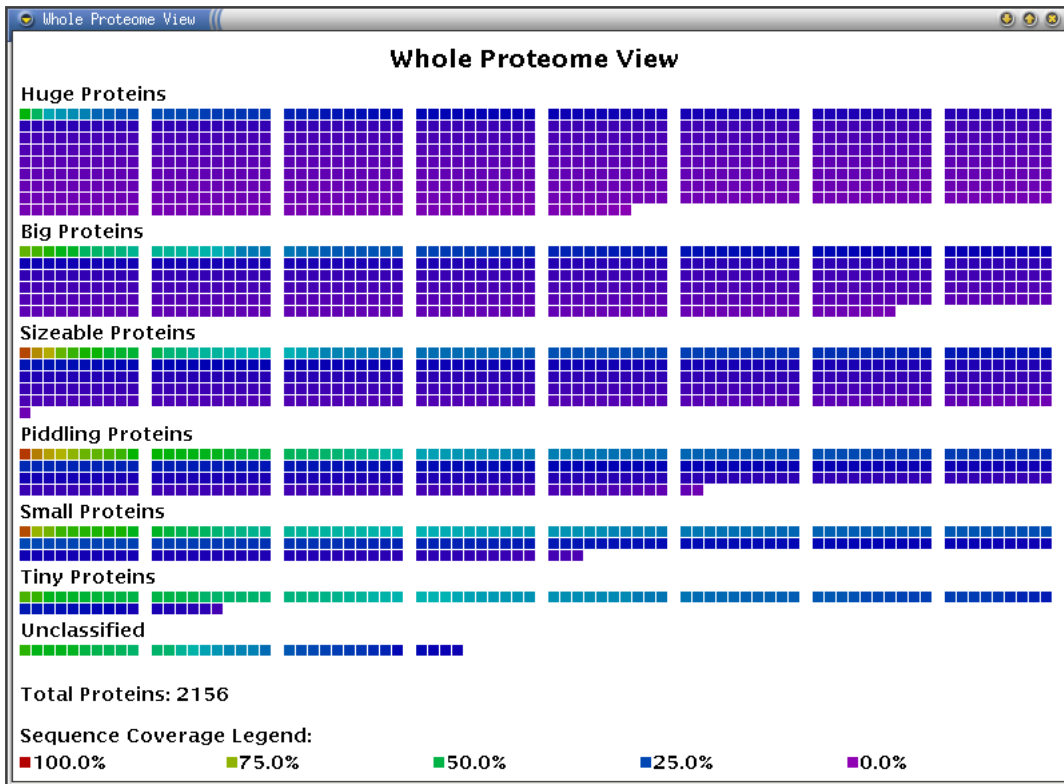


Figure 4: Sample BirdsEye Graphic for Yeast Proteome Sample. This example uses classifications to separate the proteins of the yeast_orfs database into six classes on the basis of mass, leaving a few of the smallest proteins unclassified.

```

# Set title for Bird's Eye Viewer below
Title                               Whole Proteome View
# Should the title appear?
Show-Title                           true
# Set size of dot for each protein
Dot-Size                              8
# Should we show a legend?
Legend                               true
# Should we show an overall count?
Overall-Stats                         true
# Should we show a count for each class?
Class-Stats                          false
# What font size should the title be?
Title-Font-Size                      18
# What font size should the other text be?
Other-Font-Size                      14
# Should we use a white background rather than black?
Reverse-Colors                       true

```

Figure 5: Sample BirdsEye.ini file. Modifications to these values will be reflected the next time DTASelect is run with the --BE option.

int	Rank		
float	Score		
YHR174W	3		4.5623
YLR184W	2		7.5342
...			

Figure 6: Sample AuxInfo.txt file. Two extra fields are added for each protein in this list. The first is an integer, and the heading used for the column will be “Rank.” The second is a floating point number, and its heading will be “Score.” Two proteins are listed, each with its associated values in the order they’re listed in the header of the file. Additional proteins could added on subsequent lines.

3.4 Output Interpretation

The resulting list of proteins and supporting peptides is stored in DTASelect.html and DTASelect-filter.txt. The file headers include the version of DTASelect used, the directory in which it was run, and a table showing the criteria in place for this analysis.

An example of the DTASelect.html file's body can be found in Figure 7. Multiple loci are grouped together if each is identified by the same set of peptides. Loci are shown in order of decreasing sequence coverage and are grouped by classification if `--class` has been used. Each locus name is printed in red and is followed by the number of spectra listed, the number of spectrum copies present for this protein, the sequence coverage (specifically, what percentage of the residues are represented in observed peptides), length of the database sequence in residues, molecular weight, approximate pI, and descriptive name.

The peptide identifications are shown below the protein to which they belong. Each .out file's information includes whether or not the peptide sequence is unique to this locus (indicated by an asterisk at the start of the line), the file name, the XCorr and DeltCN scores, the observed M+H⁺ mass, the rank by preliminary scoring, the percentage of ions expected that were found, the number of times this peptide was identified, and the sequence of this peptide. Peptides that are found in multiple loci have a series of plus symbols after the sequence. Each symbol links to a different locus within the HTML file that contains that peptide. Peptides that have been identified by the SEQUEST_PHOS algorithm rather than normal SEQUEST are indicated by the letter "P" at the start of the line. The peptides are listed in the order in which they align to the full protein sequence, with ties broken through ordering by observed M+H⁺ mass.

Following the list of peptides, some protein groups will show a list of similarities, linking to other protein groups containing similar peptides. The first number listed after each locus is the number of peptides in this locus matching the other. The second number is the count of peptides in this locus not matching the other. The similarity links appear only when the similarity score (as described below) is zero or greater.

```
U YGL147C  2  5   7.9% 191 21604 9.7      RPL9A
U YNL067W  2  5   7.9% 191 21692 9.7      RPL9B
  yeast.2052.2052.2  4.0341  0.3996  1807.64  1 67.9% 2 R.YVYAHFPINVNIVEK.D +
  yeast.1646.1646.2  2.5635  0.272   1381.86  1 72.7% 3 Y.AHFPIINVNIVEK.D ++
Similarities:
YLL024C(1:1)
```

```
U YGR034W  2  2  15.5% 129 14639 10.3     RPL26B
* yeast.0695.0695.2  2.865   0.2723  1257.73  1 75.0% 1 R.KAYFTAPSSER.R
  yeast.0342.0342.2  3.5671  0.2422  1101.44  1 93.8% 1 R.RDDEVLVVR.G +
```

Figure 7: Sample DTASelect.html Fragment: Locus lines supply the current assigned manual validation status ("U" means no status assigned), the locus name, the number of peptides listed below, the total number of spectrum copies representing this locus, the percentage of sequence coverage, the length of the full protein sequence, the average molecular weight of the protein, the pI, and the descriptive name of the locus from the database.

The end of the file is a table of several counts. The unfiltered row describes the data set before filtering comes into play, and the redundant and nonredundant rows describe the data appearing in the DTASelect report.

The “Unfiltered” row’s three counts are calculated directly from the DTASelect.txt file, with no filtering used at all. “Proteins” gives the count of proteins that have at least one peptide match present. “Peptide IDs” provides the number of SEQUEST result files included in the analysis. Individual .out files may contain more than one sequence with a 0.0 DeltCN, and identified sequences may match to more than one peptide. “Copies” indicates how many identifications (count of SEQUEST result files times number of top-ranked sequences per file times number of proteins including each peptide listed) are present before filtering.

The nonredundant count of proteins is the number of protein groups in the DTASelect output, while the redundant count is the number of individual proteins listed, ignoring grouping. The meaning of redundancy changes for the peptide and spectra counts; for these counts, individual sequences which were found in several different proteins are counted only once for nonredundant values but multiply for redundant counts. If the `-t 1` or `-t 2` options have been used (or the default setting used), an individual spectrum may represent a group of several identical spectra. As a result, the number of spectra that passed the criteria will likely exceed the number of peptides reported. Note that both peptide and spectrum counts are bogus when `-t 0` is specified (the redundancy of each peptide is still calculated correctly, but the redundancy is not removed, a necessity for these counts to be meaningful). For both peptide and spectrum counts, the nonredundant number counts each identification only once while the redundant number counts each identification once for every protein group in which it occurs.

In some cases, the ratio of the counts provide more information than they do individually. For example, the ratio of redundant to nonredundant proteins gives some indication of the extent to which the sequence database repeats itself. If one wants to determine the number of spectra each unique reported peptide represents, one can divide the nonredundant spectra count by the nonredundant peptide count. Although understanding the source of each count can be difficult, the information they provide can be useful.

Following the block of counts is a list of spectra which scored well in two different charge states. For example, a spectrum that contains *y* ions with minimal *b* ions may be matched to two overlapping peptide sequences when considered first as a doubly charged and then as a triply charged spectrum. This list indicates matches that should be evaluated with a skeptical eye; a spectrum should be only one charge state, and scoring well at two states almost certainly means that one of the identifications is incorrect.

For larger sequence databases, there will often be many proteins present which bear great similarity to each other. DTASelect attempts to find pairs of proteins which are similar to each other for each sample, and it reports this information in DTASelect-pst.txt when the `--similar` option is used. This file includes a row and column for each protein retained in the DTASelect results. The numbers appearing in this table represent the number of matching peptides for each protein pair minus the number of peptides that don’t match. If, for example, protein A has peptides 1, 2, and 3 while protein B has peptides 1, 2, and 4, there are four peptides matching (two copies each of 1 and 2) and two peptides not matching (3 and 4). The similarity score for this pair is $4 - 2 = 2$. Each protein is listed alongside its similarity to itself, which is always equal to two times its number of peptides.

For users who want to examine how their chromatography is affecting their results, DTASe-

lect provides information about the distribution of peptides among the different chromatographic steps that have been used on each sample and also profiles each step to show when the peptides passing the filters eluted. This information, found in the DTASelect-chroma.txt file, is presented in tab-delimited text when the `--chroma` option is used. The first column lists the “DAT” files. These files are the ones from which the .dta files (spectra) are extracted and usually represent a single dimension of chromatography. The names are inferred from the common prefixes of the SEQUEST result files. The second column shows the total number of peptides after filtering from each DAT file. The third, fourth, and fifth columns show the number of +1, +2, and +3 peptides after filtering, respectively. The remaining columns in each line group all the peptides from each DAT into sets of 100 scans; a peptide numbered 082500dtabbB05.1625.1625.2.out would be in the seventeenth group ($1625/100 = 16.25$, which rounds up to 17). A graph of the numbers after the fifth column for each row, then, will be something like a total ion current, taking into account only the peptides that remain after DTASelect filtering. The information in DTASelect-chroma.txt is helpful in determining the effectiveness of the chromatography employed for a run.

Research focusing on post-translation modifications can benefit from DTASelect’s modification report. This report, generated when the `--mods` option is used, yields information about proteins with at least one modified peptide present (see Figure 8). Available in web (DTASelect-mods.html) and text (DTASelect-mods.txt) formats, the report enumerates the residues in each protein for which at least one dynamic modification was observed. When multiple peptides are observed for a particular modification, a sequence alignment of the identified sequences is produced with a marker to show the modification’s position.

The alignment report can be helpful in showing which regions of a protein’s sequence are most amenable to mass spectrometry. The report, generated when the `--align` option is in effect, creates a table for each protein. Each row in the tables represents a region of contiguous sequence coverage and shows how the peptides in these regions align against each other. Modified residues are indicated by being colored red (or are shown in lowercase in the text report). The starting and ending sequence positions of each contiguous region are included in the first cell of each row. After each protein table, four statistics are listed. The “Maximum Depth” field shows the maximum number of peptides listed in any contiguous region for this protein (not necessarily the largest number of peptides containing a particular residue of the protein). The “Peptide Residues Observed” field sums together the length of each peptide observed for the protein and can be used to determine the multiplicity of sequence coverage for each protein. The “Sequence Coverage” and “Sequence Length” show the same numbers as in the locus lines of DTASelect.html.

Users who want to copy the peptides included in their DTASelect reports to another directory can do so with the `--copy` option. If SQT and MS2 files are found in the current directory, DTASelect will create a new subdirectory called “Subsets” and create new SQT and MS2 files in it. Only the identifications and spectra corresponding to the peptides shown in DTASelect.html will be retained in the copied SQT and MS2 files. If .dta and .out files have been used, DTASelect will create a script entitled “copylist.bash” on UNIX systems or “copylist.bat” on Windows systems.

3.5 DTASelect.txt Databases

The DTASelect.txt file is created the first time DTASelect is run in a directory. This file is useful beyond its capacity to speed up subsequent runs of the program. In essence, the file embodies the primary features of every .out file in the subdirectories. In addition, the file includes the descriptive

gi—117385—sp—P02511—CRAB_HUMAN ALPHA CRYSTALLIN B CHAIN					
19	RRPFFPFHSPS	2.8212	07160134Sela-18.1284.1284.2	1	1
80.0	RPFFPFHSPSR	3.2679	07160134Sela-18.1217.1217.3	1	1
	HSPSRLFDQFF	4.738	06130134Ssub-14.1615.1615.2	1	4
	HSPSRLFDQFFGE	5.301	06130134Ssub-14.1462.1462.2	1	47
	*				
21	HSPSRLFDQFF	2.7263	06130134Ssub-16.1923.1923.2	1	2
80.0	HSPSRLFDQFFGE	4.4624	06130134Ssub-16.1853.1853.2	1	4
	*				
35	RLFDQFFGEHLLSDLFP	2.599	07160134Sela-14.1842.1842.2	1	1
80.0	*				

Figure 8: Sample DTASelect-mods.html Fragment

name of each locus identified from the database. Perhaps its most useful feature, though, is its organization. Each locus is listed with the .out files that substantiate it. This information is recorded before criteria are used against the .out results; the DTASelect.txt file is comprehensive.

The file is tab-delimited text. Lines describing loci begin with “L,” while those starting with “D” provide the information associated with a particular spectrum’s match results. Most fields in each “D” line are printed for the HTML file and do not require additional explanation. At the end of each of these lines, however, are three fields that are used internally. The first indicates where in this locus’ sequence this peptide matches (0 for the start of the sequence). This value is used for determining sequence coverage. One is a boolean value describing whether or not the peptide is a tryptic cleavage at both ends (or is positioned at a terminus). The other boolean value describes whether or not the peptide is unique to this locus. These values are available for use by any program that reads the DTASelect.txt file.

4 Contrast

Running DTASelect in a directory is the first step to analyzing its results against other directories. Contrast draws on the information stored in the DTASelect.txt files found in each of the specified directories to assemble a master protein list and to compare the different runs. Contrast can be run in an arbitrary directory; its configuration file specifies the directories to be included in its analysis.

4.1 Execution

Using Contrast requires a bit more preliminary setup than does DTASelect. The user creates a file called Contrast.params in a directory where results are to be stored. This file specifies the directories to be included in the analysis and the sets of criteria to be used. The Contrast script is then run.

The example Contrast.params in Figure 9 compares the results of two samples under a common set of criteria that has been modified to increase the minimum necessary DeltCN. One use of this technique would determine how the protein complement of an experimental sample differs from a

control.

```
[Included Directories]
  Yeast      /data/dtabb/Yeast
  Yeast2     /data/dtabb/Yeast2
[Criteria Sets]
  NewDCN     -d 0.1
[Options]
```

Figure 9: Sample Contrast.params for Differential Analysis

The first directory is given the alias “Yeast” and the second is aliased as “Yeast2.” The criteria set is nicknamed “NewDCN.” These abbreviated titles are used to name each combination of sample and criteria (hereafter described as a “data set”). In this case, two data sets would be created, namely “Yeast-NewDCN” and “Yeast2-NewDCN.” Note that the criteria modifications are listed just the same as they would appear in DTASelect’s command line.

The Contrast.params found in Figure 10 is a more complex demonstration. Two samples will be compared, as before, but this time, two different sets of criteria are used. Each combination of sample and criteria is evaluated separately, so four different columns will be appear in the final output.

```
[Included Directories]
  Yeast      /data/dtabb/Yeast
  Yeast2     /data/dtabb/Yeast2
[Criteria Sets]
  Loose      -p 1 -d 0.0
  Strict
[Options]
```

Figure 10: Sample Contrast.params for Complex Analysis

“Strict” has no modifications listed and will be set to the default criteria. In this case, the Yeast-Loose data set is likely to retain more proteins than Yeast-Strict because of the eased restriction on peptides per locus and DeltCN. Despite the fact that these data sets come from the same sample, Contrast evaluates them separately.

The Contrast algorithm is limited to 63 columns; the number of samples multiplied by the number of criteria sets cannot exceed 63. The availability of RAM and the width of the browser windows may be more substantial limits for the program.

In some cases, such as the comparison of different peptide identification algorithms, it may be necessary to apply one set of criteria to one set of results while applying a different one to another set. This can be achieved in a section of Contrast.params titled “[Explicit Mappings].” This section should always follow [Criteria Sets] and precede the [Options] section if one is used. Each line in this section should start with the alias for an included directory. Aliases for criteria sets to be applied to each directory should follow, separated by spaces or tabs. In the above example,

one could specify that the “Loose” criteria should be applied to the “Yeast” sample while the “Strict” criteria should be applied to the Yeast2 sample by including the contents of Figure 11 in Contrast.params.

```
[Explicit Mappings]
Yeast      Loose
Yeast2     Strict
```

Figure 11: Sample of explicit mapping between criteria sets and directories

The final section of Contrast.params, [Options], can contain lines for any of several directives (see Table 7). These keywords are case insensitive.

Option	Description
Class	Incorporate protein classification information in Contrast.html.
Database	Create a new FASTA database containing the selected proteins.
Hide	Do not show proteins found in a particular sample.
Master	Include only proteins found in this data set.
Merge	Create a DTASelect.txt file integrating these samples.
Verbose	Show patterns of presence and absence for peptides as well as proteins.

Table 7: Valid Options for Contrast.params

The “class” option is similar to DTASelect’s “--class” option. Classifications.txt is read, and the master protein list assembled in the current Contrast is classified according to the file’s contents. The proteins will still be grouped according to the data sets in which they are found, but within those groups they will be sorted by the assigned classifications.

The “database” option creates a new sequence database (Contrast.fasta) containing only the proteins listed in the Contrast output. The program will search for the listed proteins in each of the databases corresponding to the samples included in the comparison. Such a reduced database can dramatically reduce the time required for subsequent SEQUEST searches of the spectra with additional post-translational modifications.

The “hide” option helps reduce the size of Contrast output. If only the proteins present in the “Experiment” sample but absent in the “Control” sample are of interest, include “hide Control” in the [Options] section, and the proteins present in any combination of the Control sample with any criteria set will be hidden. Only protein counts are displayed for hidden groups. If multiple samples are all named “Control,” they can be removed by a single “hide” statement. If multiple samples or criteria sets are to be hidden, multiple “hide” options, one on each line, should be employed.

“Master” specifies a particular data set to be used as the definitive protein list for the Contrast run. “Master” should be followed by a directory alias and then a criteria set alias (such as “Master Yeast2 Strict,” from Figure 10). This specifies that only the proteins found in Yeast2-Strict will be included from Yeast-Loose, Yeast-Strict, and Yeast2-Loose. This option would be of use if one were investigating lower-scoring peptides in evidence for a set of confidently identified proteins.

The “merge” option will combine together all the listed DTASelect.txt files into a unified DTASelect.txt created in the directory where Contrast is run. DTASelect or Contrast can subsequently be run against this newly created file, although the links to spectra and SEQUEST output files will be made on the assumption that the relative paths to spectra are unchanged from the original DTASelect.txt files.

Contrast’s “verbose” mode details the presence and absence of individual peptides for each protein listed across all data sets, showing the highest XCorr for each peptide in each data set. Verbose mode output can be substantial in length with even moderately complex samples.

4.2 Output Interpretation

Contrast’s output is a collection of files. The primary ones are Contrast.html and Contrast.txt. These files consist of a header showing the included directories as well as the criteria sets for the comparison followed by a series of tables, one for each combination of presence and absence across the data sets. Each table has a column for the locus name, one for each data set, one for cumulative sequence coverage, and one for the descriptive name for each locus. The other reports are the html and text files from DTASelect for each combination of directory and criteria set.

For each row in a Contrast table, the cell for each data set may either be blank (if the locus is not found there) or may hold a percentage indicating the sequence coverage from peptides passing that set of criteria in that sample. The percentages are hypertextual links to the correct locus in an HTML file produced by DTASelect for that data set. The “total” column holds cumulative sequence coverage for all peptides found in all data sets for this protein and links to a view of the protein’s sequence coverage. The partial results in Figure 12 correspond to the Contrast.params in Figure 9 and show a group of proteins present in Yeast but absent in Yeast2. Proteins within a single table are sorted by classification (if used) and then by locus name.

Each row normally represents one protein; however, if multiple proteins have identical sequence coverage, they are grouped together as in DTASelect’s output. The counts at the end of each table count such groups as a single protein. The values representing sequence coverage in individual data sets will be those for the first protein in the group, colored red. Each protein’s cumulative sequence coverage, however, is shown in the “total” column.

At the end of each subtable is a count of the loci in that group. Near the end of the Contrast.html file is a short table providing counts for each combination of presence and absence and a total of loci presented. The example results in Figure 13 correspond to the example Contrast.params in Figure 10 and show the summary table for two samples analyzed under two different conditions.

The last item in Contrast’s report is a two-way comparison table. Each row and column represents a data set. The protein lists for each pair of columns in the analysis are compared, and the percentage of proteins found in the row’s data set which are also found in the column’s data set is shown in each cell. If two samples (A and B) are being compared under the same criteria, for example, there will be two rows and two columns of percentages in the table. The cells for row A, column A and row B, column B each show 100%; any data set’s list of proteins is identical to itself. The cell for row A, column B will show the percentage of proteins present in A that are also found in B. The cell for row B, column A will show the percentage of proteins present in B that are also found in A. If one sample has many more proteins than another, it may be the case that it represents a large percentage of other samples’ proteins. Each cell in this table is color-coded; high percentages map to red or orange, and low percentages map to yellow or green.

Locus	Yeast NewDCN	Yeast2 NewDCN	Total	Description
...				
YBR181C	11		11	RPS6B
YBL090C			11	RPS6A
YCL037C	11		11	SRO9
YCL043C	17		17	PDI1
YDL055C	8		8	PSA1
YDL081C	28		28	RPP1A
YDR099W	15		15	BMH2
YER177W			15	RPL23B
YDR382W	49		49	RPP2B
YDR450W	16		16	RPS18A
YML026C			16	RPS18B
YEL026W	31		31	
...				
28	X		X	

Figure 12: Sample Contrast.html for Differential Analysis

R Count	NR Count	Percent	Yeast1 Loose	Yeast2 Loose	Yeast1 Strict	Yeast2 Strict
117	93	8.6%	X	X	X	X
56	50	4.6%	X	X		X
23	20	1.9%	X	X	X	
98	87	8.1%		X		X
17	15	1.4%	X		X	
91	86	8.0%	X	X		
511	494	45.8%		X		
242	234	21.7%	X			
1155	1079		498	830	128	230

Figure 13: Sample Contrast.html Summary

The most straightforward use of Contrast is simple differential analysis. If a researcher has provided two data sets and one set of criteria, as in Figure 9, the resulting Contrast.html will have two data set columns. One will represent the first sample, and the other will represent the second. Both columns are compared using the same set of criteria, so any particular sample appears in just one column. If researchers were trying to determine which proteins appeared in sample one but not in sample two, they could follow a link to the table grouping together those proteins, showing a percentage in the first column but nothing in the second. Conversely, if they wanted to determine how many proteins were in the second sample but not in the first, they could jump to that group, showing a blank in the first column but a percentage in the second. Proteins that appeared in both samples would be sorted together in the group at the top of the page. If they were only interested in a count of how many proteins fell in the first but not the second, they could follow the link to the bottom and read the numbers from the summary table rather than hunt through the rest of the document.

It is more complex to interpret a multi-sample, multi-criteria analysis. In Figure 10, two different samples and two sets of criteria are present. As a result, four columns appear in Contrast.html in addition to the three columns showing counts and percentages for each group. One can quickly learn that 93 different proteins appeared in both samples under both sets of conditions (the count rises to 117 if one counts indistinguishable proteins separately). The 93 proteins comprise 8.6% of the total, 1079 proteins. To learn how many loci appeared in the Yeast2 sample but not in Yeast, however, would require adding together the results where Yeast2 was positive in both columns and Yeast was negative in both columns. When one increases the complexity of Contrast's output, it becomes more difficult to answer simple questions from its results.

Keeping the Contrast.params file as simple as possible will pay off in simplicity of understanding the result. A complex analysis can often be split into several simple analyses, and it may be beneficial to do so.

5 Summary

DTASelect and Contrast are powerful tools for the sifting meaningful results from complex SEQUEST results. By providing a framework for the uniform, automated application of identification acceptance criteria, DTASelect greatly diminishes the work required to identify worthwhile sequence identifications from the large numbers of spectra typically produced by liquid chromatography / tandem mass spectrometry. Contrast provides meta-analytical tools for the comparison of multiple samples to each other and for the examination of samples under multiple sets of criteria. Just as SEQUEST improved the throughput of peptide spectrum identification, DTASelect and Contrast improve the throughput of SEQUEST data analysis.

A Mascot Support in DTASelect

DTASelect was originally created to analyze SEQUEST output. As a result, the pieces of information stored for each peptide and filters designed to screen them are characteristic of SEQUEST. To read Mascot results, DTASelect must translate from these values to SEQUEST equivalents. The following fields are affected by these differences:

- XCorr: Mascot's Ions Scores are reported in the place of SEQUEST's XCorr. Ions scores have a much broader range than XCorr, ranging upwards of 60 and down to around zero.
- DeltCN: Mascot's Ions Scores should be evaluated in relation to the number of peptides present in the sequence database near the observed precursor mass (Qmatch score). DTASelect looks up the Qmatch and Ions Scores and reports DeltCN as:

$$D = \frac{10^{I/10}}{Q}$$

where D is DeltCN, I is Ions Score, and Q is Qmatch. The resulting number indicates the chance that each peptide is a random match. A DeltCN of 1.0 corresponds to a 1/20 chance of random match. A DeltCN of 5.0 implies a 1/100 (1/20 divided by 5.0) chance of random match.

- M+H⁺: Mascot reports intact peptide masses, while SEQUEST reports peptides masses with a single added proton (M+H⁺ masses). The M+H⁺ masses of peptides are calculated from Mascot results to match their SEQUEST equivalents.
- Rank by Sp: Mascot does not use preliminary scoring to create a candidate list. As a result, DTASelect reports all Mascot-identified peptides with a preliminary score of 1.
- Filename: Currently, DTASelect will only detect filenames correctly if Mascot has been used on .dta files.
- Total Intensity: DTASelect records a zero for the total intensity of each Mascot identification. This value could be calculated from the intensities for each query in the .dat file, but DTASelect does not yet implement this.
- Ion Proportion: DTASelect reports the following for this measure:

$$\frac{M}{(L - 1) * X}$$

where M is the reported Ions Matched score, L is the length of the peptide sequence, and X is 2 for singly-charged and doubly-charged peptides or 4 for triply-charged peptides.

Rather than use the default settings for DTASelect, Mascot users should use a command line like `--Mascot -d 1.0 -1 0.0 -2 0.0 -3 0.0`. This should filter out peptides with worse than 1/20 chances of random matching, regardless of charge state.

Mascot files are organized into folders based on the date when they are created. To select a group of Mascot .dat files which should be grouped for an individual DTASelect, make use of the DTASelect.params file. Create the file in the directory where the DTASelect is to be run. The first line of DTASelect.params must give the default parameters to be used for DTASelect in this directory (see the above paragraph for recommended defaults). The next line should read `[DAT List]` Subsequent lines should contain only the path and filename for each .dat to be included. Comment lines should start with “#” characters.

DTASelect must be able to find the Mascot.dat file to determine the path of the sequence database used for searching. The DTASelect.ini file should contain a line labeled “mascot-path”

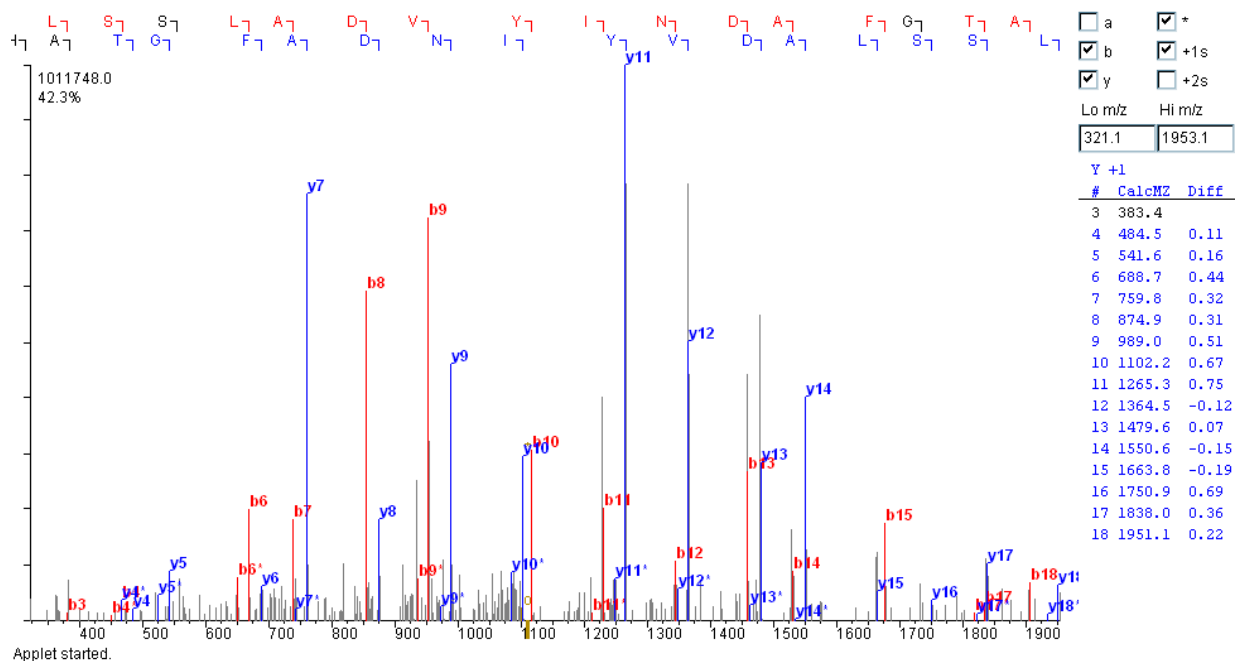


Figure 14: SpectrumApplet Sample. The sequence shown is the doubly charged HELSSLAD-VYINDAFGTAHR, with a precursor $M+H^+$ of 2217.4026.

with the path to the Mascot installation directory (which should include a subdirectory entitled “config” in which Mascot.dat can be found). The first Mascot .dat file supplies the database name.

DTASelect will expect that the individual spectra (.dta files) will be stored in directories bearing the names of the corresponding .dat files. For example, the spectra corresponding to F714280.dat should be found in a subdirectory entitled F714280.

Mascot support in DTASelect was implemented relatively late in development. As a result, features associated with Mascot can be expected to change more rapidly than those relating to SEQUEST. Additionally, Mascot support may be less robust. Your assistance in identifying areas where improvement is needed would be welcome.

B SpectrumApplet Interface

DTASelect includes an applet which can be used to view spectra. The SpectrumApplet displays spectra actively, allowing zooming on particular M/Z ranges and selective highlighting of fragment ion series. The applet is accessible to any CGI which produces an appropriately formatted HTML file. For an example of its use, see Figures 2 and 14.

The CGI-produced HTML file referencing SpectrumApplet should specify its codebase as the DTASelect installation directory. The spectral information is passed to the applet by the following required tags:

- PreMPlusH: The precursor $M+H^+$ (float)
- PreZ: The precursor charge state (integer)

- MatchSeq: The sequence corresponding to this spectrum (String)
- MZ x : The M/Z value for this peak, where x starts at 1 and increments by one for each peak. (float)
- Int x : The intensity value for this peak, where x starts at 1 and increments by one for each peak. (float)

Many HTML tags are available as options in SpectrumApplet calls. If nonstandard masses have been used for amino acids, these options pass the new mass to the applet:

- SMM x : The mass to substitute for a particular amino acid residue, where x starts at 1 and increases by 1 for each additional static modification (float)
- SMR x : The residue which has the substituted mass above (char)

If the search has specified post-translational modifications and these are included in the passed sequence, SpectrumApplet can be notified of these symbols' meanings by the following tags:

- DMM x : The mass accounted for by a particular dynamic modification, where x starts at 1 and increases by 1 for each additional static modification (float)
- DMS x : The symbol in the sequence corresponding to this this dynamic modification (char)

In some cases, peptide identification may be made with a mass modification for either the C- or N-termini of the protein or peptide. These options can be passed to SpectrumApplet by the following options:

- CPepMod: The mass modification on the C-terminus of this peptide (float)
- NPepMod: The mass modification on the N-terminus of this peptide (float)
- CProtMod: The mass modification on the C-terminus of this protein sequence (float)
- NProtMod: The mass modification on the N-terminus of this protein sequence (float)

If the identification was achieved by use of monoisotopic masses rather than average masses, the SpectrumApplet needs that information:

- AvgForFrag: Should monoisotopic masses be used rather than average masses for calculating fragment ion positions? (boolean)
- AvgForParent: Should monoisotopic masses be used rather than average masses for calculating the precursor mass? (boolean)

If the CGI needs to modify the display of the spectrm, these three tags are available:

- LoMZ: The minimum m/z shown (float)
- HiMZ: The maximum m/z shown (float)

```

<applet code="SpectrumApplet.class"
CODEBASE="http://localhost/DTASelect1.8/"
width=970 height=500>
<param NAME="PreMPlusH" VALUE="1159.72">
<param NAME="PreZ" VALUE="1">
<param NAME="MZ1" VALUE="347.1">
<param NAME="Int1" VALUE="417922.0">
<param NAME="MZ2" VALUE="348.3">
<param NAME="Int2" VALUE="101197.0">
<param NAME="MZ3" VALUE="349.7">
<param NAME="Int3" VALUE="90863.0">
[Remaining peak parameters removed]
<param NAME="MatchSeq" VALUE="R.IGSEVYHNLK.S">
</applet>

```

Figure 15: Basic HTML tags for SpectrumApplet

- ShowPrecursorLosses: Should losses from the precursor be shown? (float)

If other software is being used to recognize features of the spectrum, peaks may be linked together by lines colored as the ion series are. For example, a sequence tag algorithm may predict a “ladder” of intensities and m/z locations of several peaks. The algorithm output may be displayed against the complete peptide identification.

- LadderLength x : The length of each ladder being described, where x starts at 1 and increments by one for each ladder.
- LadderSeries x : The series of each ladder being described. Series should be 0 for +1 b ions, 1 for +1 y ions, 2 for +2 b , 3 for +2 y ions, 4 for +1 a ions, 5 for +2 a ions, 6 for precursor-related ions, and 7 for displayed ions to appear in black.
- LadInt y : The intensity of a peak in a ladder. y should begin at 1 and increase by one for each point. If two ladders of three peaks are shown, the first ladder is points 1-3 and the second ladder is points 4-6.
- LadMZ y : The m/z value for a peak in a ladder.

The parameters can be passed in any order. The sequences should be passed as they appear in SEQUEST results: R.IALSRPNAYM*FK.Y (with flanking sequence characters) is required rather than IALSRPNAYM*FK. Figure 15 shows an example of a valid call to SpectrumApplet.

In this example, The precursor’s M+H⁺ is 1159.72, and it was observed singly charged. One static modification was present, with cysteines being expected at 160.139 instead of 103 daltons. Any occurrence of “*” in the sequence corresponds to a residue modification of 16 daltons (an oxidation). Since neither a C nor a “*” appears in the sequence, these parameters could have been omitted.

The sequence fragment ions highlighted in the spectrum are the most intense falling within 0.75 M/Z of a calculated fragment ion position (the DaughterDB software manual gives a more detailed description of how the M/Z positions of fragment ions are calculated). The precursor ion, if any remains, is labeled with a brown tick mark below the x axis, and if ions appear at the expected positions for ammonia, water, acetyl, or phosphate neutral loss from the precursor, they are colored brown and marked with *, ^o, AcT, or PhS labels, respectively. The colors used for labeling sequence ions are given in Table 8. If the fragment ions lose ammonia, the resulting peaks are labeled with the same color as the source series with a “*” to indicate that they are 17 Da away from the fragment ion.

Series	Z = +1	Z = +2
<i>a</i>	Green	Light green
<i>b</i>	Red	Light Purple
<i>y</i>	Navy Blue	Light Blue

Table 8: Sequence ion color coding in SpectrumApplet

The series of letters at the top of the spectrum show how the peptide sequence aligns to the spectrum. The colors of the letters show which ion series is being described (see Table 8). If the peptide is triply charged, four sequence rows will appear rather than the two appearing for other spectra. If a sequence letter is black, no ion in the spectrum corresponded to the peak to this amino acid residue’s side. The letters in the *y* ion row(s) are in reverse order relative to those in the *b* ion row(s) because the N-terminal and C-terminal ions appear in opposite orders by M/Z .

SpectrumApplet presents the intensity of the base peak (the tallest in the spectrum) at the upper left of its display. The program computes the proportion of the spectrum’s intensity which is accounted for by fragment ions and reports it below the base peak intensity.

SpectrumApplet includes a control panel that does not appear in the DTASelect’s GUI. The options allow the labelling of a specific ion series rather than all simultaneously. In addition, users can specify high and low M/Z cutoffs; to view only the region of the spectrum after 500 M/Z , type “500” into the “Lo m/z” box and hit enter. The y axis does not change when the x axis is zoomed.

Below the control panel, users can check the correspondence between observed and predicted fragment ion M/Z s for a particular ion series. Ions that do not match to the spectrum are in black. The middle column of numbers shows the M/Z at which each fragment ion was expected to appear, and the column at the right shows the discrepancy between the expected and observed positions. The default list shown is for +1 *y* fragment ions. If series are switched off by the checkboxes, the program will instead show +1 *b*, +2 *y*, or +2 *b* ions.