Utilities for Mass Spectrometry Analysis of Proteins

User's Manual

Version 2.1.0

May 2020

To download Utilities for Mass Spectrometry Analysis of Proteins visit:
www.umsap.nl

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Utilities for Mass Spectrometry Analysis of Proteins (UMSAP) is a graphical user interface (GUI) designed to speed up the post-processing of data obtained during mass spectrometry studies involving proteins. The program is not intended to analyze a mass spectrum or a mass chromatogram, neither to identify the peaks in a mass spectrum. The main objective is the fast post-processing of the vast amount of data generated in mass spectrometry experiments involving proteins after peak identification have been performed.

The program is organized in modules with each module performing a single type of data post-processing. The reason for this clear separation is the high dependency between the type of mass spectrometry experiment performed and the way in which the resulting data must be post-processed. The modules are designed in such a way that the required user input is minimized but still users can control every aspect of the analysis. Currently, the software contains four modules but several others are already planned.

1.1 Citing Utilities for Mass Spectrometry Analysis of Proteins

If you publish in any way results obtained with UMSAP, please acknowledge the use of UMSAP by including the following sentence:

"Utilities for Mass Spectrometry Analysis of Proteins (UMSAP) was developed by Kenny Bravo Rodriguez at the University of Duisburg-Essen."

Any published work, which uses UMSAP, should include the following reference:

Kenny Bravo-Rodriguez, Birte Hagemeier, Lea Drescher, Marian Lorenz, Michael Meltzer, Farnusch Kaschani, Markus Kaiser and Michael Ehrmann. (2018). Utilities for Mass Spectrometry Analysis of Proteins (UMSAP): Fast post-processing of mass spectrometry data. Rapid Communications in Mass Spectrometry, 32(19), 1659–1667.

Electronic documents should include a direct link to the official UMSAP web page at: www.umsap.nl

1.2 Acknowledgments

I would like to thank all the persons that have contributed to the development of UMSAP, either by contributing ideas and suggestions or by testing the code. Special thanks goes to: Dr. Farnusch Kaschani, Dr. Juliana Rey and Prof. Dr. Daniel Hoffmann.

In particular, I would like to thank Prof. Dr. Michael Ehrmann for the support and useful discussions during my postdoc stay in his group at the University of Duisburg-Essen.

1.3 Copyrights Notes

UMSAP 2.1.0 is written in Python and uses the following modules and Python version:

Module	Version
Biopython	1.73
Matplotlib	3.0.2
NumPy	1.16.1
PyInstaller	3.4
Python	3.7.1
Requests	2.21.0
wxPython	4.0.4

Table 1.1: List of modules used by UMSAP.

The Copyrights Notes or License Agreements for the modules are as follow:

Biopython

Biopython is currently released under the "Biopython License Agreement" (given in full below). Unless stated otherwise in individual file headers, all Biopython's files are under the "Biopython License Agreement".

Some files are explicitly dual licensed under your choice of the "Biopython License Agreement" or the "BSD 3-Clause License" (both given in full below). This is with the intention of later offering all of Biopython under this dual licensing approach.

Biopython License Agreement

Permission to use, copy, modify, and distribute this software and its documentation with or without modifications and for any purpose and without fee is hereby granted, provided that any copyright notices appear in all copies and that both those copyright notices and this permission notice appear in supporting documentation, and that the names of the contributors or copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific prior permission.

THE CONTRIBUTORS AND COPYRIGHT HOLDERS OF THIS SOFTWARE DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE

FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

BSD 3-Clause License

Copyright (c) 1999-2019, The Biopython Contributors All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Matplotlib

Copyright Policy

John Hunter began matplotlib around 2003. Since shortly before his passing in 2012, Michael Droettboom has been the lead maintainer of matplotlib, but, as has always been the case, matplotlib is the work of many.

Prior to July of 2013, and the 1.3.0 release, the copyright of the source code was held by John Hunter. As of July 2013, and the 1.3.0 release, matplotlib has moved to a shared copyright model.

matplotlib uses a shared copyright model. Each contributor maintains copyright over their contributions to matplotlib. But, it is important to note that these contributions are typically only changes to the repositories. Thus, the matplotlib source code, in its entirety, is not the copyright of any single person or institution. Instead, it is the collective copyright of the entire matplotlib Development Team. If individual contributors

want to maintain a record of what changes/contributions they have specific copyright on, they should indicate their copyright in the commit message of the change, when they commit the change to one of the matplotlib repositories.

The Matplotlib Development Team is the set of all contributors to the matplotlib project. A full list can be obtained from the git version control logs. License agreement for matplotlib 3.0.3

1. This LICENSE AGREEMENT is between the Matplotlib Development Team ("MDT"), and the Individual or Organization ("Licensee") accessing and otherwise using matplotlib software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, MDT hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib 3.0.3 alone or in any derivative version, provided, however, that MDT's License Agreement and MDT's notice of copyright, i.e., "Copyright (c) 2012-2013 Matplotlib Development Team; All Rights Reserved" are retained in matplotlib 3.0.3 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib 3.0.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib 3.0.3.

4. MDT is making matplotlib 3.0.3 available to Licensee on an "AS IS" basis. MDT MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, MDT MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB 3.0.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. MDT SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB 3.0.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB 3.0.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between MDT and Licensee. This License Agreement does not grant permission to use MDT trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 3.0.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.

NumPy

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PyInstaller

==========
The PyInstaller licensing terms
==========

PyInstaller is licensed under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bootloader Exception

In addition to the permissions in the GNU General Public License, the authors give you unlimited permission to link or embed compiled bootloader and related files into

combinations with other programs, and to distribute those combinations without any restriction coming from the use of those files. (The General Public License restrictions do apply in other respects; for example, they cover modification of the files, and distribution when not linked into a combine executable.)

Bootloader and Related Files

Bootloader and related files are files which are embedded within the final executable. This includes files in directories:

./bootloader/
./PyInstaller/loader

About the PyInstaller Development Team

The PyInstaller Development Team is the set of contributors to the PyInstaller project. A full list with details is kept in the documentation directory, in the file "doc/CREDITS.rst".

The core team that coordinates development on GitHub can be found here: https://github.com/pyinstaller/pyinstaller. As of 2015, it consists of:

* Hartmut Goebel
* Martin Zibricky
* David Vierra
* David Cortesi

Our Copyright Policy

PyInstaller uses a shared copyright model. Each contributor maintains copyright over their contributions to PyInstaller. But, it is important to note that these contributions are typically only changes to the repositories. Thus, the PyInstaller source code, in its entirety is not the copyright of any single person or institution. Instead, it is the collective copyright of the entire PyInstaller Development Team. If individual contributors want to maintain a record of what changes/contributions they have specific copyright on, they should indicate their copyright in the commit message of the change, when they commit the change to the PyInstaller repository.

With this in mind, the following banner should be used in any source code file to indicate the copyright and license terms:

Copyright (c) 2005-20l5, PyInstaller Development Team.

Distributed under the terms of the GNU General Public License with exception for distributing bootloader.

The full license is in the file COPYING.txt, distributed with this software.

GNU General Public License

https://gnu.org/licenses/gpl-2.0.html

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full

compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Python

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 3.7.3 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.7.3 alone or in any derivative version, provided, however, that PSF's License

Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2019 Python Software Foundation; All Rights Reserved" are retained in Python 3.7.3 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.7.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.7.3.

4. PSF is making Python 3.7.3 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.7.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.7.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.7.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 3.7.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.

Requests

Copyright 2018 Kenneth Reitz

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License here.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

wxPython

Preamble

The licencing of the wxWidgets library is intended to protect the wxWidgets library, its developers, and its users, so that the considerable investment it represents is not abused.

Under the terms of the original wxWidgets licences, you as a user are not obliged to distribute wxWidgets source code with your products, if you distribute these products in binary form. However, you are prevented from restricting use of the library in source

code form, or denying others the rights to use or distribute wxWidgets library source code in the way intended.

The wxWindows Library License establishes the copyright for the code and related material, and it gives you legal permission to copy, distribute and/or modify the library. It also asserts that no warranty is given by the authors for this or derived code.

The core distribution of the wxWidgets library contains files under two different licences:

* Most files are distributed under the GNU Library General Public License, version 2, with the special exception that you may create and distribute object code versions built from the source code or modified versions of it (even if these modified versions include code under a different licence), and distribute such binaries under your own terms.

* Most core wxWidgets manuals are made available under the "wxWindows Free Documentation License", which allows you to distribute modified versions of the manuals, such as versions documenting any modifications made by you in your version of the library. However, you may not restrict any third party from reincorporating your changes into the original manuals.

wxWindows Library Licence

wxWindows Library Licence, Version 3.1
==========

Copyright (c) 1998-2005 Julian Smart, Robert Roebling et al

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS LIBRARY LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public Licence for more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your

option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.

2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.

3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.

4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

wxWindows Free Documentation License

wxWindows Free Documentation Licence, Version 3
==========

Copyright (c) 1998 Julian Smart, Robert Roebling et al

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS FREE DOCUMENTATION LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. Permission is granted to make and distribute verbatim copies of this manual or piece of documentation provided any copyright notice and this permission notice are preserved on all copies.

2. Permission is granted to process this file or document through a document processing system and, at your option and the option of any third party, print the results, provided a printed document carries a copying permission notice identical to this one.

3. Permission is granted to copy and distribute modified versions of this manual or piece of documentation under the conditions for verbatim copying, provided also that any sections describing licensing conditions for this manual, such as, in particular, the GNU General Public Licence, the GNU Library General Public Licence, and any wxWindows Licence are included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

4. Permission is granted to copy and distribute translations of this manual or piece of documentation into another language, under the above conditions for modified versions, except that sections related to licensing, including this paragraph, may also be included in translations approved by the copyright holders of the respective licence documents in addition to the original English.

WARRANTY DISCLAIMER

5. BECAUSE THIS MANUAL OR PIECE OF DOCUMENTATION IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR IT, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THIS MANUAL OR PIECE OF DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE MANUAL OR PIECE OF DOCUMENTATION IS WITH YOU. SHOULD THE MANUAL OR PIECE OF DOCUMENTATION PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

6. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE MANUAL OR PIECE OF DOCUMENTATION AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE MANUAL OR PIECE OF DOCUMENTATION (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF A PROGRAM BASED ON THE MANUAL OR PIECE OF DOCUMENTATION TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Chapter 2

Obtaining and Installing Utilities for Mass Spectrometry Analysis of Proteins

2.1 Obtaining Utilities for Mass Spectrometry Analysis of Proteins

UMSAP is distributed free of charge for anyone interested in using it. To obtain a copy of the software just register at www.umsap.nl and go to the download page.

No extra software or packages are needed for UMSAP to properly work.

So far, UMSAP have been tested in MacOS X 10.12.6 and 10.14.4 and Windows 7/10. Linux users may download the source code of the software and adapt it to their specific distribution of Linux. Support for some Linux distributions will be available in the future.

2.2 Installing Utilities for Mass Spectrometry Analysis of Proteins

Windows

Unzip the file you just downloaded from www.umsap.nl. Then, copy the folder UMSAP to the location in your file system where you want to keep it. Finally, create a shortcut to the executable file UMSAP.exe found inside the main folder UMSAP and that is all. You are now ready to use UMSAP.

MacOS X

Unzip the file you just downloaded from www.umsap.nl. Then, just move the UMSAP.app folder to /Applications/. That is all. You are now ready to use UMSAP.

Linux

Currently, there are no precompiled versions of UMSAP for Linux. Therefore, users using a computer running Linux need to install all the required modules before using UMSAP. For UMSAP 2.1.0 the list of required modules is:

Biopython 1.73, Matplotlib 3.0.2, NumPy 1.16.1, PyInstaller 3.4, Python 3.7.1, Requests 2.21.0 and wxPython 4.0.4.

If all these modules are already installed in the computer, then using UMSAP is straightforward. Go to the Downloads page and download the source files for UMSAP 2.1.0. Unzip the files. In the terminal, navigate to the newly created UMSAP folder and type python UMSAP.py. This will launch the GUI.

If the modules are not installed, then it is recommended to use conda to create a virtual environment to install everything and run UMSAP. First, check if the Linux distribution you are using (or a close enough distro) is listed here. If this is the case then you can easily install wxPython with pip as described below. If this is not the case then you have to build wxPython by yourself. Check here for how to do this.

Once you know that you can have a functional wxPython installation do the following.

First, download Miniconda for Linux from here. Then, open a terminal and navigate to the folder containing the Miniconda installer. The installer will have different names depending on which one you choose in the previous step. Once in the folder containing the Miniconda installer, execute the file with the bash command line interpreter by typing:

bash Miniconda-installer-file

and follow the on-screen instruction. After finishing the installation close the terminal and open a new one so the changes done by conda take effect.

In the new terminal window type:

conda create –name umsap
conda activate umsap
conda install python==3.7.1
pip install biopython==1.73
pip install requests==2.21.0
pip install matplotlib==3.0.2

Installing matplotlib should install numpy 1.16.1 or superior, if this is not the case type

pip install numpy==1.16.1

If you found a compatible wheel for wxPython and your Linux distribution here, then install it using pip. You will need to change the gtk version and the linux distribution to suit your case. For example, for gtk3 and ubuntu 16.04 the command line is:

pip install -U -f https://extras.wxpython.org/wxPython4/extras/linux/gtk3/ubuntu-16.04 wxPython

And finally, you are all set. Download the source files for UMSAP 2.1.0 from the Downloads page. Unzip the file and in a terminal navigate to the newly created UMSAP

folder and type python UMSAP.py. This will launch the GUI.

2.3 Uninstalling Utilities for Mass Spectrometry Analysis of Proteins

UMSAP will not create any installation file in your computer. Therefore, the only thing you need to do, to completely uninstall UMSAP, is to delete the folder UMSAP.app in MacOS X or UMSAP in Windows/Linux. In addition, you should delete any shortcut pointing to the executable file of UMSAP. That is all.

Chapter 3

Workflow in Utilities for Mass Spectrometry Analysis of Proteins

When you start UMSAP, the program will display the main window (Figure 3.1). From this window you can access all the modules and utilities either by the menu entries: Modules and Utilities or by the corresponding buttons in the right side list. A complete description of each module and utility is given in the following chapters.

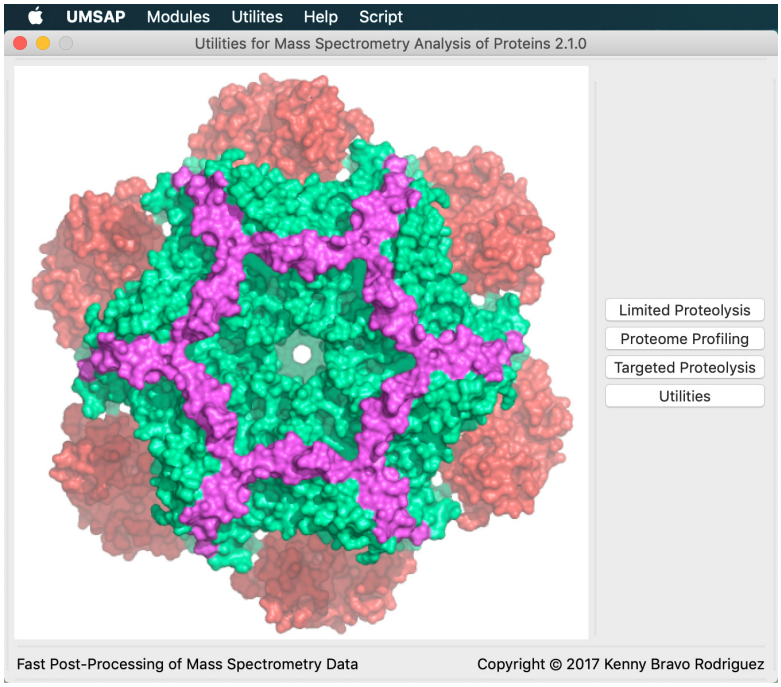

Figure 3.1: The main window of UMSAP. From this window users can access all the available Modules and Utilities.

3.1 The input data files

The main input data file for UMSAP is the file containing the detected peptide sequences after all peak assignments have been completed. The program expects this file to be a plain text file containing a table with the data. Columns in the file are expected to be tab separated. The first row in the file is expected to contain only the names of the columns. There is no limitation in the amount and type of data present in the data file. However, each module will expect certain columns to be present. Columns not needed by the modules will simply be ignored.

In addition, certain modules use other input files as well. The modules Targeted Proteolysis and Limited Proteolysis may use fasta files containing the sequences of the recombinant and native proteins used in the experiments. The fasta files must contain only one protein sequence. Alternatively, instead of a fasta file, a plain text file can be used. In this case, the plain text file must contain only the sequence of a single protein using the one letter code for amino acids. The Targeted Proteolysis module may also use a pdb file. The sequence and pdb files can be directly download from UNIPROT or PDB, respectively, by supplying the correct codes.

Certain output files generated by UMSAP can be used as input data files as well. Files that can be used this way will contain the results from previous data post-processing. These output files will be plain text files, since in some cases users will need to directly read the data contained in these files. However, it is important that these files remain unchanged to ensure the correct display of the results contained in the files by UMSAP.

3.2 Using Utilities for Mass Spectrometry Analysis of Proteins

Once you have your input data files, using UMSAP is straightforward. Just open the program and select a module or utility. In the new window, fill in the needed information and hit the Start analysis button in the bottom right corner. Depending on the amount of data and the complexity of the analysis to perform it may take a few minutes for the program to complete the task at hand. In each window performing an analysis, there is a progress bar near the Start analysis button that gives a rough guess of the remaining time needed to complete the current analysis.

Currently, the text boxes showing the path to a file do not have auto-complete capabilities. In addition, relative paths are not supported. Therefore, the full paths to files are expected. In the case of output files and folders if the text boxes are left empty, appropriate defaults values will be provided, based on the main input file path.

In order to make the program as user friendly as possible help messages will pop up from buttons and labels. The help messages will contain a brief description of what is the button or label for and what input is expected from the user. In this way, users can find basic information about a particular element of the interface without needing to go to the manual or online tutorials. If more information is needed, users may consult the manual or click the Help button in the bottom left corner of the module/utility window

to read an online tutorial.

Depending on the module or utility just run, new windows will be created to show a graphical representation of the results.

Special care have been taken to handle errors that may appear during the processing of the input files. Errors will be reported to users using dialog boxes with plain English messages so users may correct the errors right away and continue with the analysis. In the case of an unforeseen error occurring, the message error will be more code oriented. It will be helpful if users send a crash report to umsap-crashreport@umsap.nl so we can correct the error and/or create an appropriate error message.

In general, windows showing a graphical representation of results will be allowed to resize while module/utilities windows will not. The Tools menu will have options specific for each window, for example, to save an image of the results displayed in a results window. In most of the windows the functionalities in the Tools menu can be also accessed with the right mouse button.

For the modules, UMSAP will create a .uscr file after finishing the analysis. This is an input file that can be used to run an analysis one more time without users having to type in again all the needed information. Once the .uscr file is selected, using the Run Input File entry in the Script menu, the appropriate module will be launched and the information found in the .uscr file will be used to fill the needed data in the interface of the module. At this point, users may decide to update the analysis by changing some of the parameters in the interface of the module. Clicking the Start analysis button will start the analysis of the input data files.

3.3 Navigating through Utilities for Mass Spectrometry Analysis of Proteins

The entries Modules and Utilities will be available in the menu of every window. The Modules entry in the menu gives direct access to all modules. The same is true for the Utilities entry. These menu entries are the fastest way to access all the functions in UMSAP. In a typical UMSAP session, users will work with different independent windows simultaneously. The windows have descriptive names so users can quickly guess the content of any window. The scheme of the windows name is *UMSAP - Utilities or Module Name - Name of the window*. For example, the window with name *UMSAP - Utilities - Correlation coefficients (t1.corr)* will be displaying the correlation coefficient matrix saved in the file *t1.corr*.

3.3.1 Keyboard shortcuts

The following table contains the built-in keyboard shortcuts.

Shortcut	Action
Ctrl/Cmd+Q	Quit UMSAP
Ctrl/Cmd+D	Close current window
Ctrl/Cmd+R	Read UMSAP output file
Ctrl/Cmd+I	Read UMSAP .uscr file
Alt+Cmd+L	Launch the Limited Proteolysis module
Alt+Cmd+P	Launch the Proteome Profiling module
Alt+Cmd+T	Launch the Targeted Proteolysis module
Alt+Cmd+U	Launch the Utilities window
Ctrl/Cmd+L	Change selection mode in the Gel Analysis window

Table 3.1: List of built-in keyboard shortcuts.

3.4 The output files

In order to minimize user input, the names for output files and output folders can be left unspecified. In this case, UMSAP will use default names for files and folders. When names for files and folders are given, it is recommended to use names with no spaces, e.g. my-output-file or myoutputfile.The software will avoid to overwrite previous output files or folders. If a selected folder already exist and it is not empty, UMSAP will create a new folder to write in the results. The location of the new folder will be the same as the selected folder. The name of the new folder will be the same as the selected folder plus the current date and time to the second, for example, selectedfolder-20190425092904. The same is true for files unless the user forces UMSAP to overwrite a file.

3.5 Backward compatibility

UMSAP is capable to read the .tarprot file from previous versions. Other files generated by UMSAP are version dependent and cannot be read with different versions of UMSAP. Nevertheless, two tools allows to quickly generate any UMSAP file based on a .tarprot file from a previous version of UMSAP. These tools are described in subsection 7.2.7 and subsection 7.2.8. How to generate a .tarprot file is explained in Chapter 6.

Chapter 4

The Limited Proteolysis module

The Limited Proteolysis module is designed to post-process the results from an enzymatic digestion performed in two steps. The first step is assumed to be a limited proteolysis in which a large protein is split in smaller fragments. The fragments are then separated using a SDS-PAGE electrophoresis. Finally, selected bands from the gel are submitted to a full enzymatic digestion and the generated peptides analyzed using mass spectrometry.

4.1 Definitions

Before explaining in detail the interface of the module and how does the module works, lets make clear the meaning of some terms that will be used in the following paragraphs and chapters.

- *Recombinant protein*: actual amino acid sequence used in the mass spectrometry experiments. It may be identical to the native sequence of the target protein under study or not.
- *Native protein*: full amino acid sequence expressed in wild type cells.
- *Detected peptide*: any peptide detected in any of the mass spectrometry experiments including the control experiments.
- *Relevant peptide*: a detected peptide with a Score value above a user defined threshold (see page **??**).
- *Filtered peptide*: a relevant peptide with an equivalent behavior in the control and a given gel band at the chosen significance level.
- *Fragment*: group of filtered peptides with no gaps when their sequences are aligned to the sequence of the recombinant/native protein.

For example, there are three fragments in the alignment shown below. The first fragment is formed by sequences 1 to 3 since there is no gap in the sequence MKKTAIAIAVAL.

SEQ4 forms the second fragment because there is a gap between the last residue in SEQ3 and the first residue in SEQ4 and another gap between the last residue in SEQ4 and the first residue in SEQ5. For the same reason SEQ5 forms the third fragment.

```
REC.PROT  MKKTAIAIAVALAGFATVAQAASWSHPQFEKIEGRRDRGQKTQSAPGTLS    50
SEQ1      MKKTAIAIAV..........................................    10
SEQ2      ..KTAIAIAV..........................................     8
SEQ3      .....IAIAVAL........................................     7
SEQ4      ..............ATVAQAASWS............................    10
SEQ5      ....................................DRGQKTQSAPG...    11
```

4.2 The input data files

The Limited Proteolysis module requires a data file containing the detected peptides and a sequence file containing the amino acid sequence of the recombinant protein used in the study. Both files must follow the guidelines specified in **??**. In short, the data file must have a tabular format with tab separated columns and the name of the columns are expected as first row. The sequence file is expected to contain only one sequence and to be FASTA formatted with or without the header line. All columns given as input in the section *Column numbers* in Region 2 of the interface must be present in the data file. Optionally, another sequence file with the sequence of the native target protein may be specified.

4.3 The interface

The window of the Limited Proteolysis module is divided in four regions (**??**).

Region 1 contains four buttons allowing users to quickly delete all provided input and start a new analysis. The Clear all button will delete all user provided input and will empty the list box in Region 3. The Clear files button will delete the path to all user provided files and will empty the list box in Region 3. The Clear values button will delete all user provided input for the section Values in Region 2. Finally, the Clear columns button will delete all user provided column numbers.

Region 2 contains the fields where users provide the information needed in order to perform the post-processing of the input data file. The section *Files* in Region 2 will provide the path to the input data and output files. It contains five buttons.

1.- The Data file button allows users to browse the file system and select a data file. Only .txt files can be selected here. Once the data file is selected, the name of the columns in the file will be shown in the list box in Region 3. If the path to the data file is typed in, the display of the name of the columns in Region 3 can be trigger by pressing the Enter key in the keyboard while the Data file entry box has the focus of the keyboard.

2.- The Sequence (rec) button allows users to browse the file system and select the file containing the sequence of the recombinant protein under study. Only .txt or .fasta files

can be selected here. Alternatively, a UNIPROT code may be given in this field.

3.- The Sequence (nat) button allows to select the file containing the sequence of the native protein. Only .txt or .fasta files can be selected here. Alternatively, a UNIPROT code may be given in this field. The sequence of the native protein is an optional field. A value of NA means that no native sequence file will be given.

See Section 3.1 for more details about the data files.

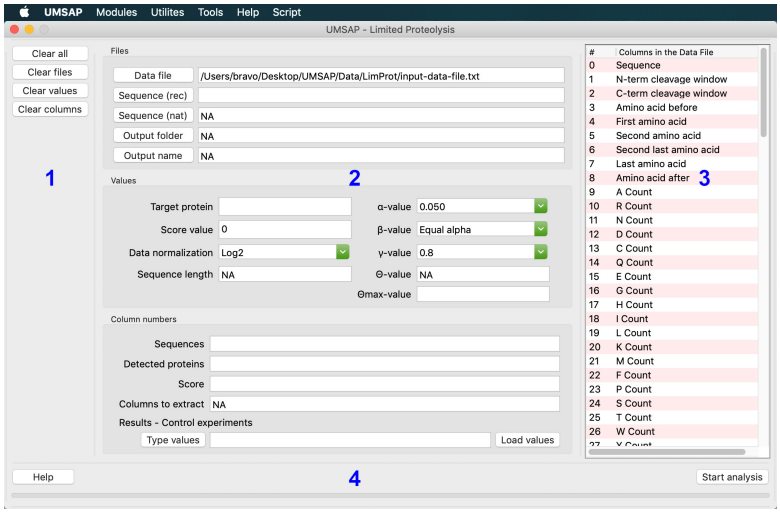

Figure 4.1: The Limited Proteolysis window. This window allows users to performed the analysis of the results obtained during a two steps enzymatic proteolysis experiment where the products from the first digestions are separated using SDS-PAGE electrophoresis. Optional parameters default to NA when the window is created. The rest of the parameters must be provided by the user.

4.- The Output folder button allows users to browse the file system and select the location of the folder that will contain the output. By default, UMSAP will create a LimProt folder inside the selected Output folder to save all the generated results. If only the name of the output folder is given, the output folder will be created in the same folder containing the Data file. If this field is left empty, then the LimProt folder will be created in the same folder containing the Data file. If the selected output folder already contains a LimProt folder, then the current date and time to the seconds will be added to the name in order to avoid overwriting the files from previous analyses.

5.- The Output name button does nothing but the text box to its right allows users to specify the name of the files that will be generated during the analysis. If this field is left empty, then the name limprot will be used for the output files.

The section *Values* in Region 2 contains nine parameters. Here, users provide information about the target protein, how the data file should be processed and which optional analysis will be performed.

1.- The parameter Target protein allows users to specify which of the proteins detected in the MS experiments was used as substrate during the enzymatic digestions. Users may type here any unique protein identifier present in the Data file. The search for the

Target protein is case sensitive, meaning that eFeB is not the same as efeb.

2.- The parameter Score value allows users to define a threshold value above which the detected peptides will be considered as relevant. The Score value is an indicator of how reliable was the detection of the peptide during the MS experiments. The value given to UMSAP depends on the program generating the Data file. Only one real number equal or greater than zero will be accepted as a valid input here. A value of zero means all detected peptides belonging to the Target protein will be treated as relevant sequences.

3.- The parameter Data normalization allows selecting the normalization procedure to be performed before running the analysis of the data in the Data file. Currently, only a $\log_2$ normalization is possible but this will be expanded to include quantile, variance stabilization and local regression normalization, among other methods.

4.- The parameter Sequence length allows users to define the number of residues per line in the sequence alignment file, see **??** for more details. Only one integer number greater than zero will be accepted here. A value of NA means that the sequence alignment file will not be generated.

5–9.- The parameters α, β, γ, Θ and Θmax are used to adjust the equivalence test(**Limentani2005**) performed to identify peptides in the selected gel bands with a similar intensity to the control bands. See Section 4.4 for more details.

The section *Column numbers* in Region 2 contains five parameters. Here, users provide the column numbers in the Data file from where UMSAP will get the information needed to perform the analysis of the limited proteolysis results. All columns specified in this section must be present in the Data file. Users must be aware that Python starts counting from 0. Therefore, the number of the columns in the Data file starts from 0 and not from 1. The column numbers displayed in the list box in Region 3 after the Data file is selected can be directly used for the parameter values.

1.- The parameter Sequences allows users to specify the column in the Data file containing the sequences of the peptides identified in the MS experiments. Only one integer number equal or greater than zero will be accepted here.

2.- The parameter Detected proteins allows users to specify the column in the Data file containing the unique protein identifier for the proteins detected in the MS experiments. It is in this column where the program will look for the Target protein value given in the section *Values* in Region 2 of the interface. Only one integer number equal or greater than zero will be accepted here.

3.- The parameter Score allows users to specify the column in the Data file containing the Score values. It is in this column where the program will look for the values to be compared against the Score threshold given in section *Values* of Region 2 of the interface.

4.- The parameter Columns to extract allows users to specify which columns in the Data file will be copy to the shorter versions of the Data file, see page **??** for more details. A range of columns may be specified as 4–10. Any number of columns may be specified here. Only integer numbers equal or greater than zero will be accepted. A value of NA means no shorter version of the Data files will be created.

5.- The parameter Results - Control experiments allows users to specify the columns in

the Data file containing the results of the control and enzymatic digestion experiments. There are three ways to provide the information for this parameter. Users can directly type the column numbers corresponding to the control and digestion experiments, load the values from a .txt file using the Load values button or use the Type values button to call a helper window, see Figure 5.2. Independently of the chosen method, the expected input here is a matrix in which each element contains the column numbers with the MS results for a gel spot. Duplicate values are not allowed.

The helper window is divided in four Regions. Region 1 allow to define the number of bands and lanes of interest in the gel and create the matrix in Region 2. Each text field in Region 2 should contain the column numbers containing the MS results for the given gel spot. The values for the text fields should be positive integer numbers or a range of integers, e.g. 60–62 or NA for empty gel spots. The column numbers can be seen in the list box in Region 3. Selected entries in the list box can be copied and then pasted to the text fields using the right mouse button or the Tools menu. Region 4 contains two buttons to Cancel or to Export the values to the window of the module.

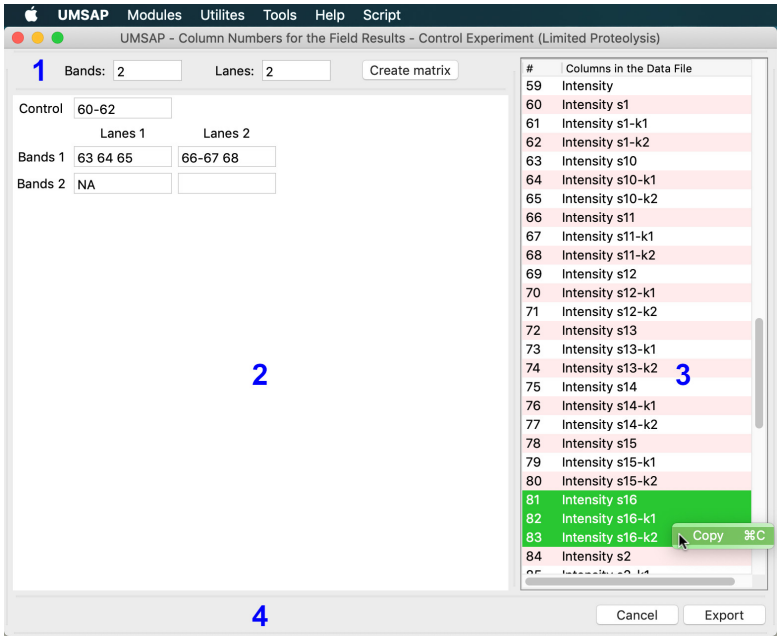

Figure 4.2: The Result - Control experiments helper window. This window allows users to specify the column numbers in the Data file containing the MS results for the selected gel spots.

The column numbers can also be loaded from a .txt file. The format of the file content is very simple. Each row of the file will be a row of the matrix and column's content are separated by a comma (,). The first row in the file will be the columns for the control experiment and no commas are needed in this case. The following example will create the same matrix shown in Figure 5.2 after the .txt file is loaded.

60–62
63 64 65, 66–67 68
NA, 70–72

The same matrix can be directly typed in the text field of the parameter Results - Control experiments of Region 2 of the interface of the module. The text to typed is similar to the content of the .txt file discussed in the previous paragraph. The only difference is that semicolons (;) are used to separate rows in the matrix. Text example:

60–62; 63 64 65, 66–67 68; NA, 70–72

Region 3 of the Limited Proteolysis module main window contains a list box that will display the number and name of the columns found in the Data file. The list box is automatically filled when the Data file is selected. Selected columns in the list box can be directly added to any field in the section *Column numbers* in Region 2 of the interface using the right mouse button over the list box or the Tools menu.

Region 4 contains two buttons and the progress bar. The Help button leads to an online tutorial while the Start analysis button will trigger the processing of the data file. The progress bar will give users a rough idea of the remaining processing time.

4.3.1 The Tools menu

The tools menu in the module window allows to copy the selected columns in the list box in Region 3 of the interface to the fields of section *Column numbers* in Region 2 of the interface. The list box in Region 3 of the interface can also be clear. In addition, through this menu users can create a .uscr file with the given options to the module before running the analysis. If something goes wrong during the analysis having the .uscr file means that users do not have to type the values of all the parameters again.

4.4 The analysis

First, UMSAP will check the validity of the user provided input. Then, the data file is processed as follow. All rows in the data file containing peptides that do not belong to the target protein are removed. Then, all rows containing peptides from the target protein but with Scores values lower than the user defined Score threshold are removed. These steps leave only relevant peptides, this means peptides with a Score value higher than the user defined threshold that belong to the Target protein. For each one of these relevant peptides the equivalence test is performed(**Limentani2005**).

The implementation of the equivalence test is based on the following equations:

$$s^* = s\sqrt{\frac{n-1}{\chi^2_{(\gamma,n-1)}}} \tag{4.1}$$

$$\Theta = \delta + s^* \left[t_{(1-\alpha,2n-2)} + t_{(1-\beta/2,2n-2)}\right]\sqrt{\frac{2}{n}} \tag{4.2}$$

$$(\bar{y}_1 - \bar{y}_2) \pm t_{(1-\alpha,n_1+n_2-2)} \cdot \sqrt{s_p^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)} \tag{4.3}$$

where s^* is an estimate of the upper confidence limit of the standard deviation, $\chi^2_{(\gamma,n-1)}$ is the (100γ)th percentile of the chi-squared distribution with $n-1$ degrees of freedom, Θ is the acceptance criterion, δ is the absolute value of the true difference between the group's mean values, t is the Student's t value, $\bar{y}$ is the measurement mean and s_p is the pooled standard deviation of the measurements calculated with:

$$s_p = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} \tag{4.4}$$

α, β, γ and Θ are the parameters defined in Region 2 in the interface of the module.

In essence, for each relevant peptide, the control experiments are used to estimate the upper confidence limit for the standard deviation using **??** and then the acceptance criterion is calculated with Equation 4.2 Finally, the confidence interval for the mean difference for the gel spot and the control is calculated and compare to Θ. Peptides with equivalent mean intensity in at least one gel spot and the control are retained while not equivalent peptides are discarded.

If the value of Θ is given in Region 2 of the module's interface then only the confidence interval for the mean difference is calculated and the value is directly compared to the given Θ value. The maximum possible Θ value must always be provided. The reason for this is that when only a few replicates of the experiments are performed the calculated Θ value may be to large and then the equivalence test is not able to detect the peptides with intensity values in the experiments similar to the control.

After the filtered peptide (FP) are identified the modules creates the output files.

4.5 The output files

All the output generated by the Limited Proteolysis module will be contained in a LimProt folder created inside the selected Output folder. If the Output folder field in the section *Files* in Region 2 of the interface is left empty, then the LimProt folder will be created in the directory containing the Data file. If the selected Output folder already contains a LimProt folder, then the current date and time to the seconds will be added to the name in order to avoid overwriting files from previous analyses. By default the LimProt folder will contain two files with extensions .limprot and .uscr and a Data_Steps folder. The name of these files is provided with the Output name field in the section *Files* of Region 2 of the interface. Depending on the user provided input extra folders and files will be created inside the LimProt folder (Figure 4.3). For the rest of this chapter we will assume that the user provided name for the Output folder was *t*, the Output name was myLimTest, the Target protein was *Mis18alpha* and all optional analyses were performed.

Information regarding the content and use of the .uscr file can be found in subsection 7.3.2.

If the parameter Sequence length is different than NA, then the file myLimTest.seq.pdf will be created. The file contains the sequence of the Target protein with the sequence of the FP highlighted. More details are given in subsection 7.1.1.

If the parameter Columns to extract is different than NA, then the folder Data will be created. The folder will contain several files that are shorter versions of the Data file. These files will contain only information regarding the Target protein. More details are given in subsection 7.3.5.

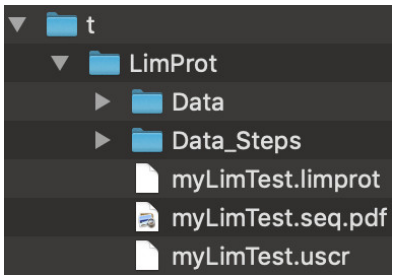

Figure 4.3: The structure of the Output folder from the Limited Proteolysis module. The folder Data and the file myLimTest.seq.pdf will be created only if requested.

4.6 Visualizing the output files

The main output of the module is the .limprot file. This file contains the list of FP, all parameters values and all the information needed to visualize the results with UMSAP. After creating the file at the end of the analysis the Limited Proteolysis module will automatically load the file and create a windows to display the results, see Figure 4.4.

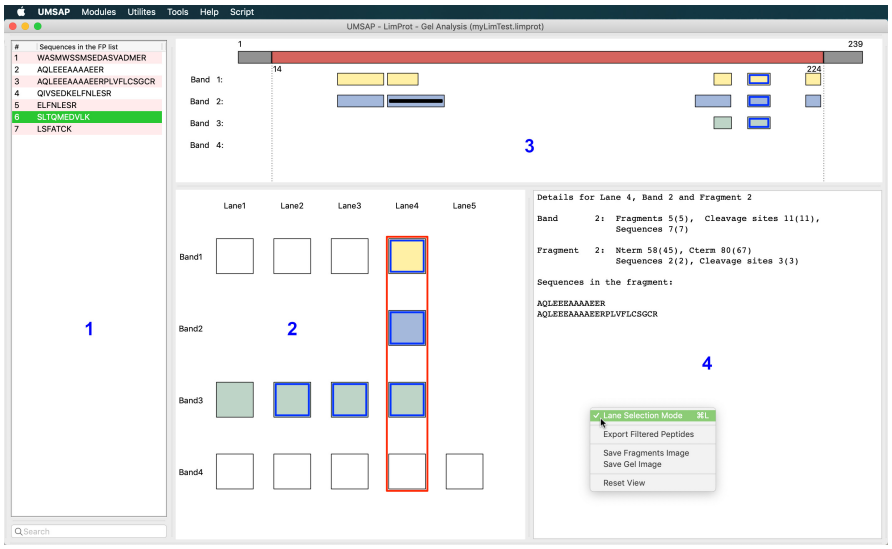

Figure 4.4: The Gel Analysis window. Users can performed here the analysis of the fragments obtained in the limited proteolysis experiments.

The Gel analysis window is divided in four Regions.

Region 1 contains a list of all FP contained in the .limprot file being shown. The search box at the bottom allows to search for a sequence in the list of FP.

Region 2 contains a representation of the analyzed gel. Here, each gel spot is represented with square. When a square is not shown this means that the corresponding gel spot was not analyzed. Empty squares represent gel spot where no peptide from the Target protein was detected with intensity values equivalent to the controls. The rest of the square will be colored according to the band they belong to or the lane. There are two selection modes available for Region 2. In the Lane selection mode selecting one of the Gel spot will also select the entire lane containing the selected gel spot. In this mode the gel spot will be colored according to the band they belong to. The selected lane will be highlighted with a red rectangle. The right mouse button, the Tools menu or the keyboard shortcut Ctrl/Cmd+L can be used to change the selection mode. In the Band selection mode, selecting a gel spot will highlight the band containing the gel spot and the gel spot are colored by lane. Selecting a band or a lane in Region 2 will display information about the band/lane in Regions 3 and 4.

Region 3 will display a graphical representation of the fragments found in each gel spot for the selected band or lane. The first fragment in this region represent the full length of the recombinant sequence of the Target protein. Here, the central red section represents the sequence in the recombinant protein that is identical to the native protein sequence while gray sections represent the sequences in the recombinant protein that is different to the native protein sequence. The fragments are color coded using the same colors of the band/lane they belong to.

Selecting a peptide from the list box in Region 1 will highlight with a blue border the gel spot in Region 2 where the peptide is found. If a lane/band in Region 2 is already selected, then the fragments shown in Region 3 that contains the selected peptide in the list box will also be highlighted with a blue border.

Region 4 will show information about the selected lane/band or gel spot in Region 2 or the selected fragment in Region 3. The displayed information for a selected band/lane includes the number of non-empty lanes/bands, the number of fragments identified in each non-empty gel spot in the band/lane and the protein regions identified. Selecting a gel spot will display this information only for the gel spot. Selecting a fragment in Region 3 will display in Region 4 the following information: number of cleavage sites and fragments, first and last residue number for the selected fragment and a sequence alignment of all peptides forming the fragment.

4.6.1 The Tools menu

The Tools menu for the window allows changing the selection mode in Region 2, export the list of FP in the list box in Region 1, Save an image of Region 2 or 3 and to reset the view of the window.

Chapter 5

The Proteome Profiling module

The Proteome Profiling modules is designed to identify differentially expressed protein under various experimental conditions. A typical example is to compare the effect of two substance over protein expression using whole cell lysates.

5.1 The input data files

The Proteome Profiling module requires only one input file. This Data file must follow the guidelines specified in **??**. In short, the Data file must have a tabular format with tab separated columns and the name of the columns are expected as first row. All columns given as input in the section *Column numbers* in Region 2 of the interface must be present in the Data file.

5.2 The interface

The window of the Proteome Profiling module is divided in four regions (Figure 5.1).

Region 1 contains four buttons allowing users to quickly delete all provided input and start a new analysis. The Clear all button will delete all user provided input and will empty the list box in Region 3. The Clear files button will delete the path to all user provided files and will empty the list box in Region 3. The Clear values button will delete all user provided input for the section Values in Region 2. Finally, the Clear columns button will delete all user provided column numbers.

Region 2 contains the fields where users provide the information needed in order to perform the post-processing of the input Data file. The section *Files* in Region 2 will provide the path to the input data and output files. It contains three buttons.

1.- The Data file button allows users to browse the file system and select a data file. Only .txt files can be selected here. Once the data file is selected, the name of the columns in the file will be shown in the list box in Region 3. If the path to the data file is typed in, the display of the name of the columns in Region 3 can be triggered by pressing the

Enter key in the keyboard while the Data file entry box has the focus of the keyboard.

4.- The Output folder button allows users to browse the file system and select the location of the folder that will contain the output. By default, UMSAP will create a ProtProf folder inside the selected Output folder to save all the generated results. If only the name of the output folder is given, the output folder will be created in the same folder containing the Data file. If this field is left empty, then the ProtProf folder will be created in the same folder containing the Data file. If the selected output folder already contains a ProtProf folder, then the current date and time to the seconds will be added to the name in order to avoid overwriting the files from previous analyses.

5.- The Output name button does nothing but the text box to its right allows users to specify the name of the files that will be generated during the analysis. If this field is left empty, then the name protprof will be used for the output files.

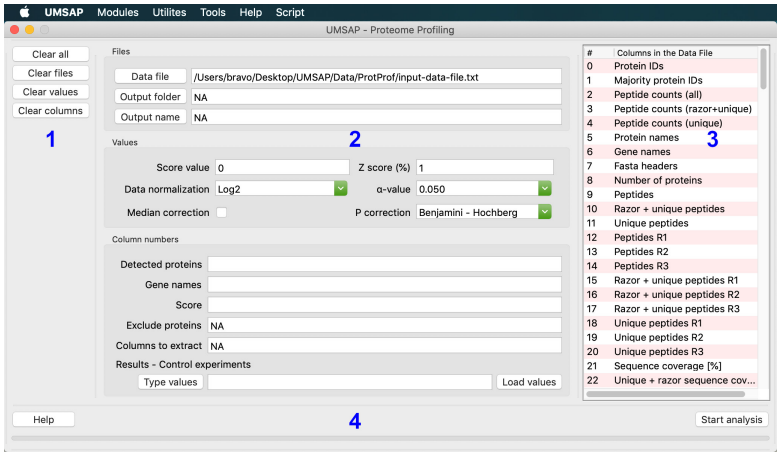

Figure 5.1: The Proteome Profiling window. This window allows users to performed a proteome profiling analysis. Optional parameters default to NA when the window is created. The rest of the parameters must be provided by the user.

The section *Values* in Region 2 of the interface contains six parameters. Here, users provide information about how the Data file should be processed and which optional analysis will be performed.

1.- The parameter Score value allows users to define a threshold value above which the detected proteins will be considered as relevant. The Score value is an indicator of how reliable was the detection of the proteins during the MS experiments. The value given to UMSAP depends on the program generating the Data file. Only one real number equal or greater than zero will be accepted as a valid input here. A value of zero means all detected proteins will be treated as relevant proteins.

2.- The parameter Data normalization allows selecting the normalization procedure to be performed before running the analysis of the data in the Data file. Currently, only a $\log_2$ normalization is possible but this will be expanded to include quantile, variance stabilization and local regression normalization, among other methods.

3.- The parameter Median correction indicates whether to apply a median correction to protein intensities in each experiment. The main advantage of this correction is to get

volcano plots that are symmetric in the x axis.

4.- The parameter Z score (%) is used to highlight the corresponding top percent of up or down regulated proteins in the resulting volcano plots. Only numbers between 0 and 100 will be accepted here.

5.- The parameter α-value presets the significance level for the t-Student analysis performed.

6.- The parameters P correction allows selecting the correction method for the p values calculated during the analysis.

The section *Column numbers* in Region 2 contains six parameters. Here, users provide the column numbers in the Data file from where UMSAP will get the information needed to perform the analysis of the module. All columns specified in this section must be present in the Data file. Users must be aware that Python starts counting from 0. Therefore, the number of the columns in the Data file starts from 0 and not from 1. The column numbers displayed in the list box in Region 3 after the Data file is selected can be directly used for the parameter values.

1.- The parameter Detected proteins allows users to specify the column in the Data file containing the protein identifiers found in the Data file. Only one integer number equal or greater than zero will be accepted here.

2.- The parameter Gene names allows users to specify the column in the Data file containing the gene names of the proteins found during the MS experiments. Only one integer number equal or greater than zero will be accepted here.

3.- The parameter Score allows to specify the column in the Data file containing the Score values. Only one integer number equal or greater than zero will be accepted here.

4.- The parameter Exclude proteins allows to specify several columns in the Data file. Proteins found in these columns will be excluded form the analysis. The module assumes that these columns contains numeric values and values greater than zero indicate that the respective protein must be eliminated from the analysis. Only integer numbers equal or greater than zero will be accepted here.

5.- The parameter Columns to extract allows to select multiple columns from the Data file in order to generate a shorter Data file. This parameter is optional. Only integer numbers equal or greater than zero will be accepted here. A range of columns can be specified as 4-10 with both numbers included.

6.- The parameter Results - Control experiments allows users to specify the columns in the Data file containing the results of the experiments. There are two ways to provide the information for this parameter. Users can load the values from a .txt file using the Load values button or use the Type values button to call a helper window, see **??**. Duplicate column numbers are not allowed here.

The helper window is divided in four Regions. Region 1 allows to define the number of conditions and relevant points analyzed, to define the kind of control experiment performed and to create the matrix in Region 2. Each text field in Region 2 should contain the column numbers containing the MS results for the given experiment. The values for the text fields should be positive integer numbers or a range of integers, e.g.

60–62 or NA for empty experiments. The column numbers can be seen in the list box in Region 3. Selected entries in the list box can be copied and then pasted to the text fields using the right mouse button or the Tools menu. Region 4 contains two buttons to Cancel or to Export the values to the window of the module.

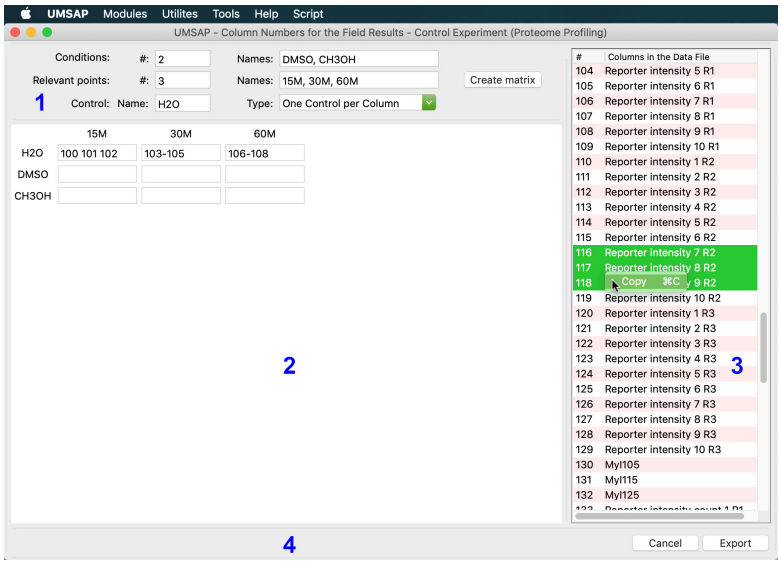

Figure 5.2: The Result - Control experiments helper window. This window allows users to specify the column numbers in the Data file containing the MS results for the selected conditions and relevant points.

The column numbers, the labels for conditions and relevant points and the information for the controls can also be loaded from a .txt file. The format of the file content is very simple. The first four lines give the values for the labels and control and the rest of the lines specify the column numbers with a comma (,) separating the values for a different condition - relevant point. The following is an example for two conditions, two relevant points and one control for each condition:

Control type : One Control per Row
Control name : MyControl
Condition names: DMSO, H2O
Relevant point names: 30min, 1D

105 115 125, 106 116 126, 101 111 121
130 131 132, 108 118 128, 103 113 123

Region 3 of the Proteome Profiling module main window contains a list box that will display the number and name of the columns found in the Data file. The list box is automatically filled when the Data file is selected. Selected columns in the list box can be directly added to any field in the section *Column numbers* in Region 2 of the interface using the right mouse button over the list box or the Tools menu.

Region 4 contains two buttons and the progress bar. The Help button leads to an online

tutorial while the Start analysis button will trigger the processing of the data file. The progress bar will give users a rough idea of the remaining processing time.

5.2.1 The Tools menu

The tools menu in the module window allows to copy the selected columns in the list box in Region 3 of the interface to the fields of section *Column numbers* in Region 2 of the interface. The list box in Region 3 of the interface can also be clear. In addition, through this menu users can create a .uscr file with the given options to the module before running the analysis. If something goes wrong during the analysis having the .uscr file means that users do not have to type the values of all the parameters again.

5.3 The analysis

First, UMSAP will check the validity of the user provided input. Then, the data file is processed as follow. All proteins found in the Exclude proteins columns are discarded. Proteins that were not identified in all conditions are discarded. Finally, all proteins with a Score value lower than the defined threshold are removed. The intensity values of the remaining proteins are normalized and then a median correction is applied to each experiment. With the resulting intensity values the fold change for each protein and experiment is calculated and two different analysis are performed.

The fold change is calculated as:

$$FC = ave(I_{C,RP})/ave(I_{Control}) \tag{5.1}$$

The first analysis is a t-test to determine if each experiment is significantly different to the corresponding control. The second analysis is a t-test, or ANOVA test if more than two conditions are studied, to determine if the values for the studied conditions are significantly different for each selected relevant point.

Finally, the corrected p values are calculated.

5.4 The output files

The Proteome Profiling module generates three files and a folder named Data_Steps. The folder Data_Steps contains a step by step account of all the calculations performed so users can check the accuracy of the calculation or perform further analysis. These file are plain text file with tab separated columns. The first line contains the name of the columns in the file.

The other three files created has extension .txt, .uscr and .protprof. The file with extension .txt contains the all the lines in the Data files but only the columns specified with the parameter Columns to extract in section Column number of Region 2 of the interface of the Proteome Profiling module. The file with extension .uscr contains all

the input given by the user so a new analysis may be performed without typing all the option values again, see **??**. The file with extension .protprof is the main output of the module. This file contains all the results and can be used to generate the graphical representation of the results.

5.5 Visualizing the output files

After creating the .protprof at the end of the analysis, the Proteome Profiling module will automatically load the file and create a windows to display the results, see Figure 5.3.This window is divided in four Regions.

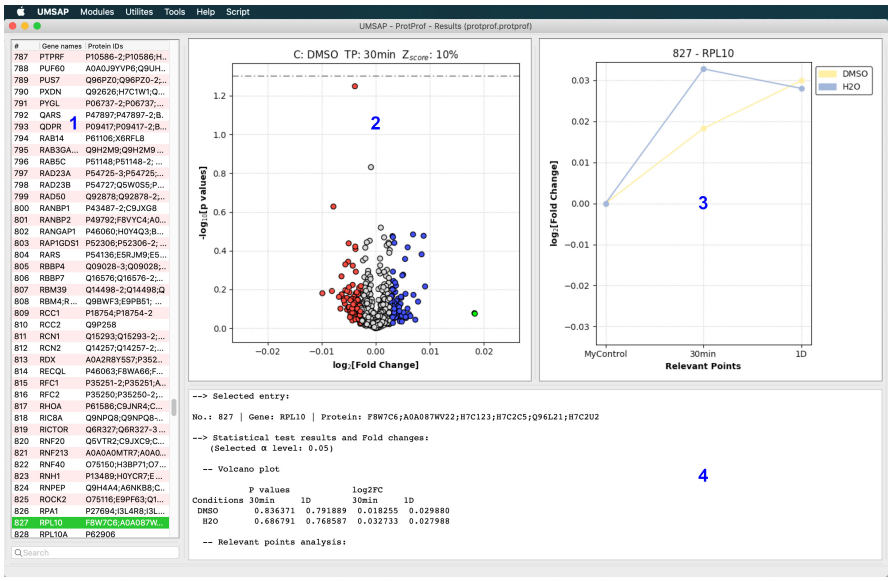

Figure 5.3: The Proteome Profiling analysis window. Users can performed here the analysis of the proteome profiling.

Region 1 contains a list of all protein IDs and gene names contained in the .protprof file being shown. The search box at the bottom allows to search for a protein in the list. Selecting a protein in the list will highlight the protein in Region 2 and display information about it in Regions 3 and 4.

Region 2 contains a volcano plot showing the results for the t-test comparing the condition (C), relevant point (RP) to the control. The volcano plot has a horizontal line indicating the significance level selected for the experiments. In addition, the points in the plot can be colored by Z-score allowing to quickly identified the top up (blue) or down (red) regulated proteins. In Figure 5.3, the top 10% up and down regulated proteins are colored. Selecting a protein in the plot will highlight the selected protein in Region 1 and display information about it in Regions 3 and 4. See subsection 5.5.1 for more options.

Region 3 contains a plot of $log_2[FC]$ vs Relevant point. The plot allows to see the behavior of the FC along the relevant points for each condition tested in the experiments.

See subsection 5.5.1 for more options.

Region 4 shows a summary of the results for a selected proteins. Proteins can be selected in the listbox in Region 1 or in the volcano plot of Region 2. The information includes a summary of the selected protein including the number of the protein in the listbox, the gene name and the protein id. Calculated p and $log_2[FC]$ values as well as average intensities and standard deviations.

5.5.1 The Tools menu

The Tools menu for the results window of the Proteome Profiling module allows to further customize the plots and to apply different filters to the protein list shown in the window.

Under the menu entry Volcano Plot, users can change the condition and relevant point shown in the volcano plot and the Z score value used to color the points in the plot. The complete state of the window can be reset or an image of the volcano plot can be created. If the condition or relevant point displayed is changed the Apply Filters menu entry allows to recalculate the filters for the current condition, relevant point displayed.

Under the menu entry Relevant Points, users can show all the proteins at once with different or the same colors. reset the state of the window or create an image of the Relevant Points graph.

The menu entry Corrected P values display the information in the .protprof file using the corrected p values.

5.5.1.1 Filters

The menu entry Filters allows to Add, Remove or Reset the filters applied to the protein list. The idea behind Filters is to identify proteins with a desired behavior and discard the rest of the proteins from the listbox in Region 1 and the plots in Regions 2 and 3. Filters are applied to the current Condition and Relevant point shown in the Volcano plot in Region 2. If the Condition or the Relevant point shown is changed, the new plot will show the proteins obtained after the filter was applied. This allows to follow the behavior of the filtered proteins in all Conditions and Relevant points. The menu entry Applied Filters in the Volcano plot submenu allows to recalculate the filters based on the new Conditions and Relevant point shown. Any number of filters can be applied. The applied filters are shown in the bottom left corner of the results window.

Filter can be removed in any given order using the menu entry Any in the Remove filter submenu. Additionally, the last applied filter can be removed with the menu entry Last Added or the shortcut Ctrl (Cmd) + Z.

Currently, the implemented filters are:

Z score

This menu entry allows to filter proteins by the Z score value of the Fold change.

Log2FC

This menu entry allows to filter proteins by the absolute value of the $log_2[FC]$.

P value

This menu entry allows to filter proteins by the P value calculated for the current Condition and Relevant point. The threshold α-value can be given in the 0 to 1 range or as $-log10$ value. Regular or corrected P values can be used in the filter.

One P value

This menu entry allows to filter proteins by P values, but here all Conditions and Relevant points are searched. Therefore, the returned list of proteins consists of proteins that pass the filters for at least one Condition - Relevant point pair.

Monotonic

This menu entry allows to filter proteins by the behavior of the $log_2[FC]$ along the Relevant points. The filter searches for proteins that have a monotonically increasing or decreasing (or both) behavior for at least one condition.

Divergent

This menu entry allows to filter proteins by the behavior of the $log_2[FC]$ along the Relevant points. In this case, the filter searches for proteins that have a monotonically increasing and decreasing behavior in at least two of the conditions tested.

Chapter 6

The Targeted Proteolysis module

The module Targeted Proteolysis is designed to post-process the mass spectrometry data acquired during the enzymatic proteolysis of a target protein by a single protease. In a typical experimental setup both the protease and the target protein are mixed together under various experimental conditions and the peptides generated during the proteolysis are identified by mass spectrometry. It is expected that several control experiments and several replicates of the tested experimental conditions are performed. The main objective of the module is to identify the peptides with intensity values that are significantly different in the control experiments and the replicates of the various experimental condition tested at the chosen significance level.

6.1 Definitions

Before explaining in detail the interface and how does the module works, lets make clear the meaning of the term Filtered peptide in the context of the Targeted proteolysis module:

- *Filtered peptide*: a relevant peptide with a significantly different behavior in the control and a given experiment at the chosen significance level.

6.2 The input data files

The Targeted Proteolysis module requires a data file containing the detected peptides and a sequence file containing the amino acid sequence of the recombinant protein used in the study. Both files must follow the guidelines specified in Section 3.1. In short, the data file must have a tabular format with tab separated columns and the name of the columns are expected as first row. The sequence file is expected to contain only one sequence and to be FASTA formatted with or without the header line. All columns given as input in the section *Column numbers* in Region 2 of the interface must be present in the data file. Optionally, another sequence file with the sequence of the native target protein and a pdb file may be specified.

6.3 The interface

The window of the Targeted Proteolysis module is divided in four regions (Figure 6.1).

Region 1 contains four buttons allowing users to quickly delete all provided input and start a new analysis. The Clear all button will delete all user provided input and will empty the list box in Region 3. The Clear files button will delete the path to all user provided files and will empty the list box in Region 3. The Clear values button will delete all user provided input for the section Values in Region 2. Finally, the Clear columns button will delete all user provided column numbers.

Region 2 contains the fields where users provide the information needed in order to perform the post-processing of the input data file. The section *Files* in Region 2 will provide the path to the input data and output files. It contains six buttons.

1.- The Data file button allows users to browse the file system and select a data file. Only .txt files can be selected here. Once the data file is selected, the name of the columns in the file will be shown in the list box in Region 3. If the path to the data file is typed in, the display of the name of the columns in Region 3 can be trigger by pressing the Enter key in the keyboard while the Data file entry box has the focus of the keyboard.

2.- The Sequence (rec) button allows users to browse the file system and select the file containing the sequence of the recombinant protein under study. Only .txt or .fasta files can be selected here. Alternatively, a UNIPROT code may be given in this field.

3.- The Sequence (nat) button allows to select the file containing the sequence of the native protein. Only .txt or .fasta files can be selected here. Alternatively, a UNIPROT code may be given in this field. The sequence of the native protein is an optional field. A value of NA means that no native sequence file will be given.

4.- The PDB file button allows to browse the file system to select a .pdb file. The .pdb file should contain the structure of the Target protein. This .pdb file will be used to map the cleavages detected in the MS experiments to the structure of the protein. This is an optional field. A value of NA means that no .pdb file will be given. See page 43 for more details.

See Section 3.1 for more details about the data files.

5.- The Output folder button allows users to browse the file system and select the location of the folder that will contain the output. By default, UMSAP will create a TarProt folder inside the selected Output folder to save all the generated results. If only the name of the output folder is given, the output folder will be created in the same folder containing the data file. If this field is left empty, then the TarProt folder will be created in the same folder containing the data file. If the selected output folder already contains a TarProt folder, then the current date and time to the seconds will be added to the name in order to avoid overwriting the files from previous analyses.

6.- The Output name button does nothing but the text box to its right allows users to specify the name of the files that will be generated during the analysis. If this field is left empty, then the name tarprot will be used for the output files.

The section *Values* in Region 2 contains eight parameters. Here, users provide information

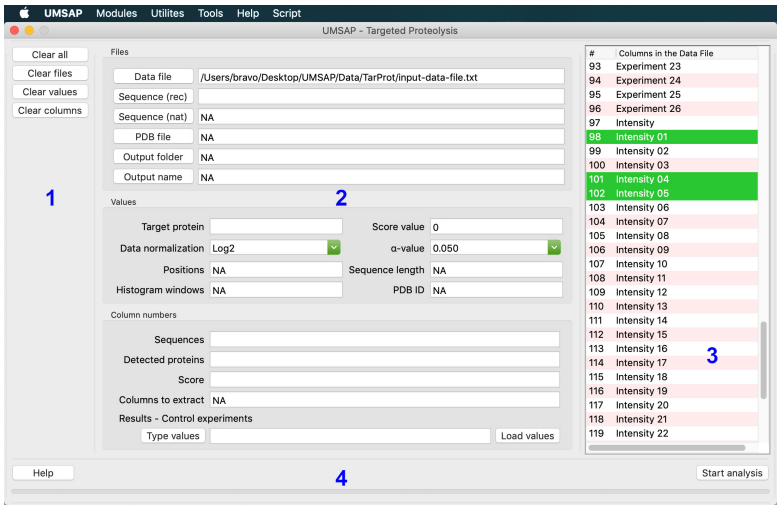

Figure 6.1: The Targeted Proteolysis window. This window allows users to performed the analysis of the results obtained during an enzymatic proteolysis experiment. Optional parameters default to NA when the window is created. The rest of the parameters must be provided by the user.

about the target protein, how the data file should be processed and which optional analysis will be performed.

1.- The parameter Target protein allows users to specify which of the proteins detected in the MS experiments was used as substrate during the enzymatic proteolysis. Users may type here any unique protein identifier present in the data file. The search for the target protein is case sensitive, meaning that eFeB is not the same as efeb.

2.- The parameter Score value allows users to define a threshold value above which the detected peptides will be considered as relevant. The Score value is an indicator of how reliable was the detection of the peptide during the MS experiments. The value given to UMSAP depends on the program generating the data file. Only one real number equal or greater than zero will be accepted as a valid input here. A value of zero means all detected peptides belonging to the target protein will be treated as relevant sequences.

3.- The parameter Data normalization allows selecting the normalization procedure to be performed before running the analysis of the data in the data file. Currently, only a $\log_2$ normalization is possible but this will be expanded soon to include quantile, variance stabilization and local regression normalization, among other methods.

4.- The parameter α-value sets the significance level for the ANCOVA test used to identify peptides with a behavior in the experiments significantly different to the control. See page 44 for more details.

5.- The parameter Positions allows users to define the number of positions to be considered during the amino acid (AA) distribution calculation, see subsection 7.2.1 for more details. Only one integer number greater than zero will be accepted here. A value of NA means that the AA distribution calculation will not be performed.

6.- The parameter Sequence length allows users to define the number of residues per line

in the short version of the sequence alignment files, see subsection 7.2.6 for more details. Only one integer number greater than zero will be accepted here. A value of NA means that the sequence alignment files will not be generated.

7.- The parameter Histogram windows allows users to define the size of the windows for the Histogram analysis, see subsection 7.2.5 for more details. Only integer numbers equal or greater than zero will be accepted here. In addition, the values must be organized from smaller to bigger values. Users may specify a fix histogram window size by given just one integer number greater than zero. In this case the histogram will have even spaced windows with the width specified by Histogram windows. If more than one number is provided here, then windows with the customs width will be created. For example, the input 50 100 will create only one window including cleavage sites between residues 50 to 99. The input 150 100 150 will create three windows including cleavages sites between residues 1 to 49, 50 to 99 and 100 to 149. Duplicate values are not allowed. A value of NA means that no histograms will be created.

8.- The parameter PDB ID allows users to specify the chain or segment in the pdb file to use when mapping the cleavages detected in the MS experiments to the structure of the target protein. There are two possibilities. When a PDB file is provided with the button PDB file in the section *Files* in Region 2 of the interface, then the expected value for PDB ID is only the chain or segment ID found in the pdb file that will be used for mapping the detected cleavages. When the pdb file should be downloaded from the PDB database, then a PDB code and the chain or segment ID is expected here. The format in this case is Code:Chain or Code:SegmentID, for example, 2y4f:A or 2y4f:PROA.

The section *Column numbers* in Region 2 contains five parameters. Here, users provide the column numbers in the data file from where UMSAP will get the information needed to perform the analysis of the enzymatic proteolysis. All columns specified in this section must be present in the data file. Users must be aware that Python starts counting from 0. Therefore, the number of the columns in the data file starts from 0 and not from 1. The column numbers displayed in the list box in Region 3 after the data file is selected can be directly used for the parameter values.

1.- The parameter Sequences allows users to specify the column in the data file containing the sequences of the peptides identified in the MS experiments. Only one integer number equal or greater than zero will be accepted here.

2.- The parameter Detected proteins allows users to specify the column in the data file containing the unique protein identifier for the proteins detected in the MS experiments. It is in this column where the program will look for the Target protein value given in the section *Values* in Region 2 of the interface. Only one integer number equal or greater than zero will be accepted here.

3.- The parameter Score allows users to specify the column in the data file containing the Score values. It is in this column where the program will look for the values to be compared against the Score threshold given in section *Values* of Region 2 of the interface.

4.- The parameter Columns to extract allows users to specify which columns in the data file will be copy to the shorter versions of the data file, see page 46 for more details. A range of columns may be specified as 4–10. Any number of columns may be specified

here. Only integers numbers equal or greater than zero will be accepted. A value of NA means no shorter version of the data files will be created.

5.- The parameter Results - Control experiments allows users to specify the columns in the data file containing the results of the control and enzymatic digestion experiments. There are three ways to provide the information for this parameter. Users can directly type the column numbers corresponding to the control and digestion experiments, load the values from a .txt file using the Load values button or use the Type values button to call a helper window. Independently of the chosen method the expected input here is a semicolon (;) separated list of column numbers. The first group of numbers define the columns containing the results for the control experiments followed by the definition of experiment 1 to n. For example, the following input define a control experiment (98–105), two experiments with three replicates each and a third experiment with four replicates: 98–105; 109–111; 112 113 114; 115–117 120. Here, any number of columns may be specified. Only integers numbers equal or greater than zero will be accepted. Duplicate values are not allowed.

Region 3 contains a list box that will display the number and name of the columns found in the data file. The list box is automatically filled when the data file is selected. Selected columns in the list box can be directly added to any field in the section *Columns in the input file* in Region 2 of the interface using the right mouse button over the list box or the Tools menu (Figure 6.1).

Region 4 contains two buttons and the progress bar. The Help button leads to an online tutorial while the Start analysis button will trigger the processing of the data file. The progress bar will give users a rough idea of the remaining processing time.

6.3.1 The Tools menu

The tools menu in the module window allows to copy the selected columns in the list box in Region 3 of the interface to the fields of section *Column numbers* in Region 2 of the interface. The list box in Region 3 of the interface can also be clear. In addition, through this menu users can create a .uscr file with the given options to the module before running the analysis. If something goes wrong during the analysis having the .uscr file means that users do not have to type the values of all the parameters again.

6.4 The analysis

First, UMSAP will check the validity of the user provided input. Then, the data file is processed as follow. All rows in the data file containing peptides that do not belong to the target protein are removed. Then, all rows containing peptides from the target protein but with Scores values lower than the user defined Score threshold are removed. These steps leave only relevant peptides, this means peptides with a Score value higher than the user defined threshold that belong to the target protein. For each one of these relevant peptides the ANCOVA test is performed.

The ANCOVA test, done to identify relevant peptides with different behavior in the

control and a given experiment, is performed in three steps. First, the intensity values for the replicates in the control and in a given experiment are normalized and organized in two data sets as indicated in Figure 6.2. Each data set consist of two points. For the control data set the intensity values in the replicates of the control experiment are allocated to both points. For the experiment data set the intensity of the replicates in the control experiment are allocated to the first point and the intensity values of the replicates in the given experiment are allocated to the second point. The second step is to find the slope of the straight line best fitting each data set. The third step is to test the homogeneity of the regression slopes. Peptides that fail this test are included in the list of filtered peptides (FP) because the slopes of the straight lines fitting the data sets are significantly different at the chosen significance level. The fact that the slopes are different implies that the peptide is found in an increased concentration in the given experiment than in the control experiment. Peptides that past this test are not included in the list of FP for the given experiment.

After all relevant peptides in all experiments have been analyzed, the output file with extension .tarprot is written. Based on the .tarprot file just created UMSAP calculates the number of cleavages per residue (subsection 7.2.2) and generates the input file with extension .uscr (subsection 7.3.2) and a copy of the FP list (subsection 7.2.4). A folder Data_Steps is created containing a step by step summary of the data processing performed by UMSAP. Finally, the requested optional analyses are performed as described in the corresponding sections of Chapter 7. The optional analyses are: amino acid distribution (subsection 7.2.1), sequence alignments (subsection 7.2.6), histograms of detected cleavages (subsection 7.2.5), mapping of detected cleavages to a .pdb file (subsection 7.2.3) and short data files (subsection 7.3.5).

If the sequence of the native protein is given the module performs a sequence alignment between the native and recombinant sequences. The alignment allows UMSAP to translate the results obtained with the residue numbers of the recombinant protein to the residue numbers of the native protein. This is done to facilitate future comparison of results between different recombinant proteins of the same native protein. However, when analyzing the results of the alignment the module assumes that the recombinant and native sequences differs only in the N and C-terminal tags while the sequence between the tags is identical. If this is not the case, e.g. there are point mutations or insertion/deletion in the sequence of the recombinant protein no native sequence file should be given to UMSAP. This restriction will be eliminated in future versions of UMSAP. Mapping of the cleavage sites to the .pdb file involves also a sequence alignment between the recombinant protein and the sequence found in the .pdb file. However, the mapping of the detected cleavage sites does not have the restriction discussed before.

6.5 The output files

All the output generated by the Targeted Proteolysis module will be contained in a TarProt folder created inside the selected Output folder. If the Output folder field in the section *Files* in Region 2 of the interface is left empty, then the TarProt folder will be created in the directory containing the data file. If the selected Output folder already

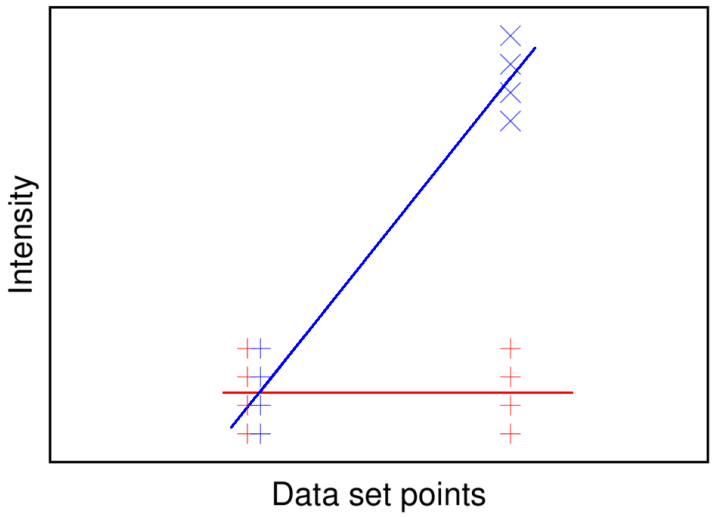

Figure 6.2: Data organization prior to the ANCOVA test. Two data sets with two points each are created, one data set for the control (red) and one for a given experiment (blue). The intensity data in the replicates of the control is used in both data points for the control (+) and in the first data point of the given experiment. The intensity data in the replicates of the given experiment is used for the second point of the data set for the given experiment (x). After this, the best fitting line for each data set is found and the slopes of the lines are compared in a test for homogeneity of the regression slopes.

contains a TarProt folder, then the current date and time to the seconds will be added to the name in order to avoid overwriting files from previous analyses. By default the TarProt folder will contain four files with extensions .tarprot, .cutprop, .filtpept and uscr and a Data_Steps folder. The name of these files is provided with the Output name field in the section *Files* of Region 2 of the interface. Depending on the user provided input extra folders and files will be created inside the TarProt folder (Figure 6.3). For the rest of this chapter we will assume that the user provided name for the Output folder was *t*, the Output name was myTest, the target protein was *efeB* and all optional analyses were performed.

If the parameter Columns to extract (see page 43) is different than NA, then shorter versions of the data file will be created and saved in a folder Data inside the main folder t. The folder t/TarProt/Data will contain three files as described in subsection 7.3.5.

If the parameter Histograms window (see page 43) is different than NA, then histograms of the detected cleavage sites will be created as described in subsection 7.2.5. The folder t/TarProt/Histograms will contain two files with the histograms.

If a PDB file was selected and the parameter PDB ID (see page 43) was set, then the detected cleavage sites are mapped to the structure in the .pdb file, as described in subsection 7.2.3 and the resulting .pdb files are saved in the folder t/TarProt/PDB. The information regarding the cleavages is saved to the beta field of the .pdb files. The .pdb files can be visualized with VMD, PyMol or Chimera.

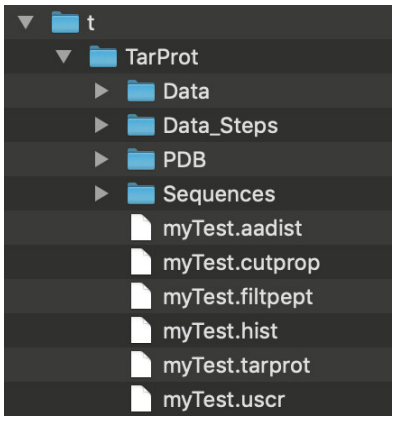

Figure 6.3: The structure of the Output folder from the Targeted Proteolysis module. Folders Data, Histograms, PDB and Sequences are optional and will be created only if requested. The same is true for the .aadist file.

If the parameter Sequence length (see page 42) is different than NA, then sequence alignment files will be created as described in subsection 7.2.6. The folder t/TarProt/Sequences will contains several sequence alignment files. These are plain text files that can be viewed with any text editor.

If the parameter Positions (see page 42) is different than NA, then an AA distribution analysis is performed as described in subsection 7.2.1. The file t/TarProt/myTest.aadist contains the AA distribution analysis.

The file myTest.cutprop contains the results of the number of cleavages per residue as discussed in subsection 7.2.2 while the file myTest.filtpept contains the list of FP as described in subsection 7.2.4. The file myTest.uscr is the input file that can be used to quickly reanalyzed the data file without having to type in all the information needed by the module, see subsection 7.3.2 for more details.

The main output of the Targeted Proteolysis module is the myTest.tarprot file. This file contains a summary of the user provided input and a table specifying the FP for each experiment. In addition, this file contains the information necessary to visualize the fragments generated during the experiments and to run all optional analyses individually without having to run the entire module.

6.6 Visualizing the output files

After the analysis of the Targeted Proteolysis module is finished new windows will appear showing a graphical representation of the results contained in the .tarprot and .cutprop files. Depending on the user input, the AA distribution analysis and the histogram for the cleavage sites in the recombinant protein will also be shown. More details about the graphical representation of the .aadist, .hist and .cutprop files is given in subsection 7.2.1, subsection 7.2.5 and subsection 7.2.2 respectively. The data contained in the shorter versions of the input file and the sequence alignment files can be easily viewed using any

text editor.

The window displaying the results in the .tarprot file is divided in four regions (Figure 6.4).

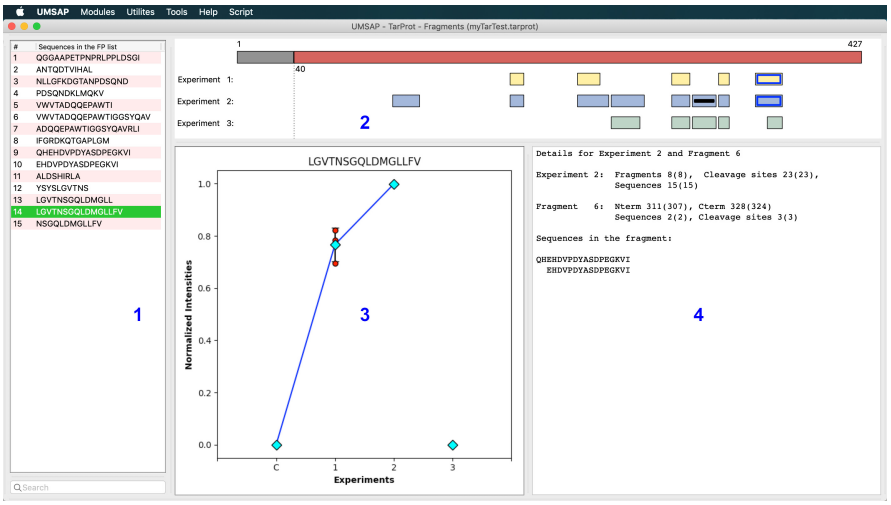

Figure 6.4: The Fragment analysis window. Users can performed here the analysis of the fragments obtained in the enzymatic proteolysis experiments.

Region 1 contains a list box displaying the complete list of FP. Selecting one sequence in the list box will highlight the fragments in Region 2 containing the selected peptide with a blue thicker border. In addition, a plot of Normalized intensities vs experiment number for the selected peptide will be displayed in Region 3. Below the list box there is a search box that allows to search for a sequence in the FP list. If the typed sequence exactly matches one peptide of the FP list, then the sequence will be selected in the list box, the fragments containing the sequence in Region 2 will be highlighted and the plot in Region 3 will be updated. If the typed sequence is found in more than one peptide of the FP list, the number and the sequence of the peptides containing the typed sequence will be shown.

Region 2 displays all the fragments generated in the enzymatic proteolysis experiment. The first line represent the full sequence of the recombinant protein. The red color represents residues from the native protein while the gray color represents residues in the recombinant protein that do not belong to the native protein sequence. The other rows represent the different experiments performed. The experiments are organized in the same way as specified with the parameter Results - Control experiments in Region 2 of the interface of the Targeted Proteolysis module, see page 44. In addition, vertical dotted lines are drawn from the recombinant protein sequence to the bottom of Region 2 in order for users to quickly identified if fragments contain only residues from the native protein or not. Selecting one fragment will highlight the fragment with a black line and will give detailed information about the fragment in Region 4.

Region 3 will display a plot of Normalized intensities vs experiment number. The plot will display the data for a single FP selected in the list box in Region 1. The values in the plot are obtained by using feature rescaling to bring all intensity values for the selected peptide into the range 0 to 1. Discarded replicates are not shown and if intensity

values were normalized prior to the Targeted Proteolysis analysis then the normalized values are used for the feature rescaling procedure. The values for replicates are shown as red circles. The average for a given experiment is shown as bigger cyan diamonds and the standard deviation is used for the bars. The blue lines only connect the control experiment and experiments showing intensity values significantly different at the chosen significance level.

Region 4 will display detailed information about a selected fragment in Region 2. In this Region regular numbers refer to the recombinant protein sequence while number between parenthesis refers to the Native sequence. The Experiment section in the text contain information about the experiment in which the fragment was identified. The information includes the total number of fragments, the total number of sequences and the total number of unique cleavage sites identified in the experiment. The Fragment section in the text gives similar information about the selected fragment and also includes the first and last residue numbers in the fragment. The rest of the lines show a sequence alignment of all the peptides in the fragment.

6.6.1 The Tools menu

The Tools menu in the window allows to extend the functionality of the windows. Through this menu users can create a copy of the FP list, save an image of the fragments or the plot and reset the state of the window. Most of these functions can be also accessed by clicking the right button of the mouse (Figure 6.4).

Chapter 7

Utilities

Currently, there are 15 utilities. Users can access the utilities in two ways. From the main interface (Figure 3.1) users can select Utilities in the list to the right and a new window will appear with a complete list of available utilities (Figure 7.1). The alternative option is to directly select the desired utility from the menu entry, Utilities. The second approach is faster since does not require to use the Utilities window (Figure 7.1).

The utilities are organized in General Utilities and utilities that are specific for a given module. The following sections describe each one the implemented utilities.

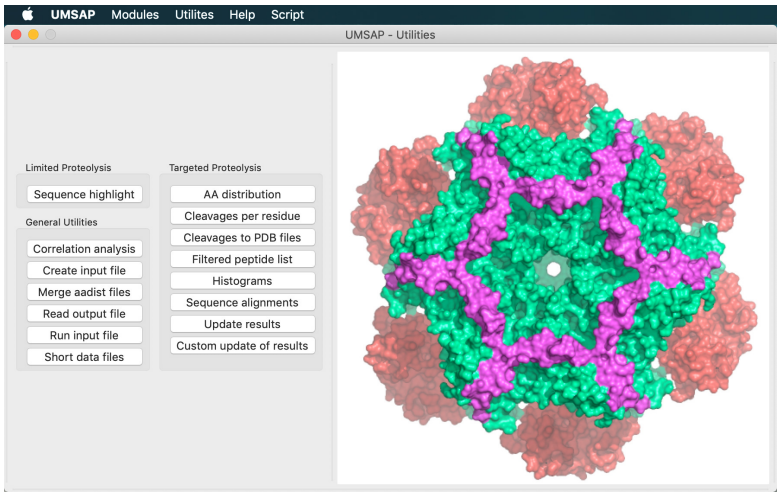

Figure 7.1: The Utilities window. From this window users can access all the available utilities.

7.1 Limited Proteolysis utilities

7.1.1 Sequence highlight

The Sequence Highlight utility allows user to highlight on the sequence of the Target protein the peptides detected in each gel lane/band considered in the .limprot file. The sequences with the highlighted peptides are saved in a .pdf file. How to generate a .limprot file is discussed in Chapter 4.

The interface

The window of the Sequence Highlight utility is divided in three regions.

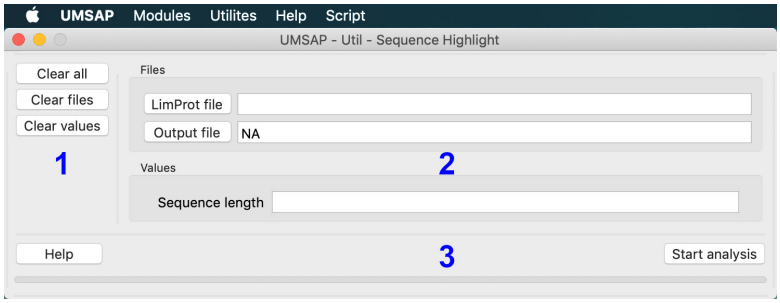

Figure 7.2: The Sequence Highlight window. This window allows to generate a .pdf file showing the location of the peptides detected in the investigated gel lane/band on the sequence of the Target protein.

Region 1 contains three buttons allowing users to quickly delete all provided input to generate a new .pdf file. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to create the .pdf file with the highlighted sequences. The Limprot file button allows users to browse the file system to select the .limprot file that will be used for the generation of the .pdf file. Only one .limprot file can be provided here. The Output file button allows users to browse the file system to select the location and name of the .pdf file. If left empty, then the .pdf file, resulting from the analysis, will be saved in the same directory containing the .limprot file and will have the same name as the .limprot file. If the folder containing the selected .limprot file already contains a .pdf file with the same name as the selected .limprot file, then UMSAP will add the current date and time to the seconds to the end of the .pdf file name in order to avoid overwriting the older .pdf file without explicit user permission.

The Sequence length parameter allows user to define the maximum number of residues per line to be used during the creation of the .pdf file. The value here must be an integer number greater than zero.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the generation of the .pdf file. The progress bar will give a rough estimate of the remaining time for

completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. After this, for each gel lane/band analyzed in the .limprot file, UMSAP will print the sequence of the Target protein to the .pdf file with the peptides found in the gel lane/band highlighted.

The output

The output is a .pdf file containing a page for each gel lane/band found in the .limprot file. As described before, each page contains the sequence of the Target protein with the detected peptides highlighted in red. In addition, the residue number of the beginning and ending of the highlighted fragments are also given. If the sequence of the native protein was provided when creating the .limprot file, then the information is given for the recombinant and native protein.

The Tools menu

This utility does not have a Tool menu.

7.2 Targeted Proteolysis utilities

7.2.1 AA Distribution

The AA Distribution utility allows user to calculate the AA distribution around the detected cleavage sites using a list of FP (see page 40). The list of FP is automatically generated from a .tarprot file. How to generate a .tarprot file is discussed in Chapter 6. The list of FP is a non redundant list.

The interface

The window of the AA Distribution utility is divided in three regions (Figure 7.3).

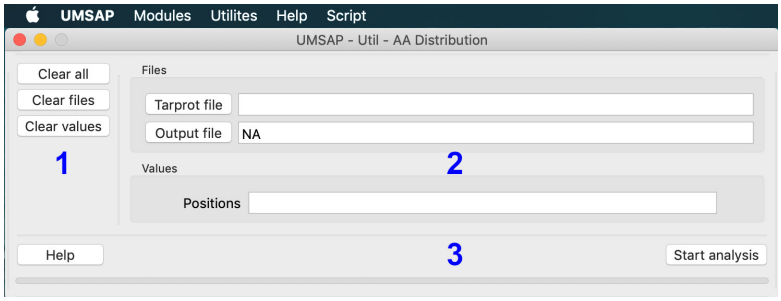

Figure 7.3: The AA Distribution window. This window allows to obtain the AA distribution around the detected cleavage sites from a .tarprot file.

Region 1 contains three buttons allowing users to quickly delete all provided input and start a new distribution analysis. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to

perform the AA distribution calculation. The Tarprot file button allows users to browse the file system to select the .tarprot file that will be used for the analysis. Only one .tarprot file can be provided here. The Output file button allows users to browse the file system to select the location and name of the Output file. If left empty, then the .aadist file, resulting from the analysis, will be saved in the same directory containing the .tarprot file and will have the same name as the .tarprot file. If the folder containing the selected .tarprot file already contains a .aadist file with the same name as the selected .tarprot file, then UMSAP will add the current date and time to the seconds to the end of the .aadist file in order to avoid overwriting the older .aadist file without explicit user permission.

The parameter Positions indicates the number of positions around the cleavage sites to be analyzed. The value here must be an integer number greater than zero.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. After this, a list of FP is created using the .tarprot file. For each FP, the sequence around the N and C terminal ends of the peptide is analyzed up to the user provided number of Positions. For the N terminus of the peptide the identity of residues in positions Pn to $P1$ is inferred from the sequence of the recombinant protein contained in the .tarprot file. The same is done for positions $P1'$ to Pn' at the C terminus of the peptide. If the N or C terminus of a peptide is the first or last residue of the recombinant protein under study the N or C terminus is excluded from the analysis. For each position the number of times that each AA appears at a given position are counted. Finally, the absolute numbers of AA appearances for each position are converted to percent taking the total for each position as the sum of all counted AA in the position.

In addition, UMSAP tests whether the obtained AA distribution is significantly different to the expected AA distribution from the proteolysis of the Target protein by a totally non-selective protease. The first step is to generate an AA distribution with the same number of positions defined by the user with the Position parameter. This distribution is generated assuming that all peptidic bonds in the recombinant protein may be cleaved by the protease with equal probability and that all peptidic bonds will be cleaved. Here, we are also assuming that all products of cleaving all peptidic bonds will be detected in the MS experiment. Then, UMSAP compares each position in both distributions using a χ^2 test with the significance level found in the .tarprot file. In order to be able to perform the χ^2 test, AAs are pooled together in the same groups as described for the color code used in the output.

The output

The output from the AA distribution calculation is a file with .aadist extension. The file will be automatically loaded and a graphical representation of the results will be shown (Figure 7.4). There are two graphical representations. The first representation shows

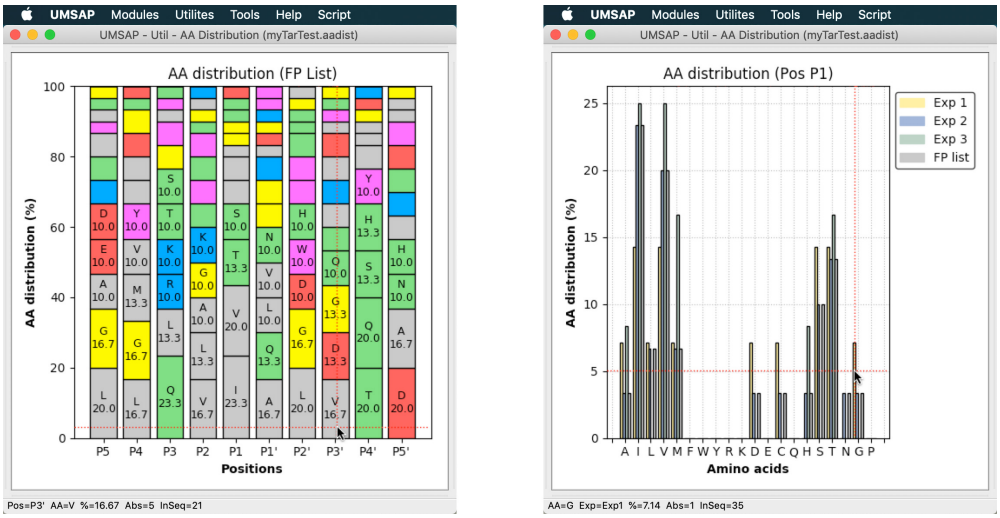

Figure 7.4: The AA Distribution analysis window. This window allows to visualize the results contained in a .aadist file.

a bar graph of the AA distribution in which each bar represents a position. AAs are color coded with positively charged AAs (R and K) in blue, negatively charged AAs (D and E) in red, polar AAs (S, T, N, H, C and Q) in green, non-polar AAs (A, V, I, L and M) in gray, aromatic AAs (F, Y and W) in pink and Gly and Pro in yellow. AAs with an occurrence higher than 10 % are labeled with the one letter code for the AA and the percentage value. For example, in Figure 7.4 the value of 16.7 % obtained for A in position $P1'$ means that A was found in position $P1'$ in the 16.7 % of the total cleavage sites detected.

The results of the χ^2 test are given in the color of the name of the position. A green color represents that the obtained distribution in the position is significantly different to a no selectivity distribution at the level of significance found in the .tarprot file. A red color represents that the distributions are not significantly different at the level of significance found in the .tarprot file. Finally, a black color indicates that the number of expected values below 5 was higher than the 20 % threshold recommended by Yates et al. and the test was not performed (**Yates1999**).

If the mouse pointer is placed on top of the bars, then information related to the bar and the AA will be shown in the status bar at the bottom of the window. The information includes the Position (Pos), the amino acid (AA), how many times does the AA appears in the position as a percent of the total AA count for the given position (%) and the absolute number (Abs) and how many times does the AA appears in the sequence of the recombinant protein (InSeq).

The second representation allows to compare the AA distribution in one position across all experiments. This is also a bar representation in which each bar represents an experiment and each position an AA. Placing the mouse pointer over a bar shows information about it. The information includes the AA (AA), the experiment (Exp), how many times does the AA appears in the position for the given experiment as a percent of the total AA count found at the position for the given experiment (%) and

the absolute number (Abs) and how many times does the AA appears in the sequence of the recombinant protein (InSeq).

The Tools menu

By default, the AA distribution for all AA in the FP list will be shown when the .aadist file is loaded. The Tools menu in the window allows user to change the displayed experiment or to select the position for which results in the experiments should be compared. In addition, users may select to save a figure of the plot and to reset the view.

7.2.2 Cleavages per Residue

The Cleavages per residue utility calculates the absolute number of cleavages detected in the MS experiments for each residue in the recombinant protein under study. The peptides used to identify the cleavage sites are the FP contained in the .tarprot file used as input for the calculation (see page 40). How to generate the .tarprot file is discussed in Chapter 6. The FP list is a non redundant list.

The interface

The Cleavages per residue utility does not have a window since there are no options to specify. When the utility is selected users will be asked to select a .tarprot file and then users must select the output file. That is all.

The analysis

First, UMSAP will check the validity of the user provided input. After this the list of FP will be generated from the .tarprot file. Then, UMSAP will count how many times each residue in the protein under study appears at the C terminus of a FP or at the $N-1$ position of a FP (N is the N terminus of the FP). The cleavages per residue value for the first and last residue of the protein under study is of course zero. This is done for every experiment in the .tarprot file and also taking into account the results for all experiments. Finally, UMSAP will take the absolute number of cleavages per residue and will normalize the values to bring them in the 0 to 1 range. Both the absolute and normalized values are written to the output file. After the analysis is done the results will be automatically loaded and displayed in a new window (Figure 7.5).

The output

The output file from the Cleavages per residue utility will be shown as a simple number of cleavages vs residue number plot (Figure 7.5). Residues with cleavages per residue higher than one third of the maximum number of cleavages per residue will be highlighted with an asterisk (*). Placing the mouse pointer inside the plot will display the residue number and the number of cleavages in the status bar at the bottom of the window. When two data sets are plotted simultaneously, the number of cleavages are given in the same order shown by the legend in the window. The gray vertical lines enclose the native residues.

The Tools menu

The window shows by default the absolute number of cleavages considering all FP for the recombinant protein. The Tools menu allows users to change this. Users may select to

plot the results for a particular experiment or to compare two experiments. In addition, only the native sequence could be plotted or the normalized cleavages per residue values may be shown. An image of the plot can be created using the Save Plot Image entry in the Tools menu. The Reset View option restores the default appearance of the window.

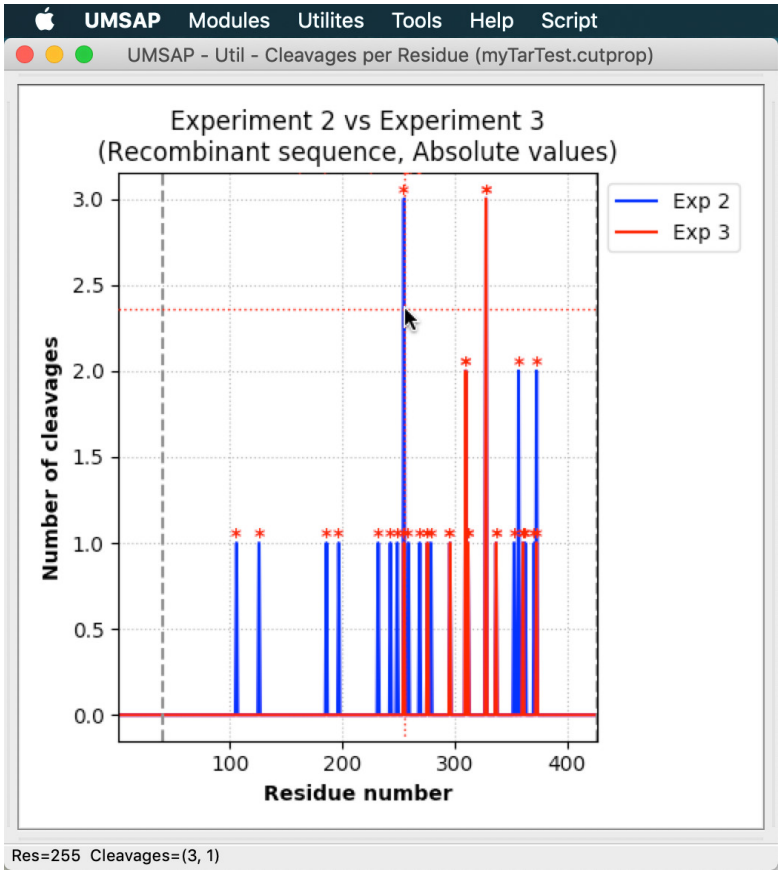

Figure 7.5: The Cleavages per Residue analysis window. This window allows to visualize the results contained in a .cutprop file.

7.2.3 Cleavages to PDB Files

The Cleavages to PDB Files utility maps the number of cleavages per residue found in a .tarprot file to a .pdb file containing the structure of the Target protein. The peptides used to identify the cleavage sites are the FP contained in the .tarprot file used as input for the calculation (see page 40). How to generate the .tarprot file is discussed in Chapter 6. The FP list is a non redundant list.

The interface

The Cleavages to PDB Files window is divided in three regions (Figure 7.6).

Region 1 contains three buttons allowing users to quickly delete all provided input and start a new distribution analysis. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear

values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to perform the mapping of the number of cleavages. The Tarprot file button allows users to browse the file system to select the .tarprot file that will be use for the analysis. Only one .tarprot file can be provided here. The PDB file allows user to browse the file system to select a .pdb file. The .pdb file will contain the structure of the Target protein. This field can be left empty if the PDB file is to be downloaded from the PDB data base. The Output folder button allows users to browse the file system to select the location of the resulting PDB folder containing the results. If left empty, then the PDB folder, resulting from the analysis, will be saved in the same directory containing the .tarprot file. If the Output folder option is left empty and the folder containing the selected .tarprot file already contains a PDB folder, then UMSAP will add the current date and time to the seconds to the end of the folder name in order to avoid overwriting the older PDB folder without explicit user permission.

The PDB ID field allows users to specify the chain or the segment id in the .pdb file that should be used for the mapping. Alternatively, if the PDB file field is left empty a code from the PDB database plus the chain or segment id may be given here. In this case, the pdb file will be directly downloaded from the PDB data base. The expected syntax in this case is code:chain or code:segment for example, 2f4y:A or 2f4y:PROA.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

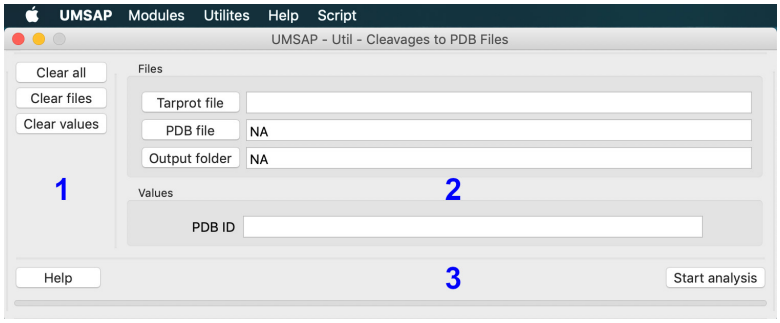

Figure 7.6: The Cleavages to PDB Files window. This window allows to map the detected number of cleavages per residue to a .pdb file containing the structure of the Target protein. The number of cleavages are mapped to the beta field in the .pdb.

The analysis

First, UMSAP will check the validity of the user provided input. Then, a temporal .cutprop file will be created. Details about the .cutprop files can be found in subsection 7.2.2. After this, the sequence from the .pdb file is extracted and aligned with the sequence of the recombinant protein found in the .tarprot file. Finally, the number of cleavages found in the .cutprop file are mapped to the corresponding residues in the .pdb file.

The output

The output from this utility is a series of .pdb files that will be saved in a PDB folder. Each file contains the number of cleavages mapped to the beta field of the corresponding residue in the .pdb structure. The results for each experiment and the FP list are mapped to individual files. The mapped values can be visualized by opening the .pdb files with VMD, PyMol or Chimera and coloring the structure by beta factors.

7.2.4 Filtered Peptide List

The Filtered Peptide List utility allows users to read a .tarprot file and to create a .filtpept file containing the FP list. How to generate the .tarprot file is discussed in Chapter 6. The FP list is a non redundant list of all peptides identified during the Targeted Proteolysis analysis.

The interface

The Filtered Peptide List utility does not have a window since there are no options to specify. When the utility is selected users will be asked to select a .tarprot file and then users must select the output file. That is all.

The analysis

First, UMSAP will check the validity of the user provided input. Then the .tarprot file will be read in and all FP will be saved in the output file.

The output

The .filtpept file has a tabular format in which the first row contains the name of the columns and columns are tab separated. It is a plain text file that can be read with any text editor. The file list all filtered peptides indicating the first and last residue number of the peptides and in which experiments were the peptide detected.

7.2.5 Histograms

The Histograms utility allows to create histograms of the identified cleavage sites using the residue numbers of the Target protein as the definition of the windows in the histograms. Histograms are created from a .tarprot file. How to generate the .tarprot file is discussed in Chapter 6. Only FP are used to create the histograms, see page 40. The list of FP is a non redundant list.

The interface

The Histograms window is divided in three regions (Figure 7.7).

Region 1 contains three buttons allowing users to quickly delete all provided input and start a new calculation. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to create the histograms. The Tarprot file button allows users to browse the file system to select the .tarprot file that will be used for the analysis. Only one .tarprot file can be selected here. The Output file button allows users to browse the file system to select the

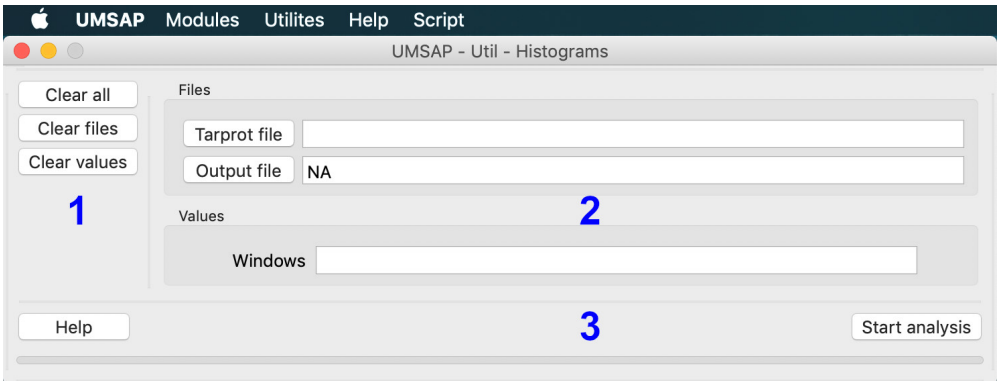

Figure 7.7: The Histograms window. This window allows to create histograms of the identified cleavage sites using the residue numbers of the Target protein as the definition of the windows in the histograms.

location of the .hist file to be created. If no Output file is selected the .hist file will be created in the same directory as the selected .tarprot file. If an older .hist file exists in the selected folder, UMSAP will append the date and time to the seconds to the end of the .hist file name in order to avoid overwriting the old .hist file.

The parameter Windows lengths allows to define the length of the windows in the histograms. Values here are expected to be integers greater than zero. A single value will results in equally spaced windows covering the entire length of the recombinant protein under study. Several values will result in custom sized windows. This may be useful if users want to define windows matching a structure related property of the Target protein e.g. secondary structure. In this case, the values must be space separated and organized from lower to higher values. For example, the input 150 100 200 220 will create four windows covering residues 1 to 49, 50 to 99, 100 to 199 and 200 to 219.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. After this, the windows of the histograms will be created and for each experiment in the .tarprot file the detected cleavage sites will be assigned to the corresponding windows. Cleavage sites are only counted once per experiment, independently of how many peptides share the same cleavage site. In addition, the total number of cleavage sites identified considering the results of all experiments is also calculated. Since most of the time the protein under study is a recombinant protein containing purification tags or only a region of the native protein, histograms are created for the residue numbers in the recombinant protein and for the residue numbers in the native protein. For this to be possible the sequence of the native protein must have been provided during the creation of the .tarprot file. In the last case cleavage sites outside the native sequence contained in the recombinant protein are discarded and the residue numbers used for the definition of the windows

and cleavages sites are the residue numbers of the native sequence. After the analysis is done the .hist file will be automatically loaded and the results shown in a new window (Figure 7.8).

The output

The Histograms analysis window will display the results contained in a .hist file (Figure 7.8). In the histogram, experiments are shown in the order specified when creating the .tarprot file. In addition, the values for the histograms considering the results from all experiments (all FP) will be displayed as the last bar and colored in gray. Placing the mouse over the plot will display information at the bottom of the window. The information displayed includes the selected window (Win), the experiment represented by the bar (Exp) and the number of cleavages (Cleavages).

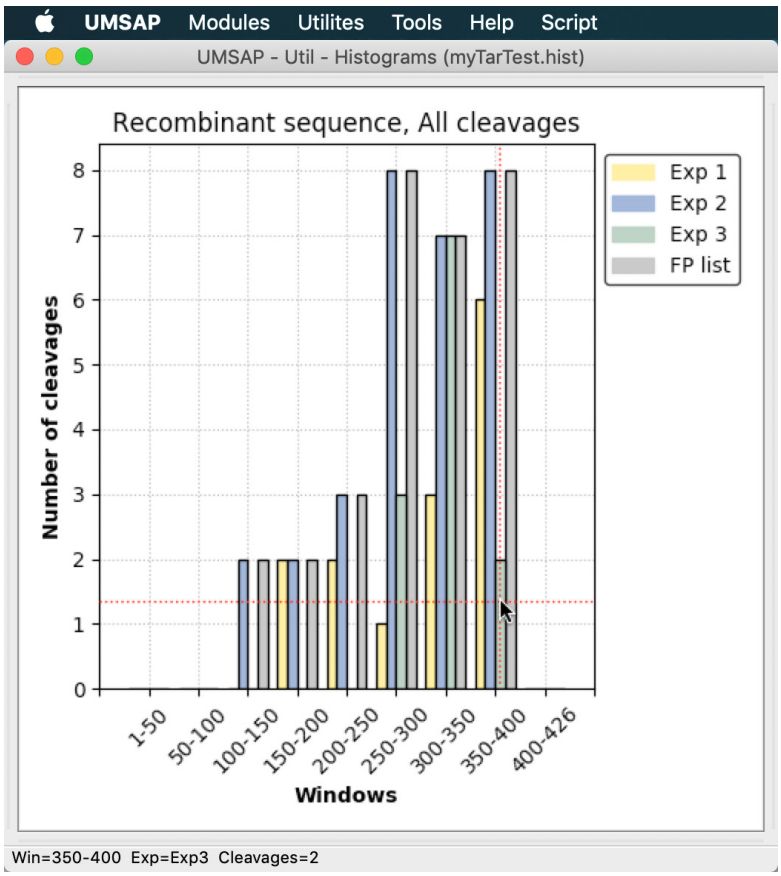

Figure 7.8: The Histograms analysis window. This window allows to visualize the results contained in a .hist file.

The Tools menu

The Tools menu allows to show the results for the recombinant or native sequence and to show only unique cleavages or the total count of the detected cleavages. In addition, users may save an image of the plot or reset the state of the plot.

7.2.6 Sequence Alignments

The Sequence Alignments utility generates sequence alignments between the FP for each experiment and the sequence of the recombinant protein. The list of FP is generated from a .tarprot file (see page 40). The list of FP is a non redundant list. How to generate the .tarprot file is discussed in Chapter 6.

The interface

The Sequence Alignments window is divided in three regions (Figure 7.9).

Region 1 contains three buttons allowing user to quickly delete all provided input and start a new calculation. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to generate the alignments. The Tarprot file button allows users to browse the file system to select the .tarprot file that will be used for the analysis. Only one .tarprot file can be provided here. The Sequence alignments utility generates multiple files that will be saved in a folder named Sequences. The Output folder button allows users to browse the file system to select a location for the output folder Sequences. If the Output folder option is left empty, the output folder Sequences will be created in the same directory as the .tarprot file. If there is a Sequence folder in the selected Output folder, then UMSAP will create a new Sequences folder with the date and time to the seconds added to the end of the name in order to avoid overwriting any file.

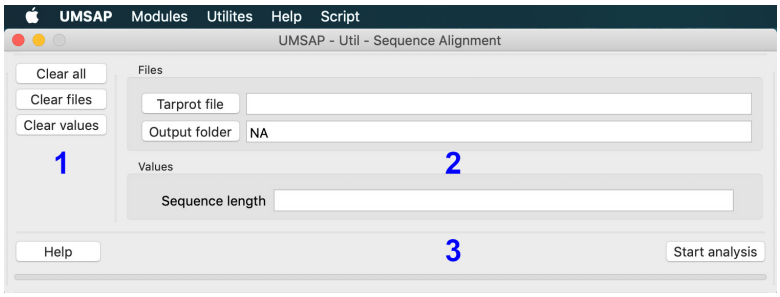

Figure 7.9: The Sequence Alignments window. This window allows to generate sequence alignment files between the FP of each experiment and the sequence of the recombinant protein under study.

The parameter Sequence length allows to define the maximum number of residues per line in the short version of the sequence alignment files. The value here is expected to be an integer greater than zero.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. After this, the list

of FP will be generated from the .tarprot file and UMSAP will generate the sequence alignments.

The output

The output of the Sequence alignments utility is composed of several files that will be saved inside a folder named Sequences. Each file will be a plain text file containing an alignment. Alignments will be generated for each experiment and for the entire FP list. The sequences of the FP in each file will be N-terminally organized. Files for the recombinant and native sequences are generated. In addition, files containing one sequence per line or the specified maximum number of residues per line are also created. The sequence alignment files can be viewed with any text editor since they are just plain text files.

7.2.7 Update Results

As discussed in Section 3.5, UMSAP can only read the .tarprot file from previous versions. The Update Results utility offers a way to quickly generate files for the optional analyses allowed in the Targeted Proteolysis module that are compatible with the current version of UMSAP.

The interface

The Update Results utility does not have a window since there are no options to specify. When the utility is selected users will be asked to select a .tarprot file and then users must select the output folder. That is all.

The analysis

UMSAP will read the .tarprot file and will perform the optional analysis specified in the .tarprot file. This will result in the creation of up to date files for the optional analyses specified in the .tarprot file. The up to date files can be viewed with the current version of UMSAP.

The output

UMSAP will generate the files discussed in this chapter as required by the specified optional analyses found in the given .tarprot file. All generated files will be saved in a TarProtUpdate folder created inside the specified Output folder. If there is already a TarProtUpdate folder in the Output folder, then the current date and time to the seconds will be add to the folder name in order to avoid overwriting previous files.

7.2.8 Custom Update of Results

The Custom Update of Results utility is similar to the Update Results utility because they both allows to use a .tarprot file from an older version of UMSAP to generate files for the optional analyses available in the Targeted Proteolysis module that are compatible with the current version of UMSAP. The main difference is that with Custom Update of Results a custom update can be done.

The interface

The Custom Update of Results utility does not have a window. When the utility is selected users will be asked to select a .tarprot file and then the interface for the Targeted Proteolysis module is created and the information found in the selected .tarprot file is used to fill the fields in the interface of the module, see Figure 6.1.

The analysis

After the interface for the Targeted Proteolysis module is created and filled with the information found in the selected .tarprot file, users may modify the values or add information to perform optional analyses that were not performed with the previous versions of UMSAP, see Chapter 6 for more details.

The output

The output generated depends on the options given to the Targeted Proteolysis module, see Chapter 6 for details.

7.3 General Utilities

7.3.1 Correlation Analysis

The Correlation Analysis utility calculates the correlation in the MS data used as input for UMSAP.

The interface

The Correlation Analysis window is divided in four regions (Figure 7.10).

Region 1 contains three buttons allowing users to quickly delete all provided input and start a new calculation. The Clear all button will delete all user provided input and will reset the state of the list boxes in Region 3. The Clear files button will delete the path to the user provided files and will reset the state of the list boxes in Region 3. Finally, the Clear values button will delete all user provided values.

Region 2 contains the fields where users provide the information needed in order to calculate the correlation between the data. The Data file button allows users to browse the file system to select the data file that will be used for the analysis. The data file is expected to be a plain text file with tab separated columns and the name of the columns in the first row of the file. In addition, columns to be analyzed must contain only numbers and must be of the same length. Only .txt files can be provided here. The Output file button allows users to browse the file system to select the location and name of the output file. If left empty, the name of the output file will be the same as the data file and the .corr file will be saved in the same folder containing the data file. If this default behavior leads to an old file been overwritten, then the date and time to the seconds will be used to make the name of the .corr file unique and avoid overwriting old files.

The Data normalization list allows users to select a normalization algorithm to be performed before the correlation analysis. Currently, the possible options are a Log_2 normalization or no normalization. The list will we expanded in future versions. The Correlation method allows to select the correlation method to use.

Region 3 contains two list boxes and a button. The list box to the left will display the names of the columns present in the data file, once the data file is selected. Loading of the column names is automatically done after selecting the data file using the Data file button in Region 1 or pressing the Enter key while the text box has the focus of the keyboard. Columns in the left list box can not be deleted, except in the case of loading a different data file or using the Clear input or Clear all buttons in Region 1. The Add columns button in the middle of Region 3 will add the selected columns in the left list box to the right list box. The columns will be added to the right list box in the same order as they are selected from the left list box.

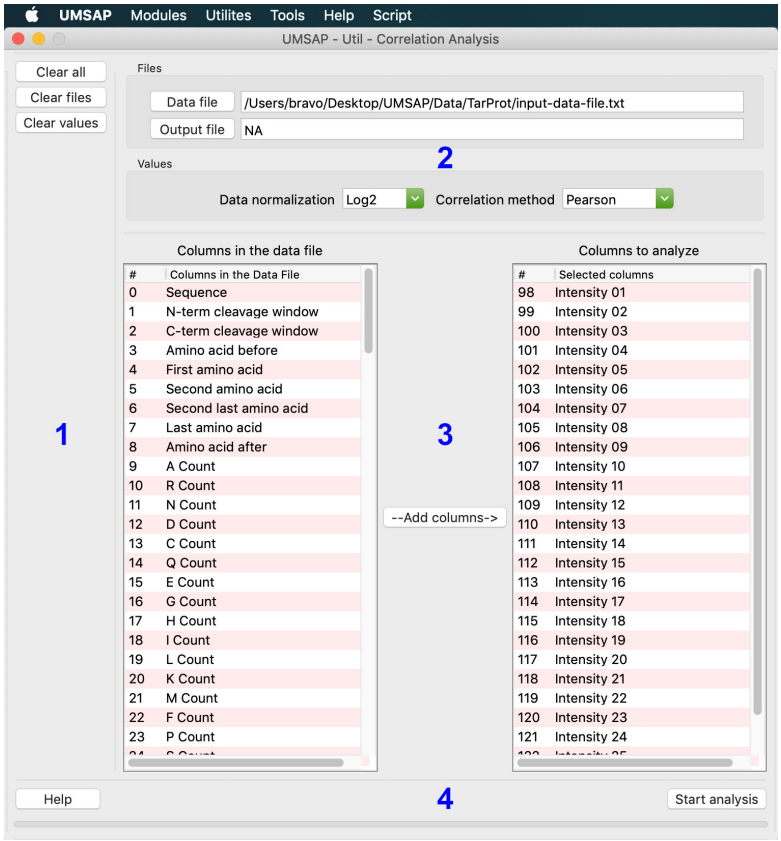

Figure 7.10: The Correlation Analysis window. This windows allows to perform a correlation analysis of the data contained in a given data file.

The list box to the right of Region 3 contains the columns for which the correlation analysis will be performed. Correlation between all columns in the right list box will be performed. The order of the rows and columns in the resulting matrix containing the correlation coefficients will be the same as the order of the columns shown in the right list box. Therefore, users are advised to fill the right list box in such a way that replicates of the same experiment are consecutive to each other in the right list box. Columns in the right list box can be deleted by selecting the columns and then using the right mouse button over the right list box or using the Tools menu. Columns in the right list box will be unique, meaning that a column can only be added once.

Region 4 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

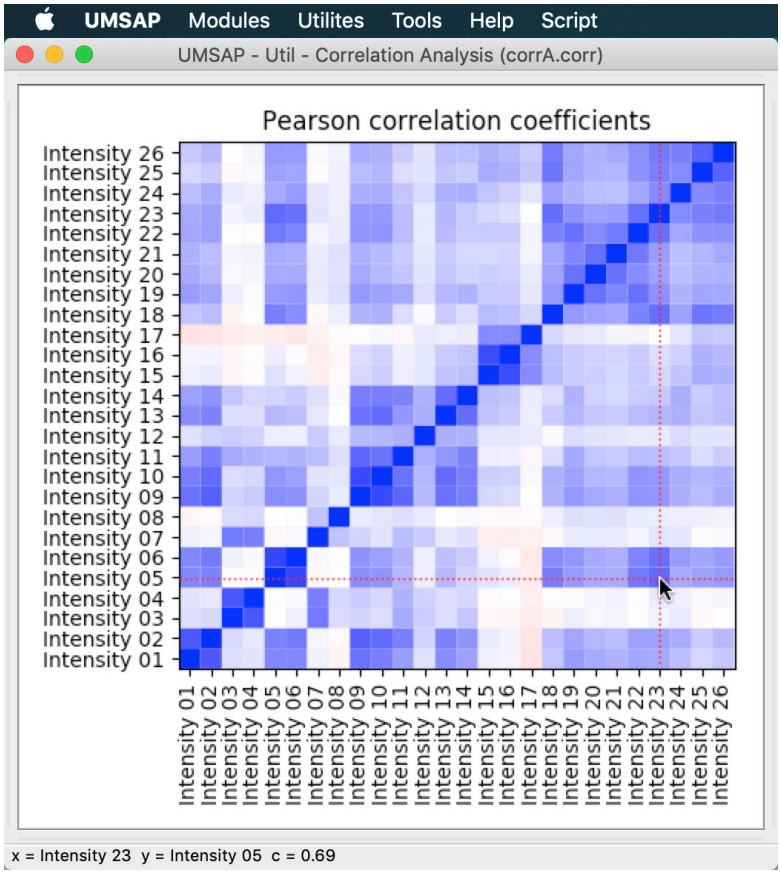

Figure 7.11: The Correlation Analysis result window. The correlation coefficients are shown in a color coded matrix. Values between -1 to 0 are shown in shades of red, 0 is shown in white and values between 0 to 1 in shades of blue. NA values are shown in green.

The analysis

First, UMSAP will check the validity of the user provided input. Then, columns in the right list box are read from the data file. The columns must contain only numbers and the same amount of rows must be found in all columns. Failing to comply with this will result in the program aborting the analysis. After this, the selected normalization procedure is applied to the data. Finally, the correlation coefficients are calculated using the selected method. If any of the coefficients cannot be calculated, then the corresponding coefficient is set to NA. After the analysis is done the results will be automatically loaded and displayed in a new window (Figure 7.11).

The output

The extension .corr is reserved for a file containing the output from a correlation analysis.

The results in a .corr file will be shown as a color coded matrix (Figure 7.11). Values between -1 to 0 will be shown in shades of red, 0 will be shown as white and values between 0 to 1 will be shown in shades of blue. NA values will be shown in green. The columns and rows of the matrix are the column numbers used to calculate the correlation. Information about a specific matrix element can be obtain by simply putting the mouse pointer over the matrix element.

The Tools menu

The Tools menu in the configuration window of the correlation analysis (Figure 7.10) allows users to empty the right list box or to remove only the selected rows.

The Tools menu in the window showing the results in a .corr file (Figure 7.11) allows user to create an image of the plot and to export the results to one of the modules in UMSAP. After selecting to export the data, the same window used to configure the Results - Control Experiment for the selected module will appear allowing users to configure this parameter and send the configuration to the module window. The exported information includes the path to the data file used to calculate the correlation coefficients.

7.3.2 Create Input File

The Create Input File utility allows user to read a .limprot, .protprof or .tarprot file to create a .uscr file. How to generate the .limprot, .protprof or .tarprot file is discussed in Chapter 4, Chapter 5 and Chapter 6 respectively. The .uscr file is used to prepare a module to perform an analysis without users having to type in all the information required by the module. Thus, if a second analysis of a data file is needed users can quickly load the .uscr file into UMSAP, apply the required modifications in the window of the module and start the analysis without having to type in or modify the options that will be the same between the old and new analysis. Each .uscr file can contain information for configuring one module and one analysis.

The interface

The Create Input File utility does not have a window since there are no options to specify. When the utility is selected users will be asked to select the .limprot, .protprof or .tarprot file and then users must select the output file. That is all.

The analysis

First, UMSAP will check the validity of the user provided input. Then the .limprot, .protprof or .tarprot file will be read in and the configuration values will be extracted and saved in the .uscr file. The utility is able to process .tarprot files from previous versions of UMSAP.

The output

The .uscr file has a simple format in which each line has a keyword and an argument. The keyword and the argument are separated by a colon (:). The following are examples of the format of the .uscr file for ech module.

Example for the Limited Proteolysis module:

Module: Limited Proteolysis
Data file: /Users/kenny/data-kbr.txt
Sequence (rec): /Users/kenny/seqA.txt
Sequence (nat): /Users/kenny/seqA-nat.txt
Output folder: /Users/kenny/test
Output name: myLimTest
Target protein: Mis18alpha
Score value: 10
Sequence length: 100
d-value: NA
dm-value: 8
Data normalization: Log2
a-value: 0.050
b-value: Equal alpha
y-value: 0.8
Sequence: 0
Detected proteins: 34
Score: 42
Columns to extract: 0 1 2 3 4-10
Results: 69-71; 81-83, 78-80, 75-77, 72-74, NA; NA, NA, NA, 66-68, NA; 63-65, 105-107, 102-104, 99-101, NA; 93-95, 90-92, 87-89, 84-86, 60-62

Example for the Proteome Profiling module:

Module: Proteome Profiling
Data file: /Users/kenny/proteinGroups-kbr.txt
Output folder: /Users/kenny/test
Output name: myProtTest
Score value: 320
Z score: 10
Data normalization: Log2
a-value: 0.050
Median correction: True
P correction: Benjamini - Hochberg
Detected proteins: 0
Gene names: 6
Score: 39
Exclude proteins: 171 172 173
Columns to extract: 0 1 2 3 4-10
Results: 105 115 125, 130 131 132; 106 116 126, 101 111 121; 108 118 128, 103 113 123
Conditions: DMSO, H2O
Relevant Points: 30min, 1D
Control Type: One Control per Column
Control Label: MyControl

Example for the Targeted Proteolysis module:

Module: Targeted Proteolysis
Data file: /Users/kenny/data-ms.txt
Sequence (rec): /Users/kenny/data-seq.txt
Sequence (nat): P31545
PDB file: NA
Output folder: /Users/kenny/test
Output name: myTarTest
Target protein: efeB
Score value: 200
Data normalization: Log2
a-value: 0.050
Positions: 5
Sequence length: 100
Histogram windows: 50
PDB ID: 2y4f;A
Sequence: 100
Detected proteins: 38
Score: 44
Columns to extract: 0 1 2 3 4-10
Results: 98-105; 109-111; 112 113 114; 115-117 120

The menu entry Script/Run Input File allows to load the .uscr file into UMSAP.

7.3.3 Merge aadist Files

The .aadist files contain an AA distribution analysis as described in subsection 7.2.1. The Merge aadist Files utility allows to merge several .aadist files into a single file.

The interface

The Merge aadist Files window is divided in four regions (Figure 7.12).

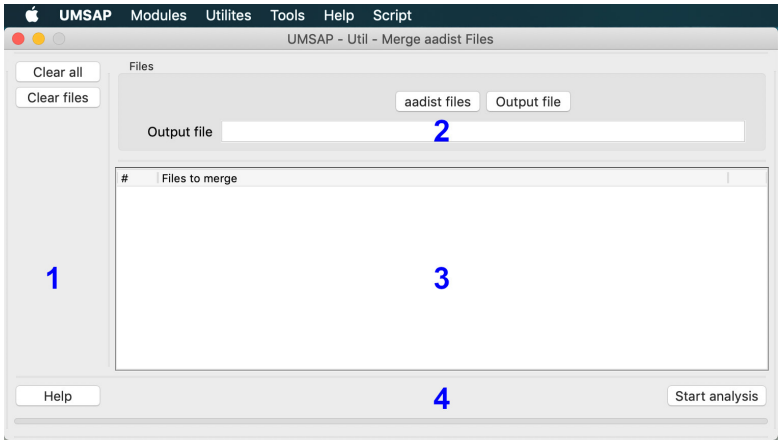

Figure 7.12: The Merge .aadist Files window. This window allows users to merge several .aadist files in a single file.

Region 1 contains two buttons allowing users to quickly delete all provided input and start a new calculation. The Clear all button will delete all user provided input and will reset the state of the list box in Region 3. The Clear files button will reset the state of the list box in Region 3 and delete all user provided paths to files.

Region 2 contains the fields where users provide the information needed in order to merge the .aadist files. The aadist files button allows users to browse the file system to select multiple .aadist file from a folder. Only .aadist files can be selected here. Once the files are selected the complete path to the files will be displayed in the list box in Region 3. The Output file button allows users to browse the file system to select the location and name of the output file.

Region 3 contains a list box showing all selected .aadist files. Files can only be added one time to the list box. The order of the files in the list box is meaningless. Selected files can be deleted from the list box by pressing the right mouse button over the list box or using the Tools menu.

Region 4 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. Then, the number of positions and experiments in each .aadist files are checked. If they do not match, the merging is aborted. Finally, UMSAP check that all AA distributions were originated from the same sequence and abort the task at hand if they do not. After this, files are merged. For merging the files, UMSAP adds the number of times each amino acids appears in a given position for each file. When all files have been merged the results will be automatically loaded and displayed in a new window (Figure 7.4). The significance level for the merged file is the highest value found in all files that were merged.

The output

The Merge aadist Files utility generates a .aadist file. This file can be visualized in the same way as the output from the AA distribution utility, see Figure 7.4 for more details.

7.3.4 Read Output File

The Read Output File utility simply loads an output file generated by UMSAP. After selecting this option from the Utility window (Figure 7.1) or the Utilities menu entry, a dialog box will be presented allowing users to select some of the output files generated by UMSAP. Currently, only .aadist, .corr, .cutprop, .hist, .limprot, .protprof, and .tarprot files can be selected. After selecting the file, the appropriate window showing the graphical representation of the file will be created.

7.3.5 Short Data Files

The Short Data Files utilities allows users to create short versions of the Data file used to create a .limprot, .protprof or .tarprot file. How to generate the .limprot, .protprof or .tarprot file is discussed in Chapter 4, Chapter 5 and Chapter 6 respectively.

The interface

The Short Data Files window is divided in three regions (Figure 7.13).

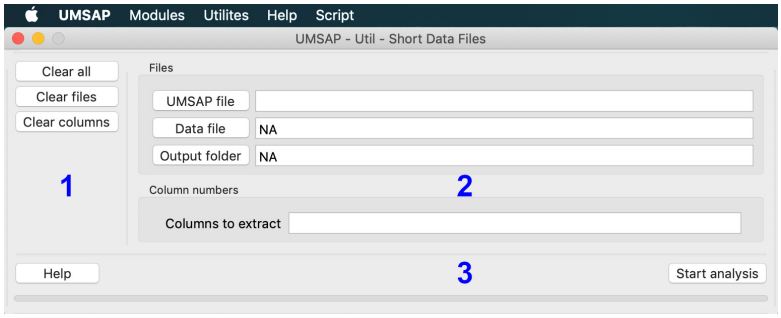

Figure 7.13: The Short Data File window. This window allows to generate smaller versions of the Data file used to generate a .tarprot file.

Region 1 contains three buttons allowing user to quickly delete all provided input and start a new analysis. The Clear all button will delete all user provided input. The Clear files button will delete the path to all user provided files. Finally, the Clear values button will delete all user provided numerical values.

Region 2 contains the fields where users provide the information needed in order to generate the short data files. The UMSAP file button allows users to browse the file system to select the .limprot, .protprof or .tarprot file that will be used for the analysis. Only one file can be provided here. The Data file button allows users to browse the file system to select the data file that will be used for the analysis. The Short Data File utility generates multiple files that will be saved in a folder named Data. The Output folder button allows users to browse the file system to select a location for the output folder Data. If the Output folder option is left empty, the output folder Data will be created in the same directory as the UMSAP file. If there is a Data folder in the selected Output folder, then UMSAP will create a new Data folder with the date and time to the seconds added to the end of the name in order to avoid overwriting any file.

The parameter Columns to extract allows to define which columns from the data file will be extracted to the short data files. The values here are expected to be integers greater than zero.

Region 3 contains the Help and Start analysis buttons and a progress bar. The Help button leads to an online tutorial while the Start analysis button will start the analysis. The progress bar will give a rough estimate of the remaining time for completing the analysis.

The analysis

First, UMSAP will check the validity of the user provided input. If only an UMSAP file

is given, then the data file location will be read from the UMSAP file. After this, the short data files will be written to the Data folder.

The output

The output consist of three files that will be saved in a Data folder. Assuming that the name of the Target protein is efeB, the name of the files will be:
all-columns-all-efeB-records.txt
selected-columns-all-efeB-records.txt
selected-columns-relevant-efeB-records.txt

These files are just shorter versions of the data file containing only relevant information about the Target protein. They are plain text files with a tabular format. The first row contains the name of the columns and columns are tab separated. The file all-columns-all-efeB-records.txt contains the same number of columns as the data file but only the rows for the target protein. The file selected-columns-all-efeB-records.txt contains only rows for the target protein and the columns specified in Columns to extract. The selected-columns-relevant-efeB-records.txt file is similar to the previous file but contains only the relevant peptides of the target protein. These files can be viewed with any text editor.

Chapter 8

License Agreement

Utilities for Mass Spectrometry Analysis of Proteins and its source code are governed by the following license:

Upon execution of this Agreement by the party identified below ("Licensee"), Kenny Bravo Rodriguez (KBR) will provide the Utilities for Mass Spectrometry Analysis of Proteins software in Executable Code and/or Source Code form ("Software") to Licensee, subject to the following terms and conditions. For purposes of this Agreement, Executable Code is the compiled code, which is ready to run on Licensee's computer. Source code consists of a set of files, which contain the actual program commands that are compiled to form the Executable Code.

1. The Software is intellectual property owned by KBR, and all rights, title and interest, including copyright, remain with KBR. KBR grants, and Licensee hereby accepts, a restricted, non-exclusive, non-transferable license to use the Software for academic, research and internal business purposes only, e.g. not for commercial use (see Clause 7 below), without a fee.

2. Licensee may, at its own expense, create and freely distribute complimentary works that inter-operate with the Software, directing others to the Utilities for Mass Spectrometry Analysis of Proteins web page to license and obtain the Software itself. Licensee may, at its own expense, modify the Software to make derivative works. Except as explicitly provided below, this License shall apply to any derivative work as it does to the original Software distributed by KBR. Any derivative work should be clearly marked and renamed to notify users that it is a modified version and not the original Software distributed by KBR. Licensee agrees to reproduce the copyright notice and other proprietary markings on any derivative work and to include in the documentation of such work the acknowledgment: "This software includes code developed by Kenny Bravo Rodriguez for the Utilities for Mass Spectrometry Analysis of Proteins software".

Licensee may not sell any derivative work based on the Software under any circumstance. For commercial distribution of the Software or any derivative work based on the Software a separate license is required. Licensee may contact KBR to negotiate an appropriate license for such distribution.

3. Except as expressly set forth in this Agreement, THIS SOFTWARE IS PROVIDED

"AS IS" AND KBR MAKES NO REPRESENTATIONS AND EXTENDS NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY PATENT, TRADEMARK, OR OTHER RIGHTS. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE AND/OR ASSOCIATED MATERIALS. LICENSEE AGREES THAT KBR SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES WITH RESPECT TO ANY CLAIM BY LICENSEE OR ANY THIRD PARTY ON ACCOUNT OF OR ARISING FROM THIS AGREEMENT OR USE OF THE SOFTWARE AND/OR ASSOCIATED MATERIALS.

4. Licensee understands the Software is proprietary to KBR. Licensee agrees to take all reasonable steps to insure that the Software is protected and secured from unauthorized disclosure, use, or release and will treat it with at least the same level of care as Licensee would use to protect and secure its own proprietary computer programs and/or information, but using no less than a reasonable standard of care. Licensee agrees to provide the Software only to any other person or entity who has registered with KBR. If Licensee is not registering as an individual but as an institution or corporation each member of the institution or corporation who has access to or uses Software must agree to and abide by the terms of this license. If Licensee becomes aware of any unauthorized licensing, copying or use of the Software, Licensee shall promptly notify KBR in writing. Licensee expressly agrees to use the Software only in the manner and for the specific uses authorized in this Agreement.

5. KBR shall have the right to terminate this license immediately by written notice upon Licensee's breach of, or non-compliance with, any terms of the license. Licensee may be held legally responsible for any copyright infringement that is caused or encouraged by its failure to abide by the terms of this license. Upon termination, Licensee agrees to destroy all copies of the Software in its possession and to verify such destruction in writing.

6. Licensee agrees that any reports or published results obtained with the Software will acknowledge its use by the appropriate citation as follows:

"Utilities for Mass Spectrometry Analysis of Proteins was developed by Kenny Bravo Rodriguez at the University of Duisburg-Essen."

Any published work, which utilizes Utilities for Mass Spectrometry Analysis of Proteins, shall include the following reference:

Kenny Bravo-Rodriguez, Birte Hagemeier, Lea Drescher, Marian Lorenz, Michael Meltzer, Farnusch Kaschani, Markus Kaiser and Michael Ehrmann. (2018). Utilities for Mass Spectrometry Analysis of Proteins (UMSAP): Fast post-processing of mass spectrometry data. Rapid Communications in Mass Spectrometry, 32(19), 1659–1667.

Electronic documents will include a direct link to the official Utilities for Mass Spectrometry Analysis of Proteins page at: www.umsap.nl

7. Commercial use of the Software, or derivative works based thereon, REQUIRES A COMMERCIAL LICENSE. Should Licensee wish to make commercial use of the

Software, Licensee will contact KBR to negotiate an appropriate license for such use. Commercial use includes: (1) integration of all or part of the Software into a product for sale, lease or license by or on behalf of Licensee to third parties, or (2) distribution of the Software to third parties that need it to commercialize product sold or licensed by or on behalf of Licensee.

8. Utilities for Mass Spectrometry Analysis of Proteins is being distributed as a research tool and as such, KBR encourages contributions from users of the code that might, at KBR's sole discretion, be used or incorporated to make the basic operating framework of the Software a more stable, flexible, and/or useful product. Licensees who contribute their code to become an internal portion of the Software agree that such code may be distributed by KBR under the terms of this License and may be required to sign an "Agreement Regarding Contributory Code for Utilities for Mass Spectrometry Analysis of Proteins Software" before KBR can accept it (contact umsap-licenses@umsap.nl for a copy).

UNDERSTOOD AND AGREED.

Contact Information:

The best contact path for licensing issues is by e-mail to umsap-licenses@umsap.nl