

# 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](http://www.umsap.nl)

Utilities for Mass Spectrometry Analysis of Proteins  
Copyright © 2017 Kenny Bravo Rodriguez.  
All Rights Reserved.

# Contents

<b>List of Figures</b>	<b>IV</b>
<b>List of Tables</b>	<b>V</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Citing Utilities for Mass Spectrometry Analysis of Proteins . . . . .	1
1.2 Acknowledgments . . . . .	2
1.3 Copyrights Notes . . . . .	2
<b>2 Obtaining and Installing Utilities for Mass Spectrometry Analysis of Proteins</b>	<b>16</b>
2.1 Obtaining Utilities for Mass Spectrometry Analysis of Proteins . . . . .	16
2.2 Installing Utilities for Mass Spectrometry Analysis of Proteins . . . . .	16
2.3 Uninstalling Utilities for Mass Spectrometry Analysis of Proteins . . . . .	18
<b>3 Workflow in Utilities for Mass Spectrometry Analysis of Proteins</b>	<b>19</b>
3.1 The input data files . . . . .	20
3.2 Using Utilities for Mass Spectrometry Analysis of Proteins . . . . .	20
3.3 Navigating through Utilities for Mass Spectrometry Analysis of Proteins	21
3.3.1 Keyboard shortcuts . . . . .	21
3.4 The output files . . . . .	22
3.5 Backward compatibility . . . . .	22
<b>4 The Targeted Proteolysis module</b>	<b>23</b>
4.1 Definitions . . . . .	23
4.2 The input data files . . . . .	24
4.3 The interface . . . . .	24
4.3.1 The Tools menu . . . . .	28

4.4	The analysis . . . . .	28
4.5	The output files . . . . .	30
4.6	Visualizing the output files . . . . .	31
4.6.1	The Tools menu . . . . .	33

# List of Figures

3.1	The main window of UMSAP . . . . .	19
4.1	The Targeted Proteolysis window . . . . .	25
4.2	Data organization prior to the ANCOVA test . . . . .	29
4.3	The structure of the Output folder from the Targeted Proteolysis module	30
4.4	The Fragment analysis window . . . . .	31

# List of Tables

1.1	List of modules used by UMSAP . . . . .	2
3.1	List of built-in keyboard shortcuts . . . . .	22

# 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](http://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

Copyright © 2005-2019, NumPy Developers.  
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 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  
=====

Copyright (c) 2010-2019, PyInstaller Development Team  
Copyright (c) 2005-2009, Giovanni Bajo  
Based on previous work under copyright (c) 2002 McMillan Enterprises, Inc.

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) 2005-2015, 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

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### 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](http://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](http://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](http://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.

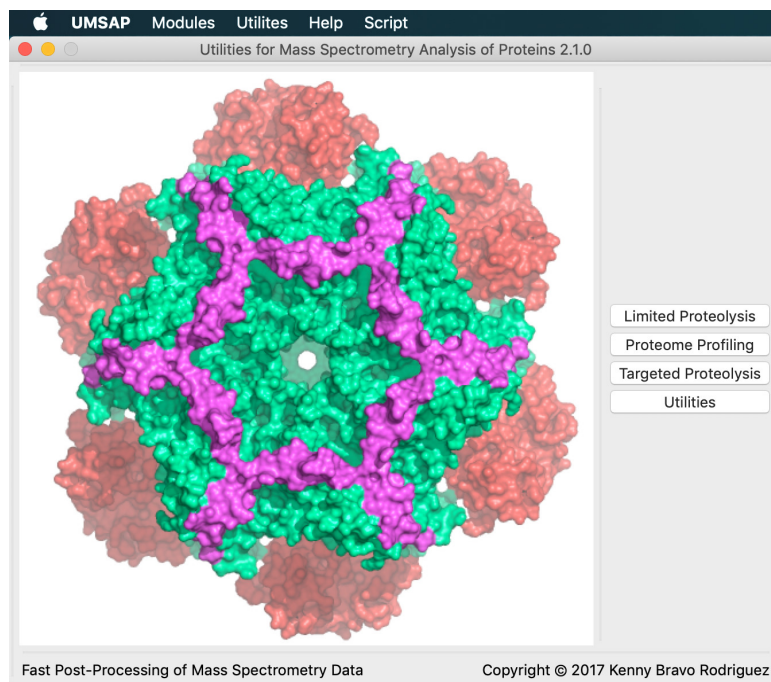
## **2.3 Uninstalling Utilities for Mass Spectrometry Analysis of Proteins**

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](mailto: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 .usr 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 .usr 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 .usr 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

**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 ?? and ?. How to generate a .tarprot file is explained in Chapter 4.

## Chapter 4

# 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.

### 4.1 Definitions

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

- *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 26).
- *Filtered peptide*: a relevant peptide with a significantly different behavior in the control and a given experiment 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	MKKTAIAIAVALAGFATVAQAASWSHPQFEKIEGRRDRGQKTQSAPGTL	50
SEQ1	MKKTAIAIAV.....	10
SEQ2	..KTAIAIAV.....	8
SEQ3	.....IAIAVAL.....	7
SEQ4	.....ATVAQAASWS.....	10
SEQ5	.....DRGQKTQSAPG...	11

## 4.2 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 *Columns in the input file* 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.

## 4.3 The interface

The window of the Targeted Proteolysis module is divided in four regions (Figure 4.1).

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.

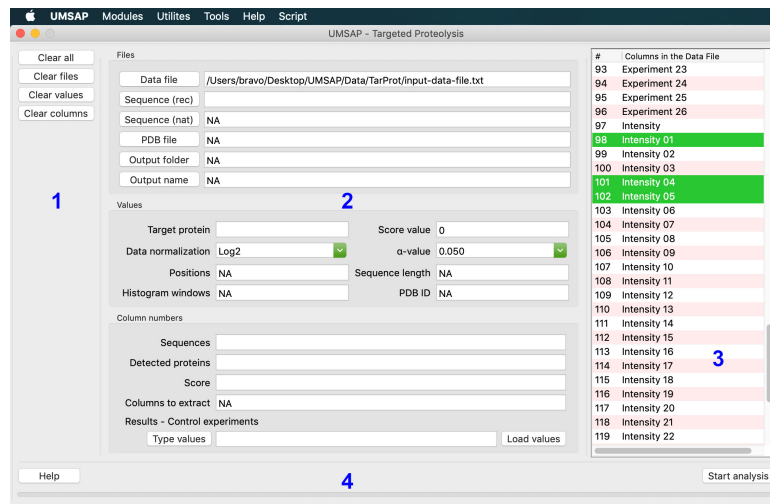
1.- The Data file button allows users to browse the file system and select a data file. Only .txt files can be selected here. Once the data file is selected, the name of the columns in the file will be shown in the list box in Region 3. If the path to the data file is typed in, the display of the name 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 26 for more details.

See Section 3.1 for more details about the data files.



**Figure 4.1: 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.

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 *User defined values* in Region 2 contains eight 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 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  $\alpha$ -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 28 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 ?? 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 ?? 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 ?? 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 *Columns in the input file* 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 since they are already corrected.

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 *User defined 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 the section *User defined values* in 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 30 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 4.1).

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.

### 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 .usr file with the given options to the module before running the analysis. If something goes wrong during the analysis having the .usr file means that users do not have to type the values of all the parameters again.

## 4.4 The analysis

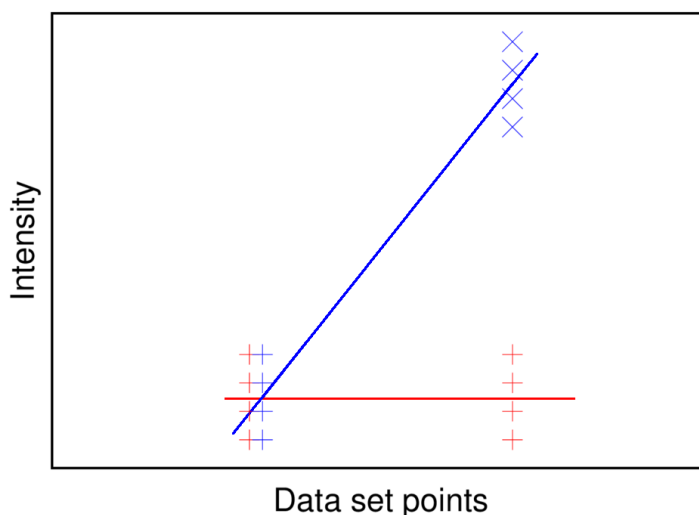
First, UMSAP will check the validity of the user provided input. Then, the data file is processed as follow. All rows in the data file containing peptides that do not belong to the target protein are removed. Then, all rows containing peptides from the target protein but with Scores values lower than 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 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 4.2. 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 (??) and generates the input file with extension

.usr (??) and a copy of the FP list (??). 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 ???. The optional analyses are: amino acid distribution (??), sequence alignments (??), histograms of detected cleavages (??), mapping of detected cleavages to a .pdb file (??) and short data files (??).

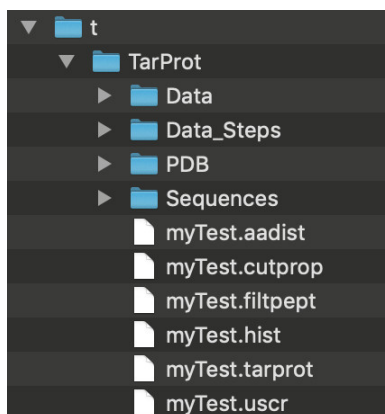
If the sequence of the native protein is given the module performs a sequence alignment between the native and recombinant sequences. The alignment allows UMSAP to translate the results obtained with the residue numbers of the recombinant protein to the residue numbers of the native protein. This is 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.



**Figure 4.2: 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.

## 4.5 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 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 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 *myTest*, the target protein was *efeB* and all optional analyses were performed.



**Figure 4.3: 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 Columns to extract (see page 27) 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 ??.

If the parameter Histograms window (see page 26) is different than NA, then histograms of the detected cleavage sites will be created as described in ??. 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 26) was set, then the detected cleavage sites are mapped to the structure in the .pdb file, as described in ?? 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.

If the parameter Sequence length (see page 26) is different than NA, then sequence alignment files will be created as described in ??. 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 26) is different than NA, then an AA distribution analysis is performed as described in ???. The file t/TarProt/myTest.aadist contains the AA distribution analysis.

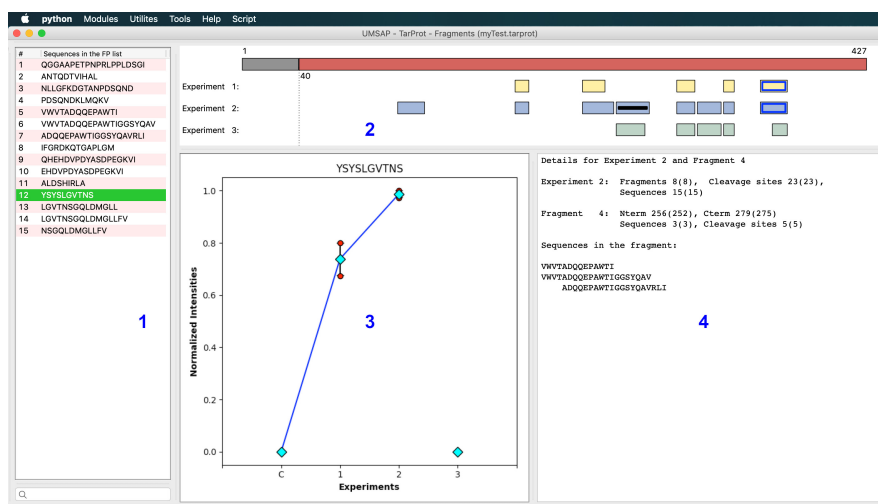
The file myTest.cutprop contains the results of the number of cleavages per residue as discussed in ??? while the file myTest.filtpept contains the list of FP as described in ???. The file myTest.uscr is the input file that can be used to quickly reanalyzed the data file without having to type in all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

## 4.6 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 ???, ??? and ??? 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 4.4).



**Figure 4.4: 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 27. 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.

### 4.6.1 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 4.4).