Unique Peptide Finder (UPF) – User manual

Version 0.912

Content

1. Introduction

Reference note

It should be expressly mentioned that the UPF software is currently (April 2008) in beta status.

1. a. Preface

Proteomics is a powerful methodology to investigate protein expression in cells, tissues, organs or whole organisms. One fundamental idea of proteomic approaches is the expression investigation of thousands of proteins at the same time. Proteome analysis more and more appears into the spotlight of classical fundamental as well as clinical research. Differential quantitative proteome analysis allows direct comparison of proteomes of different cellular states whereas descriptive qualitative approaches provide an insight in the protein composition of a given cell, organelle, tissue etc. Protein identification usually is done by mass spectrometry (MS). MS can be either performed in the form of whole-protein analysis ("top-down" approach) or by investigation of enzymatically produced peptides ("bottom-up" approach).

In bottom-up proteomics approaches trypsin or other proteases are used to digest the proteins into a set of peptides prior to their mass spectrometric analysis (Kocher and Superti-Furga, 2007). Substantial advantages when working with peptides instead of proteins such as superior solubility of peptides in a wide variety of solvents as well as their lower nonspecific adsorption to surfaces make the bottom-up approach very attractive for comprehensive proteome analysis. On the other hand, after protein digestion the number of molecular species increases dramatically, which complicates the analysis (Lohaus et al., 2007). Unfortunately, all information about a particular intact protein is lost. Therefore, determination of proteins on the

basis of detected peptides implies some uncertainty since a peptide may be part of different proteins.

The detection of unique peptides is especially important, when doing targeted proteomics. In this (special) case the proteins that shall be analyzed by mass spectrometry are already known before the analysis.

As an example, multiple reaction monitoring (MRM) turned out to be a fast and efficient technique, which allows quantifying proteins either relatively or absolutely by means of triggering a set of specific peptides (Janecki et al., 2007; Le Blanc et al., 2003; Mayya et al., 2006; Wolf-Yadlin et al., 2007). Analysis of very similar proteins usually deals with only a small set of unique peptides that needs to be calculated for unambiguous identification of these proteins.

Manually this can for example be done by theoretic digestion of the sequences of interest and blasting of every resulting peptide [www.ncbi.nlm.nih.gov/blast/Blast.cgi]. However, performing BLAST searches manually in order to identify unique peptides is both tedious and time-consuming.

Nowadays sample preparation, nanoHPLC, mass spectrometry and protein identification in modern proteomics can be automated (Alterovitz et al., 2006) , but to our knowledge no software is available for automated calculation of unique peptides for unambiguous identification of proteins. Here we report the construction of a software tool, which is called Unique Peptide Finder (UPF). UPF is used to calculate sets of unique peptides from protein sequences. Furthermore, UPF provides the possibility to calculate both the frequency of peptides and to retrieve information from which proteins a particular peptide can be derived.

The automation of this process requires the theoretic digestion of sequences of interest and a comparison with a pre-digested protein sequence database avoiding

the discussed problems caused by using the BLAST function against a non-digested sequence database.

The software will help on the interpretation of regular proteomics data and the generation of peptide lists for targeted proteomics especially when it comes to a desired differentiation between very similar proteins.

The next section deals with the installation of the UPF software. Because UPF consists of two modules each module is described in its own chapter: chapter two deals with the first module (named command line tool) and the second module (named query tool) was explained in chapter three.

1. b. Installation

Software requirements

Because UPF is a Java™-based software solution (JDK 6, Sun Microsystems, Inc., Santa Clara, CA, United States), users need to install the Java Runtime environment (JRE), which can be obtained via the internet from http://www.java.com/en/. Avoiding problems due to possible incompatibility with former versions of the JRE it is strongly recommended to download the actual Java™ version.

In order to identify unique peptides with respect to a given set of proteins, it is necessary to generate databases, that contain all peptides from a whole set of proteins. Because the UPF software uses the relational database model for storing peptide information obtained from pre-digested protein sequences, users need to have access to an adequate database management system (DBMS). During the development of the software, Microsoft™ SQL Server 2000 (Microsoft Corp., Redmond, WA, United States) was used as database management system. However

for providing the use of UPF free of charge, the open source database software MySQL can be utilized as well. MySQL can be downloaded as MySQL Community server version, which is available free of charge from http://dev.mysql.com/downloads/mysql.

An important objective during the development was to provide easy access to actual and common protein databases. Currently, designated input can be IPI and UniProtKB/Swiss-Prot databases in FASTA file format. Because both IPI and UniProtKB/Swiss-Prot databases can be easily obtained from the internet, no such databases were shipped with UPF.

Actual IPI databases in FASTA format are available from ftp://ftp.ebi.ac.uk/pub/databases/IPI/current/.

The UniProtKB/Swiss-Prot database in FASTA format is available from ftp://ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/. Processing of the NCBInr database will be integrated into UPF in the near future. These input databases can be stored elsewhere.

Installing the UPF software

UPF was developed and tested for the Microsoft Windows™ XP™ operating system. Nevertheless, the software should be executable on Linux operating systems as well. Users, which want to use UPF in combination with the Windows™ Vista™ operating system and the current Windows™ version (Version 5.0) of the MySQL database management system may be referenced to a possible bug that prevent configuration of MySQL. Please refer to the MySQL bug system (http://bugs.mysql.com/ ; Bug ID: 30823) for further information concerning this issue.

The software is supplied as a zip file named UPFv_X.zip where X denotes a version number. Please extract this file in a directory of your choice. The zip file

contains two executable jar files, each of them representing a module of the UPF software, different configuration files, database driver for Microsoft SQL Server 2000 (jtds-1.2.2.jar), MySQL (mysql-connector-java-5.0.8-bin.jar) and this manual.

UPF consists of two modules: The command line tool (UPFcommTool.jar) is used for generating the necessary databases. The query tool (UPFQueryTool.jar) performs database queries and the presentation of the results. It is possible to separate the modules in different directories or even different computers. Nevertheless, each module is delivered with its own configuration file: the command line tool stores its properties in the file named digestDBProp.txt, the query tool uses the QueryToolProp.txt for its settings. These files are used in extensive in order to ensure a maximum of flexibility. It is therefore indispensable to store the configuration file in the same directory as the corresponding module.

It is recommended to download adequate database drivers from the internet, if you use different versions of Microsoft SQL Server 2000 or MySQL DBMS, respectively.

2. Command line tool

2. a. Layout of the databases

For running the command line tool, adequate database tables need to be created by the user. For each considered enzymatic digestion a particular peptide database has to be created. Each of those databases consists of two tables. The first table stores information retrieved from the headers of the protein entries of the FASTA file (e.g. alias accession numbers, taxonomic identification, etc.) Great importance is attached on retrieving a maximum of information. The second table is linked to the first table and stores sequences and masses of the peptides obtained from an in *silico* digestion. Therefore, starting from a specific peptide sequence 'A', which occurs in the second table, it is possible to track any protein from the first table that contains peptide sequence 'A'.

Currently, a specific structure is necessary to meet the requirements of the command line tool, i.e. particular column names and data types (Table 1 and 2) are needed. Because the length of the entries is highly flexible, the specification given in table 1 and 2 should be interpreted as a proposal of wide scope. Nevertheless, users have to ensure sufficient lengths of the database attribute.

Except for the unique identifier (i.e. column/attribute 'ID' in both the protein and the peptide database table) 'null' entries are allowed for each attribute of the database. Since ID serves as a unique identifier for the peptide or protein entry the option *auto_increment* has to be applied to this attribute of the database table. Additionally the option *primary key* can be applied to the ID attribute of both protein and peptide tables.

Table 1: Structure of the protein table generated with the command line tool using FASTA file formatted IPI or UniProtKB/Swiss-Prot databases as program input

Database Structure of IPI Databases – Protein table			
Column name (attribute)	**Description**	**Data type[1]**	**Length**
ID	Unique identifier for each data record of the protein table	bigint	8
ac	Main accession number (here, i.e. the IPI accession number)	varchar	250
gi_ac	Cross references	varchar	250
uniprot_ac		varchar	250
trembl_ac		varchar	250
ensembl_ac		varchar	250
refseq_ac		varchar	250
vega_ac		varchar	250
hinv_ac		varchar	250
genesymbol_ac		varchar	250
taxid	Taxonomic description	varchar	50
protname	Name of the protein	ntext* mediumtext+	16
sequence	Sequence given in the one letter code	ntext* mediumtext+	16
mass	Calculated mass of the protein	float	8
numb_X	Number of X symbols that occur in the sequence	int	4
numb_B	Number of B symbols that occur in the sequence	int	4
numb_Z	Number of Z symbols that occur in the sequence	int	4
flag_WC	Flag indicating, whether a non – standard amino acid is part of the protein sequence	bit	1
Database Structure of Swissprot Databases – Protein table			
Column name	**Description**	**Data type**	**Length**
ID	Unique identifier for each data record of the protein table	bigint	8
ac	Main accession number (here, i.e. the Uniprot / Swissprot accession number)	varchar	250
identifier_short	Shortform, that combines an abbreviation of the protein name and for the taxonomic	varchar	250

[1] Data types used exclusively with Microsoft SQL Server 2000 was labelled with *, data types used used exclusively with MySQL was labelled with +.

	identifier		
protname	Name of the protein	ntext* mediumtext+	16
tax	Taxonomic description	varchar	8000
sequence	Sequence given in the one letter code	ntext* mediumtext+	16
mass	Calculated mass of the protein	float	8
numb_X	Number of X symbols that occur in the sequence	int	4
numb_B	Number of B symbols that occur in the sequence	int	4
numb_Z	Number of Z symbols that occur in the sequence	int	4
flag_WC	Flag indicating, whether a non – standard amino acid is part of the protein sequence	bit	1

Table 2: Structure of the peptide table generated with the command line tool using FASTA file formatted IPI or UniProtKB/Swiss-Prot databases as program input

Database Structure of the peptide table for both IPI and UniProtKB/Swiss-Prot databases			
Column name	**Description**	**Data type**	**Length**
ID	Unique identifier for each data record of the peptide table	bigint	8
sequence	Sequence of the peptide	varchar	8000
massMono	Calculated monomolecular mass of the peptide	float	8
massAV	Calculated average mass of the peptide	Float	8
prot_id	Protein identification number, that is used for identifying the protein from which the peptide was derived	bigint	8
flag_WC	Flag indicating, whether a non – standard amino acid is part of the peptide sequence	bit	1

2. b. Configuring the command line tool

The command line tool has to be configured by editing the file *digestDBProp.txt,* which was delivered with the UPFv_X.zip file.

It is expected that the configuration file was located in the same directory which contains the executable jar file. Usually, an adequate location of the

configuration files should be assured simply by unpacking the zip file. *digestDBProp.txt* is designed in the common Windows™ ini-file style, i.e. the file contains lines of key and values pairs, e.g. the line *'DBMS=mysql'* assigns the value *'mysql'* to the key *'DBMS'*. This special key and value pair defines that MySQL is used as database management system when running the command line tool.

A number of related key and value pairs can be grouped in a section. Sections are marked with headers, where the section name is given in brackets. Each section can be addressed by its section name. The *digestDBProp.txt* stores properties necessary for establishing the database connection as well as specific parameters, which are needed to perform the theoretical digest, e.g. the cleavage sites of a particular peptidase.

The [Database] section is completely shown in the following:

```
[Database]
DBMS=mysql
DBdriverMS_SQLServer=net.sourceforge.jtds.jdbc.Driver
DBdriverMySQL=com.mysql.jdbc.Driver
```

In this section UPF users simply need to edit the value of the *DBMS* key. Whether the MS SQLServer (value *ms_sqlserver*) or the MySQL database (value *mysql*) is used by the commond line tool is governed by the *DBMS* property.

Other sections of interest are sections concerning peptidase related properties. The [Trypsin] section stores all properties, which are need by the command line tool in order to perform a tryptic digestion. This section is completely shown in the following:

```
[Trypsin]
modifyAllPeptideTermina=true
isCompleteModification=false
iAverage=true
```

isEnzymeNTerminus=false
isEnzymeNotPException=false
usePrimeAndSecMods=false
isMultipleModifications=false
isCombinedDigest=false
cleavageSites=KR

Two keys are of particular interest: as the name implies, the key *cleavageSites* stores the cleavage sites specific for trypsin. The key *isEnzymeNotPException* defines if the amino acid in the immediate vicinity of a cleavage site leads to a misscleavage at this position. For this, *false* should be specified as value for this property.

The command line tool supports tryptic digestion per default. However, the configuration file *digestDBProp.txt* can easily be extended for the use of other peptidases. Therefore, the user simply needs to add a new section at the end of the configuration file. Simply copy and paste the complete [Trypsin] section. Then the section name, i.e. the term inside of the brackets, has to be substituted for an appropriate identifier of the 'new' enzyme. Furthermore, the values for the keys *cleavageSites* and *isEnzymeNotPException* have to be specified.

Please avoid editing the other properties of the [Trypsin] section, because these other key and value pairs were applied in order to meet future development of the UPF software. Changing these values may lead to malfunction.

Entries in the [IPI], [NCBI] and [Swissprot] should not be edited by the user. This holds in particular for the regular expressions. Regular expressions provide a succinct and flexible means for identifying patterns of characters and can therefore be used for parsing the header of IPI and UniProtKB/Swiss-Prot databases in order to obtain protein related information. These properties are identified by keys that start with the prefix *regex*.

2. c. Generating databases with the command line tool

The command line tool can be executed from the command line. Start the windows command line interface and change into the directory where the zip.file was extracted. Make sure that the following three files are present: aminoAcids.cfg digestDBProp.txt and UPFcommTool.jar.

The UPF command line tool needs to be started with following program arguments, which are listed in table 3. The program arguments should be separated by blanks when invoking the command line tool.

Table 3: Arguments used for the UPFcommTool for generating pre-digested databases

Number of the argument	*Description or structure of the program argument, respectively*	*Example*
1	IP address/name of the database	//127.0.0.1/uniquepept
2	Username	root
3	Password	rootpassword
4	Name of the enzyme	Trypsin
5	Abslute path of the database	E:\databases\ipi.HUMAN.v3.38.fasta
6	Type of the database	IPI
7	First part of the database table names	ipihuman338tryp

An example for invoking the command line tool is given in the following, where the arguments listed in the column 'Example' of table 3 are used:

java –jar UPFcommTool.jar //127.0.0.1/uniquepept root rootpassword Trypsin E:\databases\ipi.HUMAN.v3.38.fasta IPI ipihuman338tryp

Using these specific arguments the database *uniquepept* is used on the local computer, which is specified by the address of the localhost (*127.0.0.1*). *root* is the user name and *rootpassword* the password which allows access to the database. Parameters given in the [*Trypsin*] section of the configuration file *digestDBProp.txt* are used to perform digestion of the *IPI* database *ipi.HUMAN.v3.38.fasta*. This

specific input database is specified by the absolute pathname *E:\databases\ipi.HUMAN.v3.38.fasta*. The Protein and the Peptide table is denominated by the string *ipihuman338tryp.* This string is extended during run time with the suffixes *prot* and *pept* in order to match the names of the Protein and the Peptide table, which have to be created by the user before (cp. Chapter 2.1 Layout of the databases).

Calls of the UPFcommTool can be added to shell scripts or batch files. Furthermore, scheduling services like *cron* (Linux) or *Task Scheduler* (Windows) can be used for launching the UPFcommTool at pre-defined times or after specified time intervals. This may facilitate maintenance of a couple of pre-digested databases, when synchronisation with up-to-date FASTA formatted input databases is required.

3. Query tool

3. a. Usage of the query tool

The query tool is used for performing both database queries with respect to the input given by the user.

The query tool basically consists of a tool bar, two different areas (A text field and a summary table) and a panel with several buttons used for program control (Fig. 1).

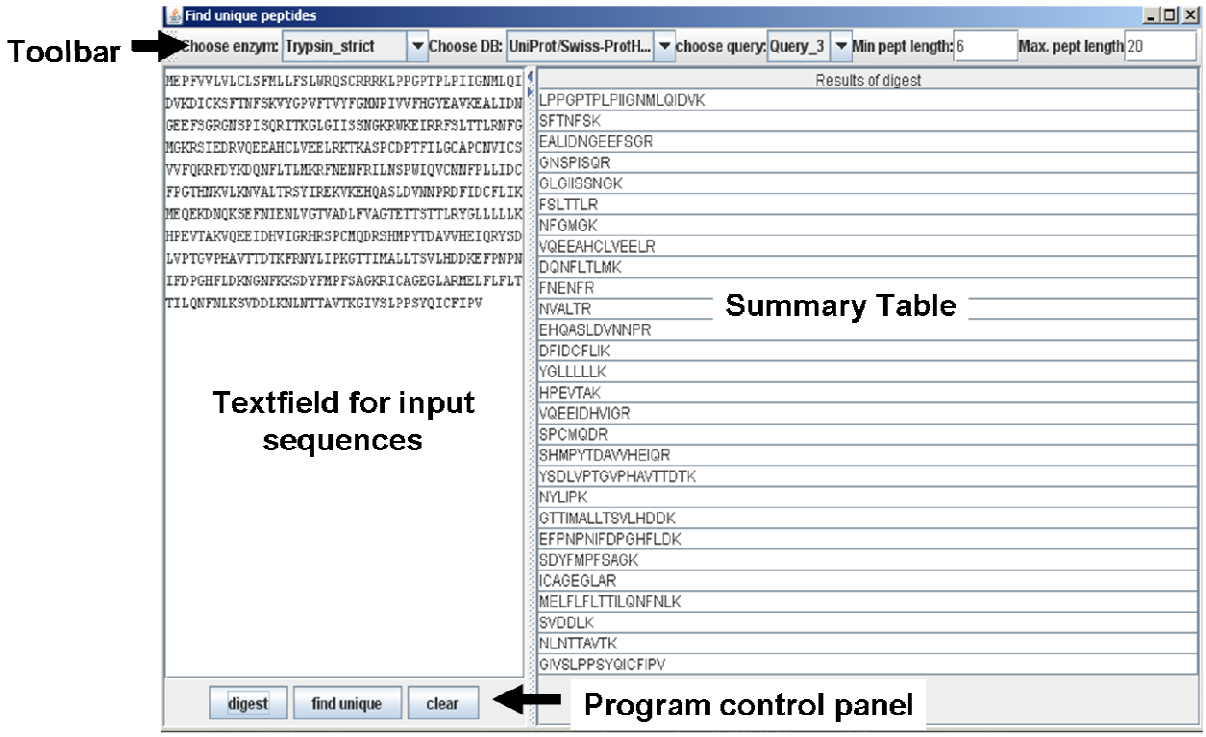


Figure 1: Screenshot of the query tool showing the basic construction of the graphical user interface (GUI). Four different sections can be distinguished: A toolbar is located at the top of the program window. The middle area is separated in a text field on the left hand side and a summary table on the right hand side. The query tool was controlled by the three buttons of the control panels, which was located at the bottom left.
The summary table shows the peptides obtained from the theoretical digest human Cytochrome P450 (CYP) isoform 2C8 sequence. Note that the results are restricted to peptides with length between 6 and 20 amino acids.

Input sequences can be supplied by copy and paste in the text field on the left hand side. Sequence information must be given in the one letter code. **Note that it is**

indispensible to carefully check for eliminating possible *blanks* and *word-wraps* within the sequence. Any blank or word-wrap will lead to mistakes when performing the in *silicio* digestion. Hence, it is recommended to use a editor software like the Windows™ notepad for preparing the sequences prior to copying them into the text field.

It is reasonable to configure the settings in the tool bar before performing a digestion or the database query (Fig. 2).

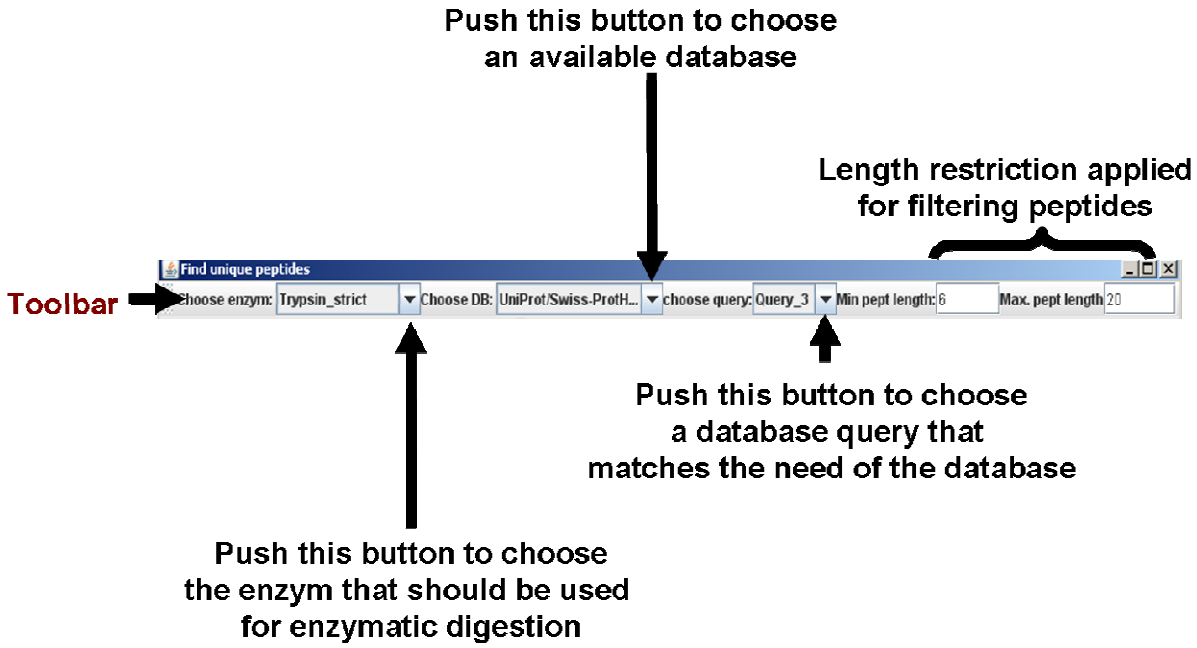


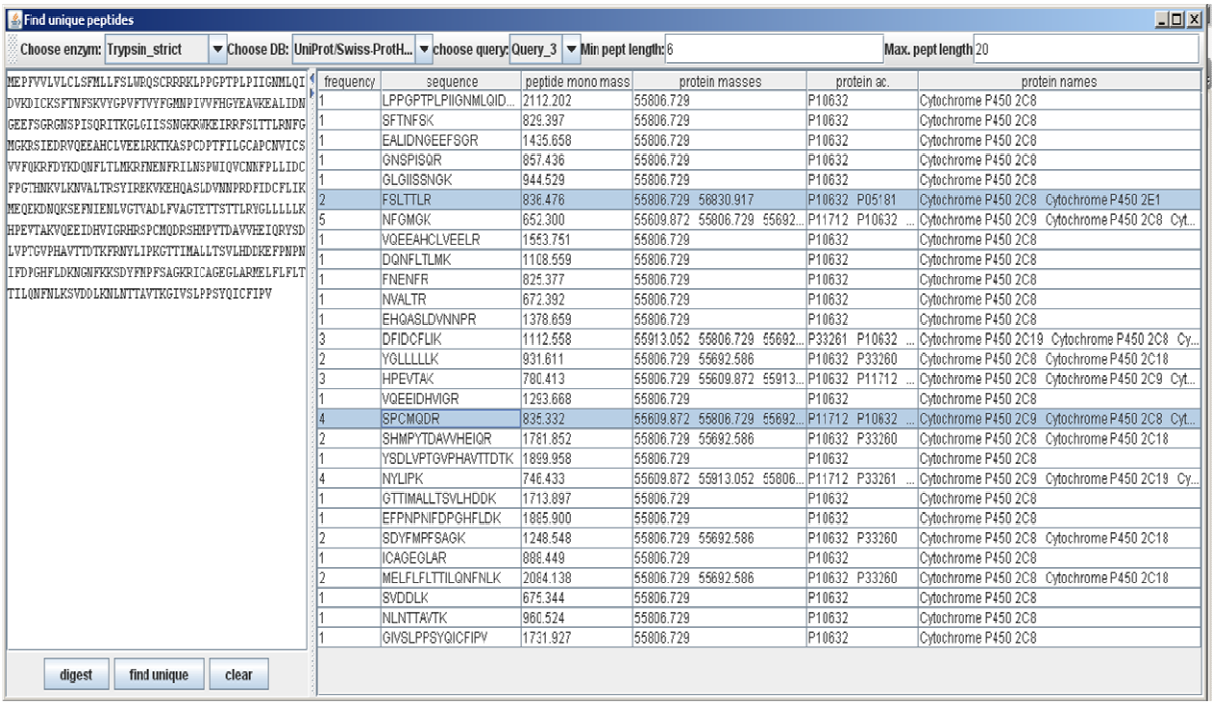
Figure 2: View of the toolbar with different opportunities to configure both enzymatic digestion and database searches.

There are three combo boxes, i.e. lists that can be expanded by pushing a button, in the tool bar. The combo box on the left is used to choose a protease. Similarly, the other combo boxes are used to choose a database (combo box in the middle) and a database query, which is predefined in the configuration file of the query tool in order to match the need of the actual database (combo box on the

right). Additionally, a peptide length restriction can be made to exclude peptides which are very short or very long, because these couldn't be monitored in a regular LC-ESI-MS/MS setup. Moreover, excluding very short peptides significantly reduces the time needed for the database search. Two text fields located rightmost on the tool bar are used to define the maximum and the minimum length of peptides that should be considered.

After pressing the *digest* button in the control panel, the summary table on the right hand side shows all theoretic peptides of this protein specific for the used protease (Fig. 1).

This list is used to compare it with the predigested protein database, which is selected in the tool bar. For that, simply push the *find unique* button. After a while, the summary table shows the results subject to the chosen database query (Fig. 3).



frequency	sequence	peptide mono mass	protein masses	protein ac.	protein names
1	LPPGPTPLPIIGNMLQID...	2112.202	55806.729	P10632	Cytochrome P450 2C8
1	SFTNFSK	829.397	55806.729	P10632	Cytochrome P450 2C8
1	EALIDNGEEFSGR	1435.658	55806.729	P10632	Cytochrome P450 2C8
1	GNSPISQR	857.436	55806.729	P10632	Cytochrome P450 2C8
1	GLGIISSNGK	944.529	55806.729	P10632	Cytochrome P450 2C8
2	FSLTTLR	836.476	55806.729 55830.917	P10632 P05181	Cytochrome P450 2C8 Cytochrome P450 2E1
5	NFGMGK	652.300	55609.872 55806.729 55692...	P11712 P10632 ...	Cytochrome P450 2C9 Cytochrome P450 2C8 Cyt...
1	VQEEAHCLVEELR	1553.751	55806.729	P10632	Cytochrome P450 2C8
1	DQNFLTLMK	1108.559	55806.729	P10632	Cytochrome P450 2C8
1	FNENFR	825.377	55806.729	P10632	Cytochrome P450 2C8
1	NVALTR	672.392	55806.729	P10632	Cytochrome P450 2C8
1	EHQASLDVNNPR	1378.659	55806.729	P10632	Cytochrome P450 2C8
3	DFIDCFLIK	1112.558	55913.052 55806.729 55692...	P33261 P10632 ...	Cytochrome P450 2C19 Cytochrome P450 2C8 Cy...
2	YGLLLLLLK	931.611	55806.729 55692.586	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
3	HPEVTAK	780.413	55806.729 55609.872 55913...	P10632 P11712 ...	Cytochrome P450 2C8 Cytochrome P450 2C9 Cyt...
1	VQEEIDHVIGR	1293.668	55806.729	P10632	Cytochrome P450 2C8
4	SPCMQDR	835.332	55609.872 55806.729 55692...	P11712 P10632 ...	Cytochrome P450 2C9 Cytochrome P450 2C8 Cyt...
2	SHMPYTDAVVHEIQR	1781.852	55806.729 55692.586	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	YSDLVPTGVPHAVTTDTK	1899.958	55806.729	P10632	Cytochrome P450 2C8
4	NYLIPK	746.433	55609.872 55913.052 55806...	P11712 P33261 ...	Cytochrome P450 2C9 Cytochrome P450 2C19 Cy...
1	GTTIMALLTSVLHDDK	1713.897	55806.729	P10632	Cytochrome P450 2C8
1	EFPNPNIFDPGHFLDK	1865.900	55806.729	P10632	Cytochrome P450 2C8
2	SDYFMPFSAGK	1248.548	55806.729 55692.586	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	ICAGEGLAR	886.449	55806.729	P10632	Cytochrome P450 2C8
2	MELFLFLTTILQNFNLK	2084.138	55806.729 55692.586	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	SVDDLK	675.344	55806.729	P10632	Cytochrome P450 2C8
1	NLNTTAVTK	960.524	55806.729	P10632	Cytochrome P450 2C8
1	GIVSLPPSYQICFIPV	1731.927	55806.729	P10632	Cytochrome P450 2C8

Figure 3: View of the query tool after pushing the *find unique* button. The summary table shows information for each peptide obtained from the theoretical digest of the sequence given in the text field on the left (note that the sequences in the sequence column correspond to the sequences listed in figure 1). The frequency (leftmost column) listed is calculated subject to the occurrence of a particular peptide within

the database. frequency are Further columns show (from the left to the right) the sequence of the peptide, the monoisotopic mass calculated for the peptide, all protein masses, protein accession numbers and protein names, in which the peptide can be found. Some cells of the table may contain multiple items. These items are separated by blanks. The highlighted peptide information lines were further discussed in the text (Chapter 3.b.).

Note that no reply is given while executing the database query. This may take a while, especially when very short peptides or even single amino acids are not excluded from the database query. Thus, please be patient.

Results can be marked within the summary table and transferred into style sheet programs via the clipboard.

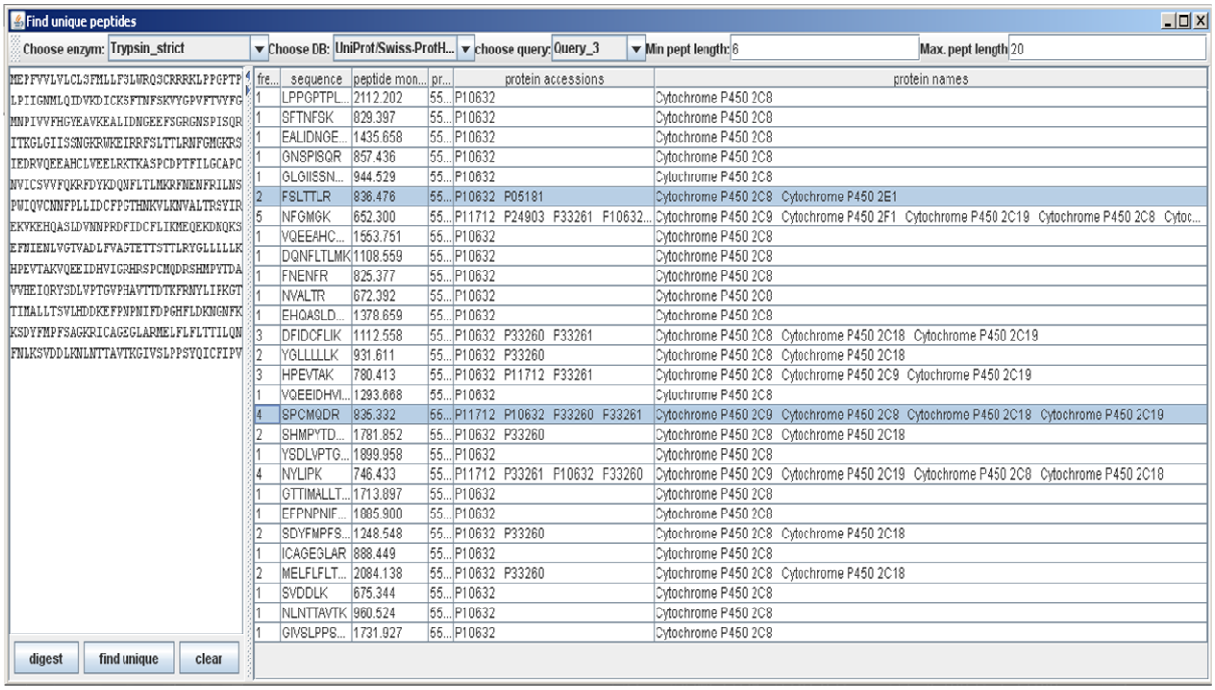
3. b. A brief example for the application of the query tool.

To show the usability of UPF several isoforms from the human Cytochrome P450 (CYP) family have been selected and their sequences have been analyzed.

In human 57 CYP isoforms have been identified so far, which are classified based on their sequence homology. CYPs are responsible for the oxidative metabolism of many xenobiotics as well as organic endogenous compounds. Especially members of the CYP3A* and CYP2C* families are highly homologous; CYP2C9 and CYP2C19 for example show a sequence homology of 91% (calculated by sequence alignment). Because no antibodies are available for the same purpose, mass spectrometry is the method of choice to differentiate between these protein isoforms.

In the following a subset of the UniProtKB/Swiss-Prot database, which considers only entries of human proteins, was used in order to evaluate the software. The sequence of CYP2C8 (left, Fig. 1 to 4) was *in silico* digested with trypsin (toolbar at the top, Fig. 1, 3 and 4). With a peptide length restriction of six to twenty amino

acids, eighteen unique peptides can be found for CYP2C8 (peptides showing a frequency of 1, Fig. 3 and 4).



fre...	sequence	peptide mon...	pr...	protein accessions	protein names
1	LPPGPTPL	2112.202	55...	P10632	Cytochrome P450 2C8
1	SFTNFSK	829.397	55...	P10632	Cytochrome P450 2C8
1	EALIDNGE	1435.658	55...	P10632	Cytochrome P450 2C8
1	GNSPISQR	857.436	55...	P10632	Cytochrome P450 2C8
1	GLGIISSN	944.529	55...	P10632	Cytochrome P450 2C8
2	FSLTTLR	836.476	55...	P10632 P05181	Cytochrome P450 2C8 Cytochrome P450 2E1
5	NFGMGK	652.300	55...	P11712 P24903 P33261 F10632	Cytochrome P450 2C9 Cytochrome P450 2F1 Cytochrome P450 2C19 Cytochrome P450 2C8 Cytoc...
1	VQEEAHC	1563.751	55...	P10632	Cytochrome P450 2C8
1	DQNFLTLMK	1108.559	55...	P10632	Cytochrome P450 2C8
1	FNENFR	825.377	55...	P10632	Cytochrome P450 2C8
1	NVALTR	672.392	55...	P10632	Cytochrome P450 2C8
1	EHQASLD	1378.659	55...	P10632	Cytochrome P450 2C8
3	DFIDCFLIK	1112.558	55...	P10632 P33260 F33261	Cytochrome P450 2C8 Cytochrome P450 2C18 Cytochrome P450 2C19
2	YGLLLLLK	931.611	55...	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
3	HPEVTAK	780.413	55...	P10632 P11712 P33261	Cytochrome P450 2C8 Cytochrome P450 2C9 Cytochrome P450 2C19
1	VQEEIDHVI	1293.668	55...	P10632	Cytochrome P450 2C8
4	SPCMQDR	836.332	55...	P11712 P10632 P33260 F33261	Cytochrome P450 2C9 Cytochrome P450 2C8 Cytochrome P450 2C18 Cytochrome P450 2C19
2	SHMPYTD	1781.852	55...	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	YSDLVPTG	1899.958	55...	P10632	Cytochrome P450 2C8
4	NYLIPK	746.433	55...	P11712 P33261 P10632 F33260	Cytochrome P450 2C9 Cytochrome P450 2C19 Cytochrome P450 2C8 Cytochrome P450 2C18
1	GTTIMALLT	1713.897	55...	P10632	Cytochrome P450 2C8
1	EFPNPNIF	1885.900	55...	P10632	Cytochrome P450 2C8
2	SDYFMPFS	1248.548	55...	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	ICAGEGLAR	888.449	55...	P10632	Cytochrome P450 2C8
2	MELFLFLT	2084.138	55...	P10632 P33260	Cytochrome P450 2C8 Cytochrome P450 2C18
1	SVDDLK	675.344	55...	P10632	Cytochrome P450 2C8
1	NLNTTAVTK	960.524	55...	P10632	Cytochrome P450 2C8
1	GIVSLPPS	1731.927	55...	P10632	Cytochrome P450 2C8

Figure 4: View of the query tool similar to figure 3 except that the columns protein accessions and protein names are expanded to show all entries of the hightlighted table rows.

In case a peptide is homolog in another or several other proteins, the protein name information is very useful to evaluate the results in more detail; the peptide SPCMQDR for example is found to be present in all four CYP2C isoforms (CYP2C8, CYP2C9, CYP2C18, and CYP2C19). Furthermore the peptide FSLTTLR can be used indeed to differentiate CYP2C8 from the other members of the CYP2C subfamily, but it is homolog in another CYP isoform, namely CYP2E1. Finally the tool calculates eight unique peptides for CYP2C9, 19 unique peptides for CYP2C18 and 15 unique peptides for CYP2C19.

These results show that even for highly homologous proteins in general enough theoretically generated unique peptides exist. Nevertheless, by disfavored digestion or ionization of some peptides or due to the low abundance of the protein in general proteins often get identified by only few peptides. The application of UPF

tremendously reduces the time needed to argue about the uniqueness of those peptides to minutes in comparison to hours if a manual Blast search is done.

3. c. Configuring the query tool

Just as the command line tool, the query tool was configured by use of a configuration file (named QueryToolProp.txt), which is also designed in the windows ini file style. In the following opportunities for customizing the query tool are discussed.

Different sections need to adjusted for a adopting UPF to a user specific situation (e.g. network structure, used databases)

The [Database] section looks like:

```
[Database]
DBMS=mssqlserver
DBdriverMS_SQLServer=net.sourceforge.jtds.jdbc.Driver
DBdriverMySQL=com.mysql.jdbc.Driver
IP=127.0.0.1
DBname=uniquepept
user=root
password=secret
```

Users need to specify the keys *DBMS, IP, user* and *password*. In the above given example Microsoft SQL Server 2000 is used as database management system (value *mssqlserver*). Because this database is located on the local computer, the localhost (127.0.0.1) is assigned to the key *IP*. The name of the database is *uniquepept* and the user *root* with password *secret* is used to gain access to the database. When using a MySQL database please assign the value *mysql* to the key *DBMS*.

Three further sections (named [Databases], [Queries] and [Enzymes]) have similar structure. These sections were used to customize the graphical user interface (GUI) of the query tool without editing the source code. Examples are listed below:

```
[Databases]
numbDatabases=3
DB1=IPIhuman
DB2=IPIrat
DB3=UniProt/Swiss-ProtHum


[Queries]
numbQueries=4
nameQuery1=Query_1
nameQuery2=Query_2
nameQuery3=Query_3
nameQuery4=Query_4

[Enzymes]
numbEnzymes=1
enz_1=Trypsin_strict
```

Each section refers to another combo box in the tool bar (cp. Fig. 2). In the example the key and value pairs *numbDatabases=3, numbQueries=4* and *numbEnzymes=3* define the number of the databases, queries and enzymes, which names will be shown in the related combo box. These names are specified in the remaining lines of each section.

With the given structure of the [Databases] section there are three different pre-digested databases available that were generated with the command line tool.

For example, the first item named *IPIhuman* refers to a database which was generated with respect to the IPI database, which contains human protein entries. The third item named *UniProt/Swiss-ProtHum* refers to a subset of the swissprot database, which contains only human entries.

After generating further databases with the command line tool the user have to adopt this section for executing database queries against this database. Simply add a new key and value pair named *DB4=DBname*. In doing so, the value *DBname* can be arbitrarily chosen. Of course, the value should be as short and concrete as

possible. The new database can then be selected from the combo box in the middle of the tool bar.

The [Enzymes] section can be extended in a similar way. However, after adding a new key and value pair, it is necessary to add a new section configuration file. Let's assume that you want to perform studies using pepsin for enzymatic digestion. Simply add 1 to the value of the key *numbEnzymes*. Then append a further line to the [Enzymes] section like enz_X=Pepsin where X has to be chosen with respect to the number of other enzymes in the section.

Furthermore add a new section to the configuration file by copying the [Trypsin] section. Change the header into [Pepsin]. Also change the values of the keys *cleavageSites, isEnzymeNotPException and isEnzymeNTerminus.* Then the new section that extends UPF for digestion with pepsin should look like (changed values are printed in red):

[Pepsin]
modifyAllPeptideTermina=true
isCompleteModification=false
iAverage=true
isEnzymeNTerminus=true
isEnzymeNotPException=true
usePrimeAndSecMods=false
isMultipleModifications=false
isCombinedDigest=false
cleavageSites=LF

Note that the header of the new section must correspond to the value given in the [Enzymes] section.

The [Queries] section deals with different database queries. Like in the [Enzymes] section each value of the *nameQueryX* (where X denotes a successive number) corresponds to a header of a further section in the configuration file. Of course the user can edit both corresponding values in order to achieve a more descriptive representation in the graphical user interface.

The Query tool is shipped with four different queries. Query_1 and Query_2 deal with database queries that retrieve information from the peptide table. Because the peptide table does not vary with respect to a given input database both queries can be applied to any pre-digested database, e.g. IPI or UniProtKB/Swiss-Prot databases.

Query_3 and Query_4 are designed to retrieve additional information concerning the protein entries of UniProtKB/Swiss-Prot (Query_3) and IPI (Query_4) databases. Currently these queries can not be customized by the user, because some information about the structure of the sql query is assumed within the source code (e.g. data types). This information is indispensible for an adequate representation of the results in the summary table.

However it is planned to integrate opportunity for customizing the queries in order to match information from the headers of other databases into future UPF versions.

Because finding unique peptides depends on both the proteins of an input database and the chosen enzyme, each combination of database and enzyme is represented by a section in the configuration file. For example the section that deals with database tables obtained from tryptic cleavage of FASTA formatted IPI database that contains human protein entries only may be look like:

```
[IPIhumanTrypsin]
ProteinTable=IPIHUMANTRYP_prot
PeptideTable=IPIHUMANTRYP_pept
```

The header of the section is combined from the corresponding entries from the [Databases] and the [Enzymes] section. It is expected that the identifier of the database is given as the first part of the header name followed by the identifier of the enzyme. No blank or any other additional character should be inserted into this combination.

Within the section the full names of both the protein table and the peptide table, which have been prepared by the command line tool before, are given.

4. References

Alterovitz G, Liu J, Chow J, Ramoni MF (2006). Automation, parallelism, and robotics for proteomics. Proteomics 6: 4016-4022.

Janecki DJ, Bemis KG, Tegeler TJ, Sanghani PC, Zhai L, Hurley TD, Bosron WF, Wang M (2007). A multiple reaction monitoring method for absolute quantification of the human liver alcohol dehydrogenase ADH1C1 isoenzyme. Anal. Biochem. 369: 18-26.

Kocher T, Superti-Furga G (2007). Mass spectrometry-based functional proteomics: from molecular machines to protein networks. Nat Methods 4: 807-815.

Le Blanc JC, Hager JW, Ilisiu AM, Hunter C, Zhong F, Chu I (2003). Unique scanning capabilities of a new hybrid linear ion trap mass spectrometer (Q TRAP) used for high sensitivity proteomics applications. Proteomics 3: 859-869.

Lohaus C, Nolte A, Bluggel M, Scheer C, Klose J, Gobom J, Schuler A, Wiebringhaus T, Meyer HE, Marcus K (2007). Multidimensional chromatography: a powerful tool for the analysis of membrane proteins in mouse brain. J. Proteome Res. 6: 105-113.

Mayya V, Rezual K, Wu L, Fong MB, Han DK (2006). Absolute quantification of multisite phosphorylation by selective reaction monitoring mass spectrometry: determination of inhibitory phosphorylation status of cyclin-dependent kinases. Mol. Cell. Proteomics 5: 1146-1157.

Wolf-Yadlin A, Hautaniemi S, Lauffenburger DA, White FM (2007). Multiple reaction monitoring for robust quantitative proteomic analysis of cellular signaling networks. Proc. Natl. Acad. Sci. U. S. A. 104: 5860-5865.

5. Disclaimer

SOFTWARE DISCLAIMER

SOFTWARE OBTAINED FROM THE MEDIZINISCHES-PROTEOM-CENTER (MPC) IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF FITNESS FOR A PURPOSE, OR THE WARRANTY OF NON-INFRINGEMENT. WITHOUT LIMITING THE FOREGOING, THE MPC MAKES NO WARRANTY THAT:

1. THE SOFTWARE WILL MEET YOUR REQUIREMENTS
2. THE SOFTWARE WILL BE UNINTERRUPTED, TIMELY, SECURE OR ERROR-FREE
3. THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF THE SOFTWARE WILL BE EFFECTIVE, ACCURATE OR RELIABLE
4. THE QUALITY OF THE SOFTWARE WILL MEET YOUR EXPECTATIONS
5. ANY ERRORS IN THE SOFTWARE OBTAINED FROM THE MPC WILL BE CORRECTED.

SOFTWARE AND ITS DOCUMENTATION MADE AVAILABLE FROM THE MPC:

6. COULD INCLUDE TECHNICAL OR OTHER MISTAKES, INACCURACIES OR TYPOGRAPHICAL ERRORS. THE MPC MAY MAKE CHANGES TO THE SOFTWARE OR DOCUMENTATION MADE AVAILABLE DIRECTLY FROM THE MPC OR ON ITS WEB SITE.
7. MAY BE OUT OF DATE, AND THE MPC MAKES NO COMMITMENT TO UPDATE SUCH MATERIALS.

THE MPC ASSUMES NO RESPONSIBILITY FOR ERRORS OR OMMISSIONS IN THE SOFTWARE OR DOCUMENTATION.

IN NO EVENT SHALL THE MPC BE LIABLE TO YOU OR ANY THIRD PARTIES FOR ANY SPECIAL, PUNITIVE, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT THE MPC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS SOFTWARE.

THE USE OF THE SOFTWARE OPTAINED FROM THE MPC IS DONE AT YOUR OWN DISCRETION AND RISK AND WITH AGREEMENT THAT YOU WILL BE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT RESULTS FROM SUCH ACTIVITIES. NO ADVICE OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED BY YOU FROM THE MPC OR FROM THE MPC WEB SITE SHALL CREATE ANY WARRANTY FOR THE SOFTWARE.