

# New Advances in Proteomics Informatics for the Trans-Proteomic Pipeline

Luis Mendoza, David Shteynberg,  
Michael Hoopmann, Eric Deutsch

ASMS evening Workshop  
June 4<sup>th</sup>, 2019 – A304

# Agenda

- **Introduction to TPP and overview of the latest features in 5.2**
- Proteogenomics & PEFf applications
- Analyzing PTMs with PTMProphet
- Cross-linking analysis with Kojak 2.0
- Deploying the TPP using Docker containers and cloud computing platforms
- Open Discussion and Feedback

# Group presentations at ASMS

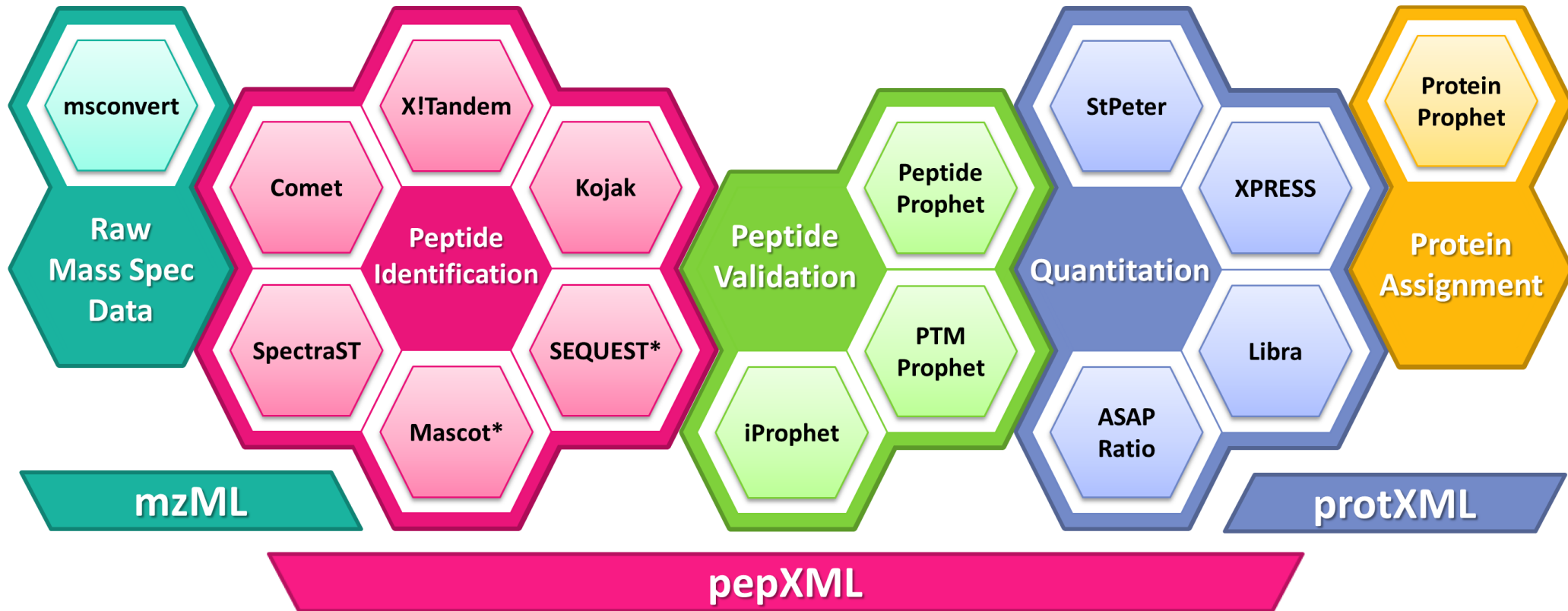
## For more info:

- Kojak 2.0 (MP 044)
- PEFF in TPP (MP 417)
- PTMPProphet (WP 400)

## Booth #118

[www.tppms.org/asms/](http://www.tppms.org/asms/)

# Trans-Proteomic Pipeline (TPP) Overview



**Free and open source** suite of **software tools** and **file formats** that facilitates and standardizes proteomics analysis

# TPP components

- PSM and Peptide validation
- Protein inference
- Quantification
- Spectral Libraries: build and search
- Combine search engines
- Cloud capabilities
- etc...

[www.tppms.org](http://www.tppms.org)

# New features in 5.2.0 (March '19)

- **tpp2mzid**, a new tool for converting *PepXML* and *ProtXML* to *mzIdentML*.
- New **ProteoMapper** tools: **clips** and **promast**, for fast and efficient mapping of peptide sequences to a proteome
  - Supports variants via *PEFF* format
  - Able to do fuzzy matching of sequences, including isobaric (with and without mass modifications)
- New quantitation tool **QUANTIC**, which uses matched ion intensities of identified (MS/MS) spectra
- New tool to generate quantitative **volcano plots** on Proteins and Peptides
- Added a simple utility to **compute p-values** for some common distributions using GSL
- Support for *Magnum* and *Morpheus* search engines
- Expanded UI support for up to 12 channels of isobaric labeling, including **TMT-11**

# Agenda

- Introduction to TPP and overview of the latest features in 5.2
- **Proteogenomics & PEFF applications**
- Analyzing PTMs with PTMProphet
- Cross-linking analysis with Kojak 2.0
- Deploying the TPP using Docker containers and cloud computing platforms
- Open Discussion and Feedback

# What is PEFF?

- PEFF is a “unified format for protein sequence databases to be used by sequence search engines and other associated tools, to enable consistent extraction, display and processing of information such as **post-translational modifications, mutations** and other processing events,” among others.
- The format is plain text, largely FASTA-like for backwards compatibility.

[psidev.info/peff](https://psidev.info/peff)



# PEFF Basics



“\” Keywords

```
>nxp:NX_Q5EE01-1 \PName=Centromere protein w isoform Iso 1
\GName=CENPW \NcbiTaxId=9606 \TaxName=Homo Sapiens \Length=88
\SV=61 \EV=265 \PE=1 \VariantSimple=(4|L)(6|M)(6|V)(8|P)(8|F)
(11|R)(19|H)(19|C)(20|D)(24|Q)(28|L)(28|P)(31|R)(32|*)(40|N)(41|F)
(45|V)(47|F)(52|R)(53|*)(53|Q)(57|D)(59|G)(63|F)(64|V)(12|H)(26|C)
(62|T)(63|S)(74|R)(78|T)(80|M)(86|I)(86|G) \Processed=(1|88|mature
protein)
MALSTIVSQRKQIKRKAPRGFLKRVFKRKKPQLRLEKSGDLLVH } Sequence
LNCLLFVHRLAEESRTNACASKCRVINKEHVLAAAKVILKKS RG
>nxp:NX_Q5EE01-1 \DbUniqueId=NX_Q5EE01-1 ...
...
```

PEFF entry

# Single Amino Acid Variants in PEF

```
>nxp:NX_Q5EE01-1 \PName=Centromere protein w isoform Iso 1
\GName=CENPW \NcbiTaxId=9606 \TaxName=Homo Sapiens \Length=88
\SV=61 \EV=265 \PE=1 \VariantSimple=(4|L)(6|M)(6|V)(8|P)(8|F)
(11|R)(19|H)(19|C)(20|D)(24|Q)(28|L)(28|P)(31|R)(32|*)(40|N)(41|F)
(45|V)(47|F)(52|R)(53|*)(53|Q)(57|D)(59|G)(63|F)(64|V)(12|H)(26|C)
(62|T)(63|S)(74|R)(78|T)(80|M)(86|I)(86|G) \Processed=(1|88|mature
protein)
MALSTIVSQRKQIKRKAPRGFLKRVFKRKKPQLRLEKSGDLLVH
LNCLLFVHRLAEESRTNACASKCRVINKEHVLAAAKVILKKSRRG
>nxp:NX_Q5EE01-1 \DbUniqueId=NX_Q5EE01-1 ...
...
```

# PEFF in TPP

- Search with Comet
- Results in *PepXML*
- Visualize, filter, sort, export...
- Map sequences
- Protein inference

In progress:

- Results validation (decoys...?)
- In/dels
- Signal / transit / etc
- Proteogenomics
- Make your own!
- Other...!?

K. <a href="#">RTVS167.00DNSLSNSR</a> .G <sup>FA</sup>
R. <a href="#">S167.00VAAC160.03T181.01EEPKER</a> .G <sup>FA</sup>
K. <a href="#">LPYGAR</a> .R <sup>FA</sup>
K. <a href="#">IMHDVIKK</a> .V <sup>FA</sup>
K. <a href="#">RHFNAPSHIR</a> .R <sup>FA</sup>
R. <a href="#">Y243.03VANKTMS167.00K</a> .C <sup>FA</sup>
K. <a href="#">SVAMC160.03EM147.04EK</a> .K <sup>FA</sup>
R. <a href="#">YGTVEPQDK</a> .C <sup>FA</sup>
- . <a href="#">M147.04ALLHSGR</a> .V <sup>FA</sup>
R. <a href="#">GVGFDRMTAISEK</a> .R <sup>FA</sup>
R. <a href="#">DLSAS167.00AGHPR</a> .A <sup>FA</sup>
R. <a href="#">TPDGNLDQC160.03K</a> .- <sup>FA</sup>
K. <a href="#">KM147.04VEDGQS167.00GDLK</a> .K <sup>FA</sup>
R. <a href="#">DALCAQNASCEK</a> .T <sup>FA</sup>

TPP :: Peptide Sequence Explorer

Input: [single peptide](#) or a [peptide list](#) (max. 5000 sequences)

Database:

Peptide Sequence:

Fuzzy mapping settings:

Number of unknown amino acids:

Match mass tolerance:

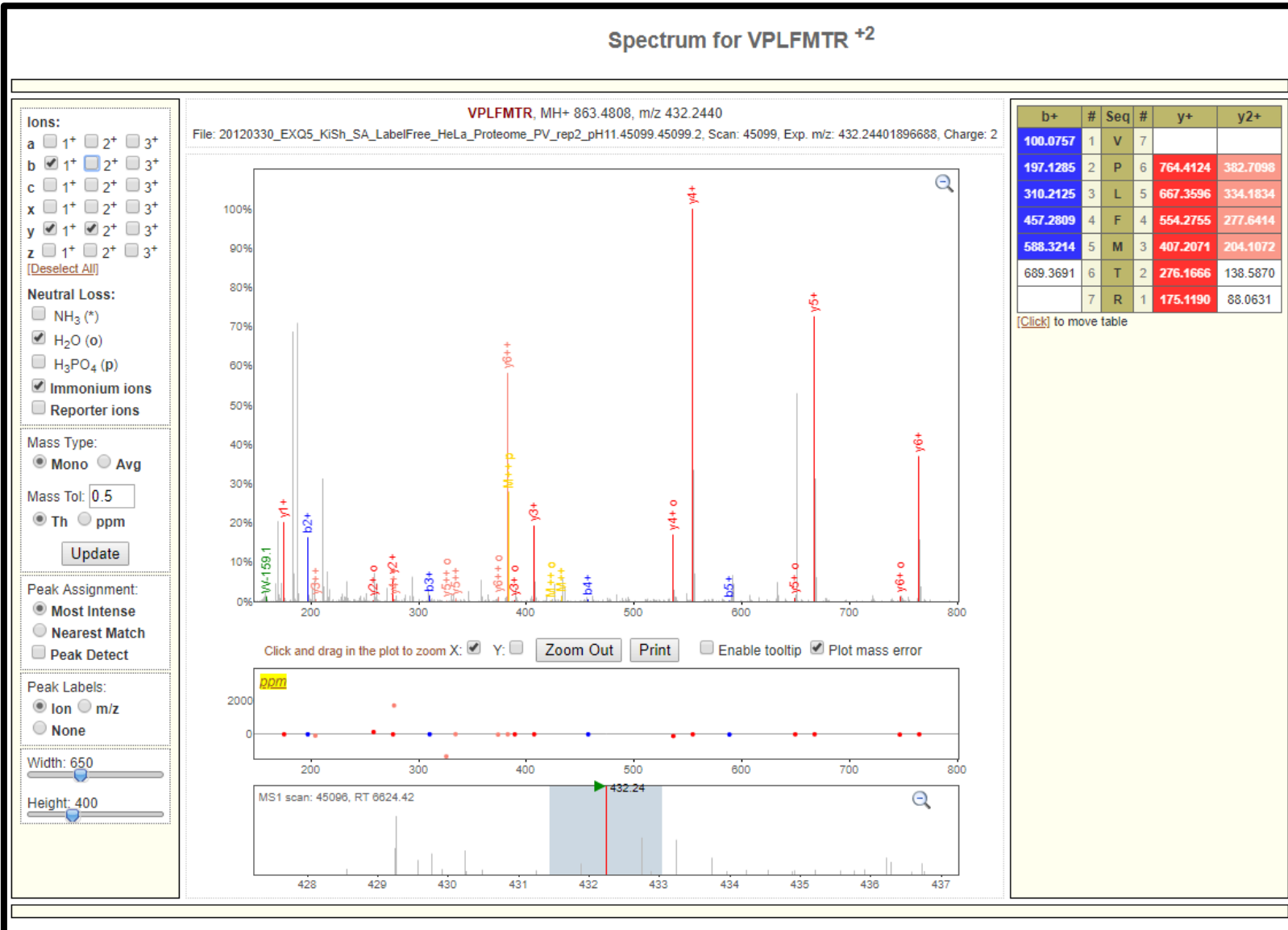
Column Descriptions

Mapping Results :: 468 total results found [Download \(xls\)](#)

Hide redundant mappings

Peptide Sequence	Protein(s)	Position	Original Sequence	nSubs	% original seq
NPSVSVVLR(+14.0157)	> NX_P20062-1	51	L.NPSLYVGLR.L	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
NPSVSVVLR(+14.0157)	> NX_P20062-2	51	L.NPSLYVGLR.L	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
PSQGVKVL(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	6	<div style="width: 60px; height: 10px; background-color: #ccc;"></div>
PSQGVKVL(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	6	<div style="width: 60px; height: 10px; background-color: #ccc;"></div>
PSQVVKGL(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	6	<div style="width: 60px; height: 10px; background-color: #ccc;"></div>
PSQVVKGL(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	6	<div style="width: 60px; height: 10px; background-color: #ccc;"></div>
PSQAAKLR(+14.0157)(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	7	<div style="width: 70px; height: 10px; background-color: #ccc;"></div>
PSQVAKLR(+14.0157)(+0.9840)	> NX_P62805-1	4	G.RGKGGKGLG.K	7	<div style="width: 70px; height: 10px; background-color: #ccc;"></div>
PSQVLTGLR(+14.0157)	> NX_Q6F008-1	1474	A.PSRALGTLR.Q	2	<div style="width: 20px; height: 10px; background-color: #ccc;"></div>
PSQVLTGLR(+14.0157)	> NX_Q6F008-2	1284	A.PSRALGTLR.Q	2	<div style="width: 20px; height: 10px; background-color: #ccc;"></div>
PSQASVLR(+14.0157)	> NX_Q8N1W1-1	1616	D.PSQPSNVSH.K	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
PSQASVLR(+14.0157)	> NX_Q8N1W1-3	1572	D.PSQPSNVSH.K	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
PSQASVLR(+14.0157)	> NX_Q8N1W1-4	1616	D.PSQPSNVSH.K	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
PSQASVLR(+14.0157)	> NX_Q8N1W1-5	1303	D.PSQPSNVSH.K	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
PSQASVLR(+14.0157)	> NX_Q8N1W1-6	1616	D.PSQPSNVSH.K	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
PSQVSVLR *	> NX_Q8NH41-1	332	K.PSQVSVLR.N	0	<div style="width: 0px; height: 10px; background-color: #ccc;"></div>
PLN\$SVLR	> NX_Q86N98-1	107	N.PLDG\$VLLR.N	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
PVN\$SVLR(+14.0157)	> NX_Q86N98-1	107	N.PLDG\$VLLR.N	4	<div style="width: 40px; height: 10px; background-color: #ccc;"></div>
TPNVSVLR	> NX_Q9P2E9-1	89	P.APNTVTLR.E	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
TPNVSVLR	> NX_Q9P2E9-2	89	P.APNTVTLR.E	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
TPNVSVLR	> NX_Q9P2E9-3	89	P.APNTVTLR.E	3	<div style="width: 30px; height: 10px; background-color: #ccc;"></div>
PVNTSVLR	> NX_Q9UGM1-1	130	E.PVNTVTLR.Y	1	<div style="width: 10px; height: 10px; background-color: #ccc;"></div>
Non-neXtProt mappings					
PSQVSVLR *	> A0A126GWD1	308	K.PSQVSVLR.N	0	<div style="width: 0px; height: 10px; background-color: #ccc;"></div>
PSQVSVLR *	= A0A286YFH6 = ENSP00000493198.1	308	K.PSQVSVLR.N	0	<div style="width: 0px; height: 10px; background-color: #ccc;"></div>
PSQVSVLR *	= NP_001005486.1 = ENSP00000304077.4	332	K.PSQVSVLR.N	0	<div style="width: 0px; height: 10px; background-color: #ccc;"></div>

# PEFF in PeptideAtlas



Canonical  
Protein  
sequence: K.VPLFMSR.A

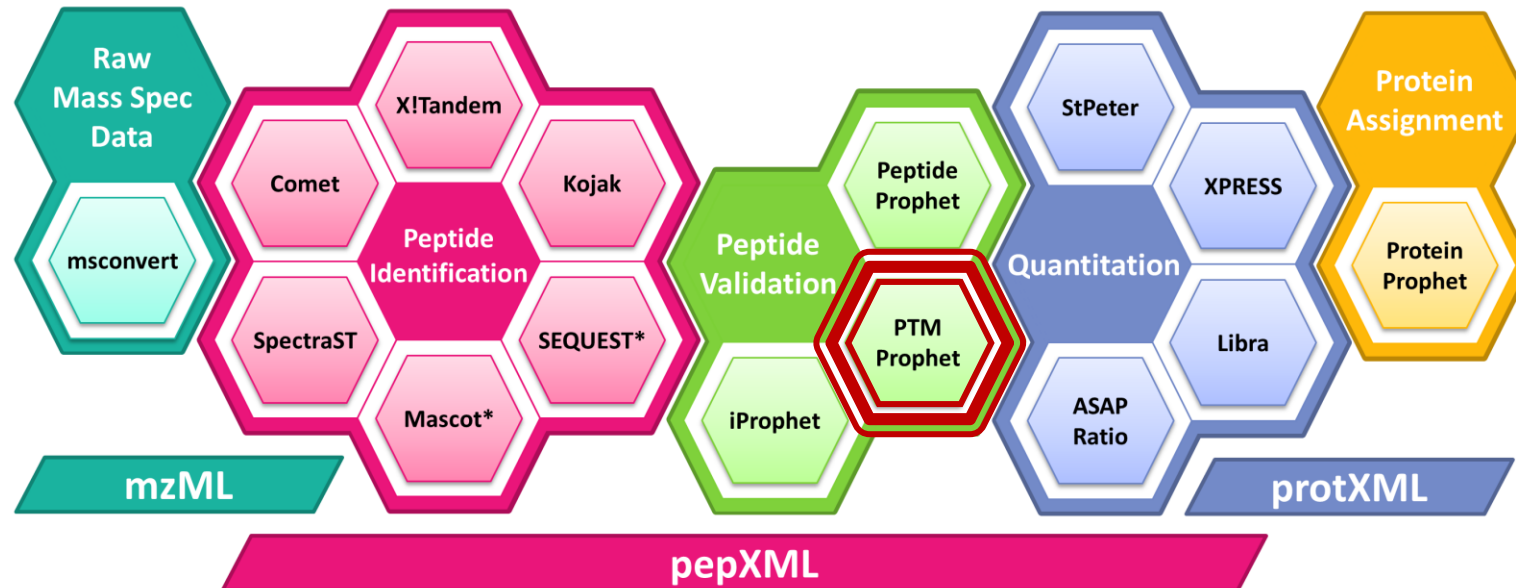
Observed  
Peptide  
sequence: K.VPLFMTR.A

Known variant  
dbSNP: rs1147990

# Agenda

- Introduction to TPP and overview of the latest features in 5.2
- Proteogenomics & PEFF applications
- **Analyzing PTMs with PTMProphet**
- Cross-linking analysis with Kojak 2.0
- Deploying the TPP using Docker containers and cloud computing platforms
- Open Discussion and Feedback

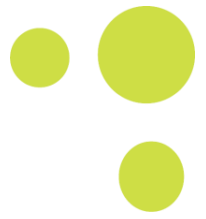
# PTMProphet: Localizing PTMs



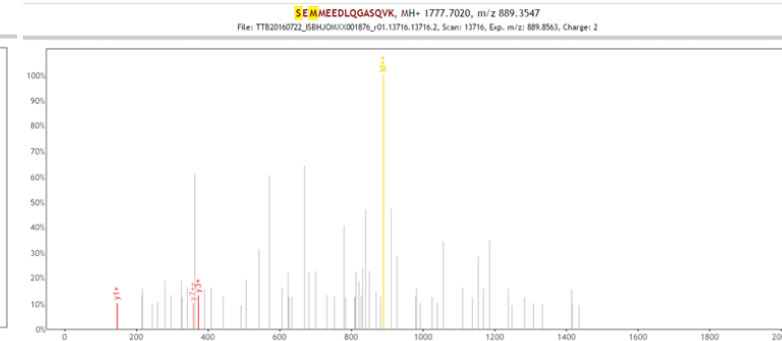
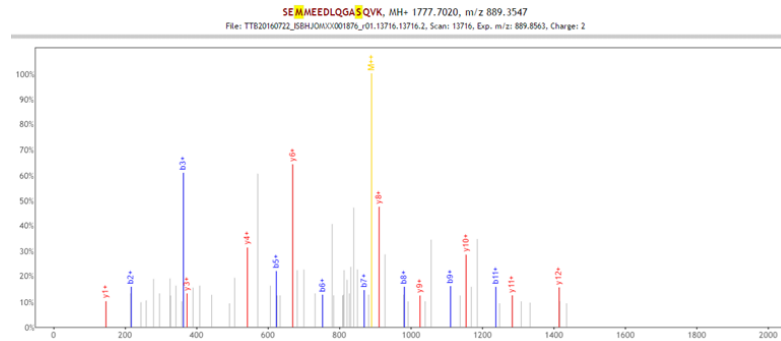
## GOALS:

1. For each Modified PSM, Compute Accurate Probabilities of each potential modification site being modified
2. Compute **Information Content** thereby allowing comparing PSMs having different numbers of potential modification sites and number of modifications

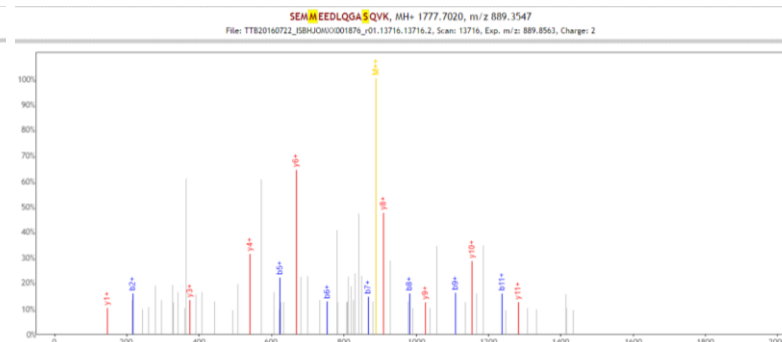
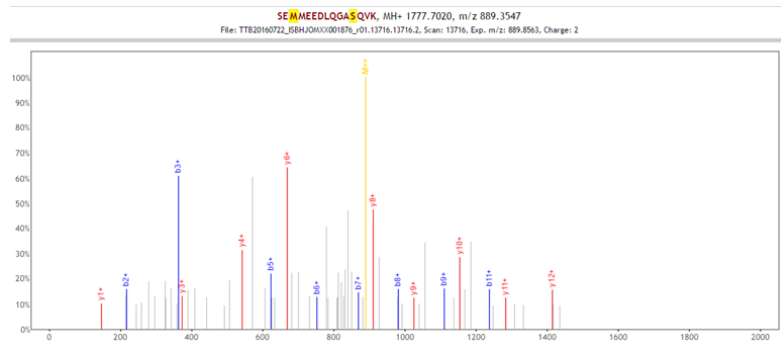
# Simple Peptide: One +16, One +80



SE**M**MEEDLQGA**S**QVK vs **S**EM**M**MEEDLQGA**S**QVK



SE**M**MEEDLQGA**S**QVK vs SEM**M**MEEDLQGA**S**QVK



# PTMProphet Algorithm



1. For each PSM, evaluate all possibilities of peptide modification
2. Sum matched peak intensities:  $\Psi(P)$  for peptide  $P$
3. Compute  $\Psi$  for each peptide possibility
4. For each potential PTM site  $s$  on the peptide, compute:  
$$P^{mod} = \operatorname{argmax}(\forall P \text{ with site } s \text{ modified} \mid \Psi(P))$$
$$P^{unmod} = \operatorname{argmax}(\forall P \text{ with site } s \text{ unmodified} \mid \Psi(P))$$

5. Compute “common” matched peak intensity:

$$C(P^{mod}, P^{unmod})$$

6. Compute discretized observed maximum noncommon intensities:

$$O^m = \frac{\Psi(P^{mod}) - C(P^{mod}, P^{unmod})}{i} \text{ and } O^u = \frac{\Psi(P^{unmod}) - C(P^{mod}, P^{unmod})}{i}$$

7. Compute observed maximum noncommon matched peaks:

$$M^m \text{ and } M^u$$

8. Compute probability for each potential PTM site
9. EM > 0 - apply expectation / maximization algorithm until probabilities remain constant
10. Normalize all probabilities by the number of modifications in the peptide
11. Record the output in pepXML

**S(0.000)EM(1.000)M(0.000)EEDLQGAS(1.000)QVK**



# PROBLEM: Site Probabilities May Not Be Comparable

- Different numbers of potentially modified sites in different peptides

SEMMEEDLQGASQVK (2 phospho sites)

SESSEEDLQGASQVK (4 phospho sites)

- Different numbers of modifications in different peptides

<sup>p</sup>SES<sup>p</sup>SEEDLQGA<sup>p</sup>SQVK (3 phospho mods)

<sup>p</sup>SESSEEDLQGASQVK (1 phospho mod)

## PTMProphet Information Content Stats:

### $H_t^{norm}$ : Multiple Modification & Site Normalized Shannon's Entropy

Quantifies the amount of information stored in the PTM site assignment for a peptide with  $s$  modification sites and  $m$  modifications, range:  $[0, 1]$

### $M_t$ : Localized Modifications Estimate

Estimates the number of modifications confidently localized that can be used to directly compare PSMs containing  $m$  modification, range:  $[0, m]$

### $I_t$ : Normalized Per-Modification Information Content

Estimates the per-modification localization certainty that can be used to directly compare PSMs with different number of modifications, range:  $[0, 1]$

# PTMProphet Site Probabilities Not Directly Comparable

- For example, site probability 0.75 implies different conclusions in different PSMs

- If  $s=4$  and  $m=3$  and

**S(0.750)ES(0.750)S(0.750)EEDLQGAS(0.750)QVK**

then,  $H_t^{norm} = 1.0$  ( $H_t = 3.0$ )

$I_t = 0.0$

$M_t = 0.0$

Maximum Uncertainty

- If  $s=4$  and  $m=1$  and

**S(0.000)ES(0.750)S(0.250)EEDLQGAS(0.000)QVK**

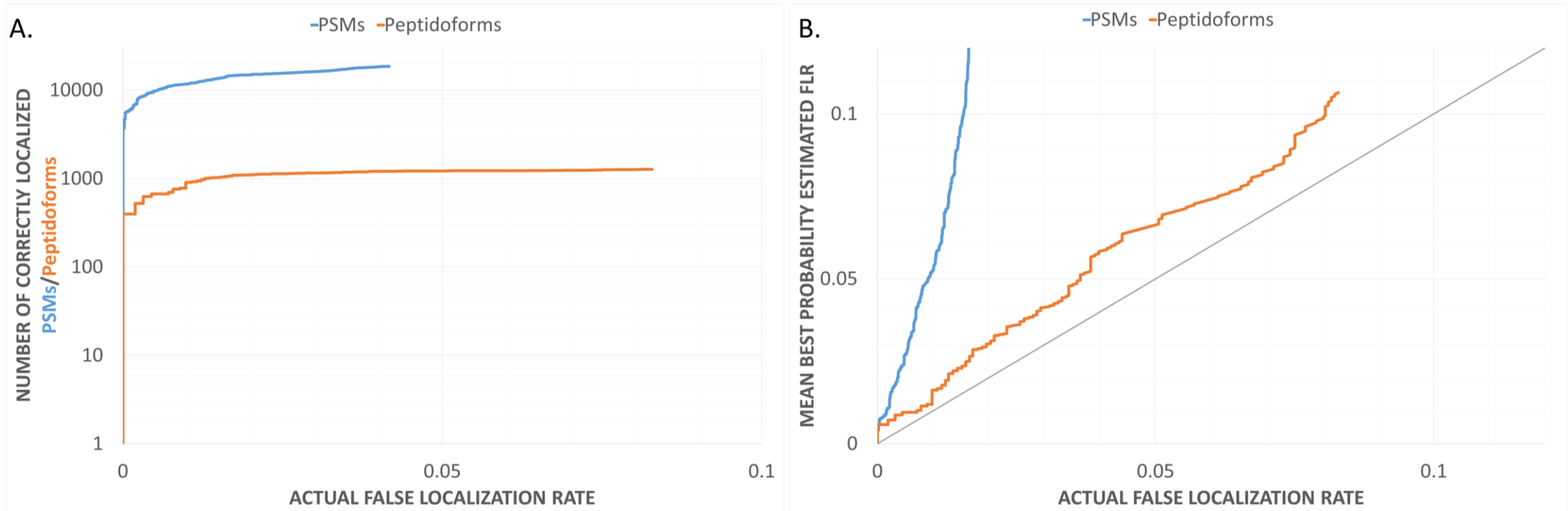
then,  $H_t^{norm} = 0.41$  ( $H_t = 0.41$ )

$I_t = 0.59$

$M_t = 0.59$

Mid-range Uncertainty

# Conservative FLR Estimates



## **$B_t$ : Mean Best Probability Statistic**

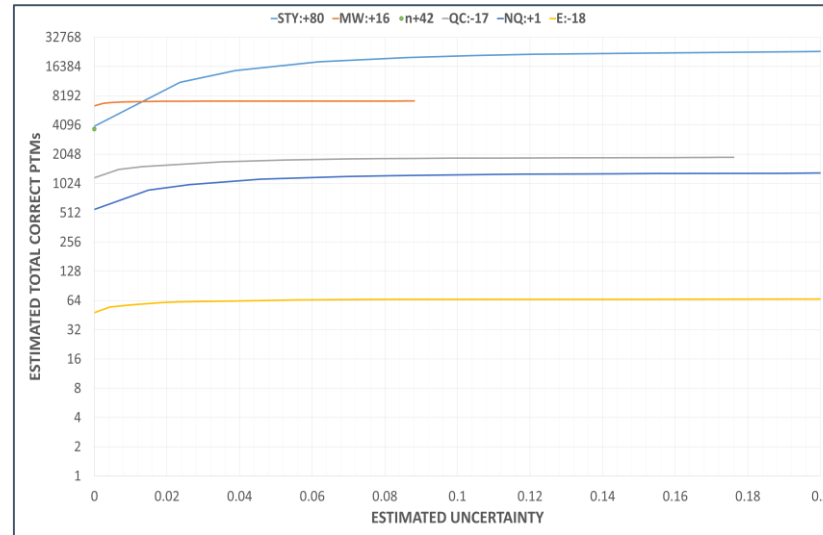
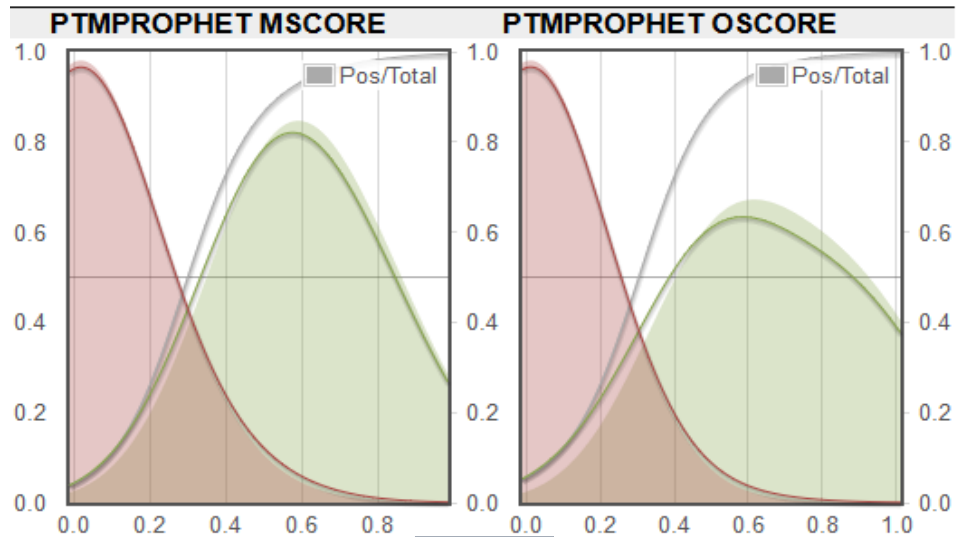
Easy to compute and works well in practice, should be considered in the context of Information Content, range:  $[0, 1]$

# PTMProphet: Wednesday Poster 400



$$Mscore = \frac{M^m}{M^m + M^u}$$

$$Oscore = \frac{O^m}{O^m + O^u}$$



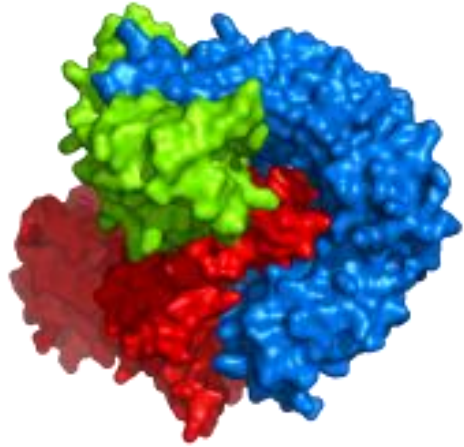
$$\frac{\Pr(Mscore, Oscore | Modified) \Pr(Modified)}{\Pr(Mscore, Oscore)}$$

$$H_t^{norm} = -\frac{1}{m} \sum_{i=1}^s p_i \log_{s/m} p_i$$

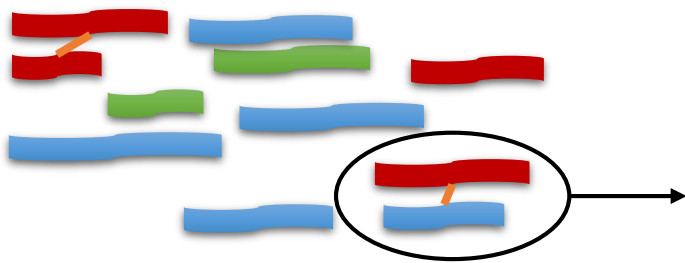
# Agenda

- Introduction to TPP and overview of the latest features in 5.2
- Proteogenomics & PEFf applications
- Analyzing PTMs with PTMProphet
- **Cross-linking analysis with Kojak 2.0**
- Deploying the TPP using Docker containers and cloud computing platforms
- Open Discussion and Feedback

# Kojak: Identifying protein interactions through cross-linking

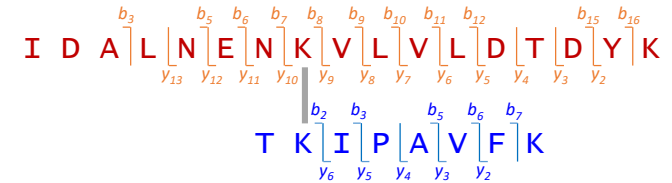
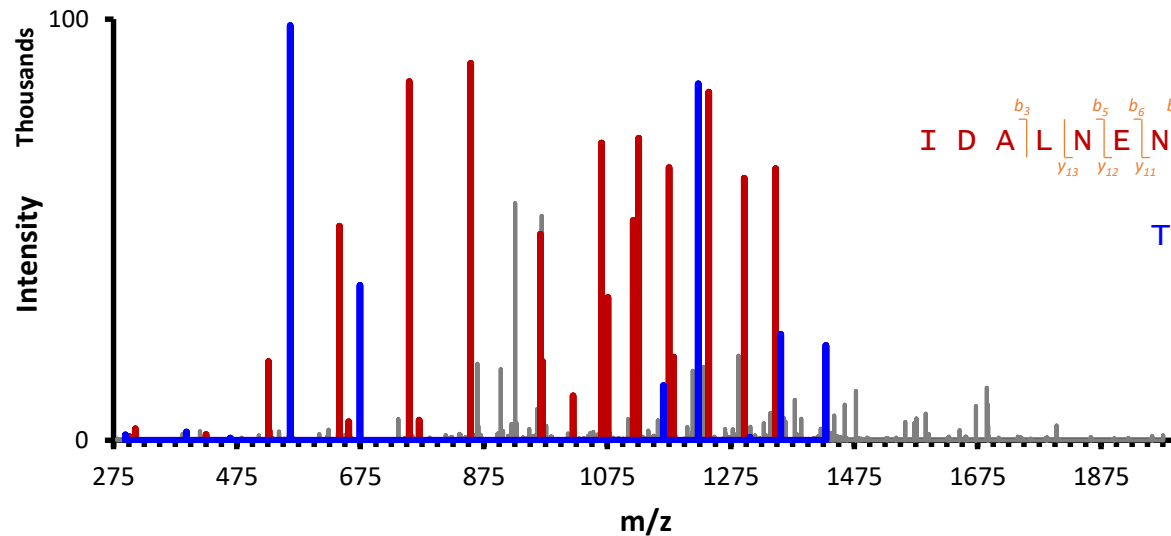


Enzymatic Digestion

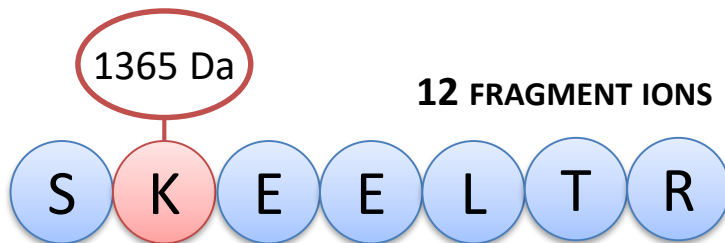
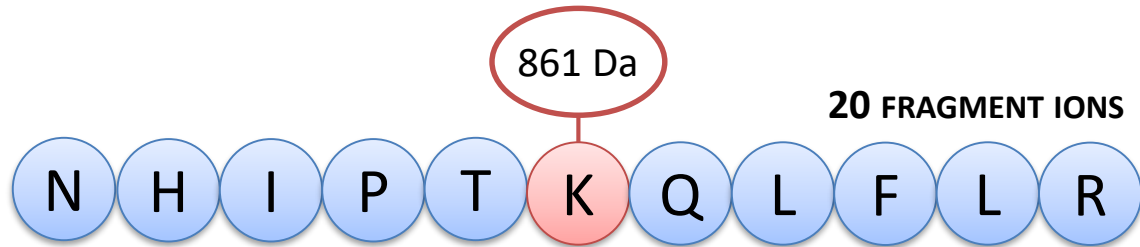
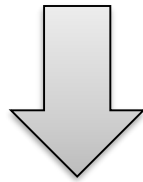
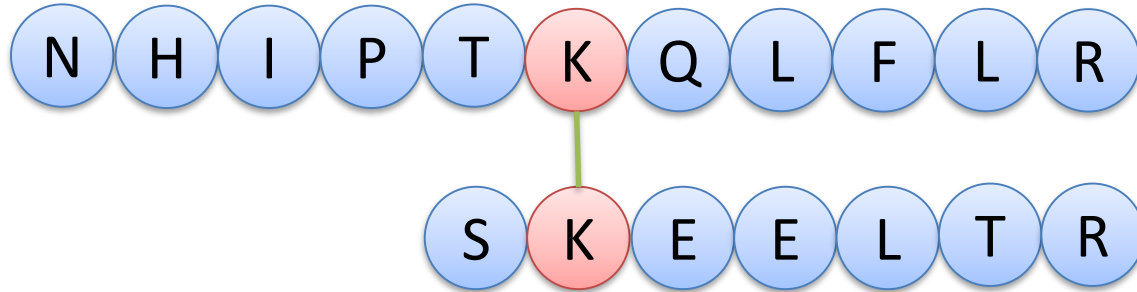
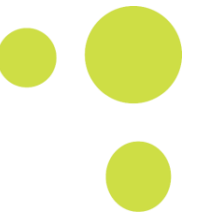


Kojak performs sequence identification of MS/MS spectra from two linked peptides.

- Can be used to infer protein structure.
- Can be used to infer protein-protein interactions.

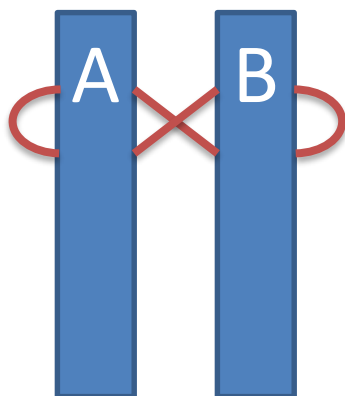


# XL Fragmentation Properties



- Consider a cross-linked peptide to be two peptides with modification masses.
- Match peptide fragment ions separately, starting with the largest (by mass) peptide.
- Largest peptide likely has highest potential score.

# Dealing With Homodimers



Intra-protein links:

$^{14}\text{N}$ - $^{14}\text{N}$  (A1-A2)

$^{14}\text{N}$ - $^{14}\text{N}$  (B1-B2)

Inter-protein links:

$^{14}\text{N}$ - $^{14}\text{N}$  (A1-B2)

$^{14}\text{N}$ - $^{14}\text{N}$  (B1-A2)

New Kojak Parameter:

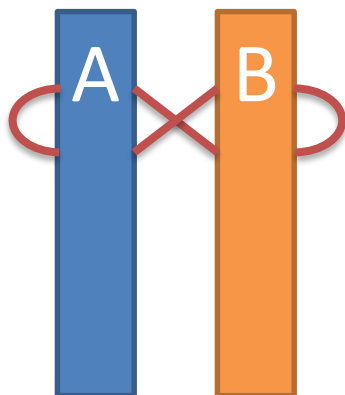
$^{15}\text{N}$ \_filter = 15n

>ProteinX

MAAAVVAATVPAQSMGADGASSVHWFRKGL  
**RS**AVV**KVK**GSRSRGQRLFKEWGVTRDAAIMKM  
 AKHLLPSLEELGFPTTEGLGPAVWQGGE

Total Mass: 729.47 Da

b <sup>+</sup>		AA		y <sup>+</sup>
88.039	1	S		
159.076	2	A	6	643.450
258.144	3	V	5	572.413
357.213	4	V	4	473.344
485.308	5	K	3	374.276
584.376	6	V	2	246.181
		K	1	147.112



Intra-protein links:

$^{14}\text{N}$ - $^{14}\text{N}$  (A1-A2)

$^{15}\text{N}$ - $^{15}\text{N}$  (B1-B2)

Inter-protein links:

$^{14}\text{N}$ - $^{15}\text{N}$  (A1-B2)

$^{15}\text{N}$ - $^{14}\text{N}$  (B1-A2)

> $^{15}\text{n}$ -ProteinX

MAAAVVAATVPAQSMGADGASSVHWFRKGL  
**RS**AVV**KVK**GSRSRGQRLFKEWGVTRDAAIMKM  
 AKHLLPSLEELGFPTTEGLGPAVWQGGE

Total Mass: 738.45 Da

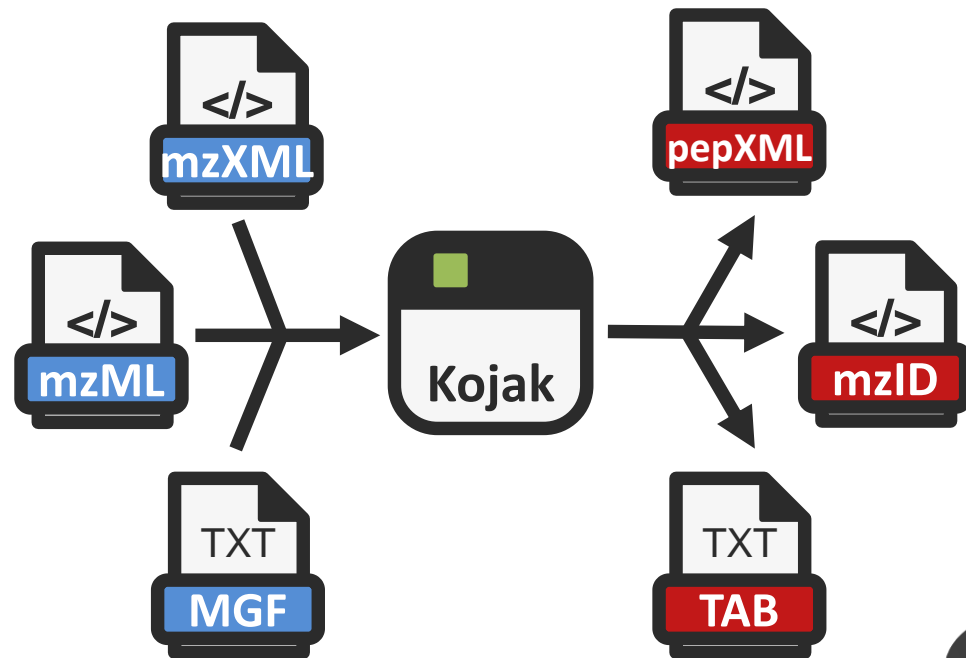
b <sup>+</sup>		AA		y <sup>+</sup>
89.036	1	S		
161.070	2	A	6	651.426
261.135	3	V	5	579.392
361.201	4	V	4	479.326
491.290	5	K	3	379.261
591.355	6	V	2	249.172
		K	1	149.106

● Natural isotope abundance

● Heavy isotope labeled

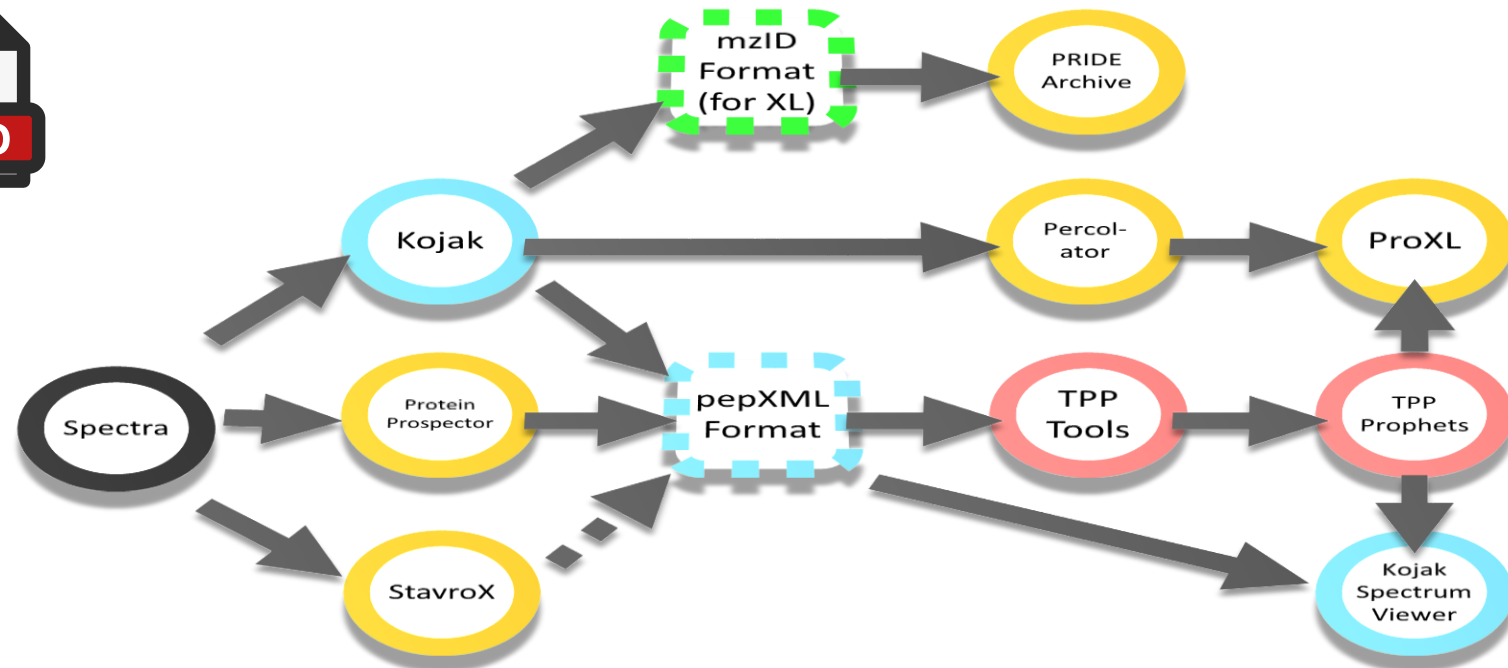


# More Supported File Formats & The Extended Kojak 2.0 Pipeline



- Novel, in-house (blue circle)
- Existing, in-house (orange circle)
- New, 3<sup>rd</sup> party (green circle)
- Existing, 3<sup>rd</sup> party (yellow circle)


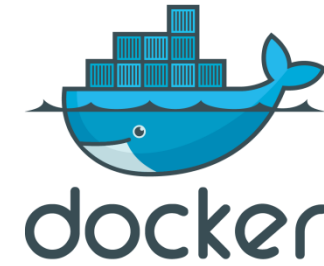
- No conversion required.
- Uses PSI open standards.



# Agenda

- Introduction to TPP and overview of the latest features in 5.2
- Proteogenomics & PEFf applications
- Analyzing PTMs with PTMProphet
- Cross-linking analysis with Kojak 2.0
- **Deploying the TPP using Docker containers and cloud computing platforms**
- Open Discussion and Feedback

# Docker Container Platform



**BioContainers**  
**7.6Ktools, 17.6Kversions, 54.8Kcontainers and packages**

[Quick Start](#) [Registry](#)

<a href="#">toppred/1.10-4-deb</a>	Add debian package toppred/1.10-4-deb/Dockerfile	a year ago
<a href="#">tpmcalculator/0.0.1</a>	remove CMD	11 days ago
<a href="#">tpp</a>	Updated for TPP 5.2 release version (#321)	a month ago
<a href="#">transdecoder/3.0.1dfsg-1-deb</a>	Add debian package transdecoder/3.0.1dfsg-1-deb/Dockerfile	a year ago
<a href="#">transrate-tools/1.0.0-1b1-deb</a>	Add debian package transrate-tools/1.0.0-1b1-deb/Dockerfile	a year ago
<a href="#">transtermhp/2.09-3b1-deb</a>	Add debian package transtermhp/2.09-3b1-deb/Dockerfile	a year ago
<a href="#">tree-ppuzzle/5.2-8b3-deb</a>	Add debian package tree-ppuzzle/5.2-8b3-deb/Dockerfile	a year ago

# Running TPP on Docker



```
% apt --yes install docker.io
```

```
% docker pull spctools/tp
```

```
% docker run -v /tmp/tpdata:/data spctools/tp comet -p
```

```
% docker run -dit --user=root -p 10401:10401 -v /tmp/tpdata:/data spctools/tp apache2ctl -DFOREGROUND
```

Running the TPP docker image

**Contents [hide]**

- 1 Overview
- 2 Installing Docker
- 3 Downloading spctools/tp docker image
- 4 Verifying that the image has been downloaded
- 5 Running a specific TPP program from the docker image
- 6 Obtaining a copy of the default parameter files
- 7 Running the TPP GUI via Docker
- 8 Checking to make sure the docker is running
- 9 Stopping the GUI when you're done
- 10 Troubleshooting Notes
- 11 Testing the container and GUI with a sample dataset (optional)

**Overview**

The docker system allows the distribution of pre-built computational units called images, which can be run on any system with the docker software installed. The spctools/tp image is built on the BioContainers ubuntu image, and uses many of the BioContainers conventions. On some systems you may have to run these commands using sudo, a mechanism for elevating user privileges for specific tasks. The initial download described below will be relatively slow the first time it is run, since most if not all the 'layers' will have to be downloaded. Subsequent invocations will generally be much faster.

**Installing Docker**

Detailed installation guidance on how to install Docker on your computer is beyond the scope of this document, but if you're using Ubuntu 18.04 LTS, it should be as easy as:

```
sudo apt --yes install docker.io
```

**Downloading spctools/tp docker image**

As mentioned above, this may be slow the first time it is run. Subsequent runs should be considerably faster. The example below shows a pull of the latest (default) version, you can also specify a specific version if desired.

```
docker pull spctools/tp
Using default tag: latest
Trying to pull repository docker.io/spctools/tp ...
latest: Pulling from docker.io/spctools/tp
Digest: sha256:37220874949d6f710eec6661ef57ec2b3cc42f9066d1496a283c4eb32387e4da
Status: Image is up to date for spctools/tp:latest
```

Find tutorials at: <http://tppms.org/tutorial>

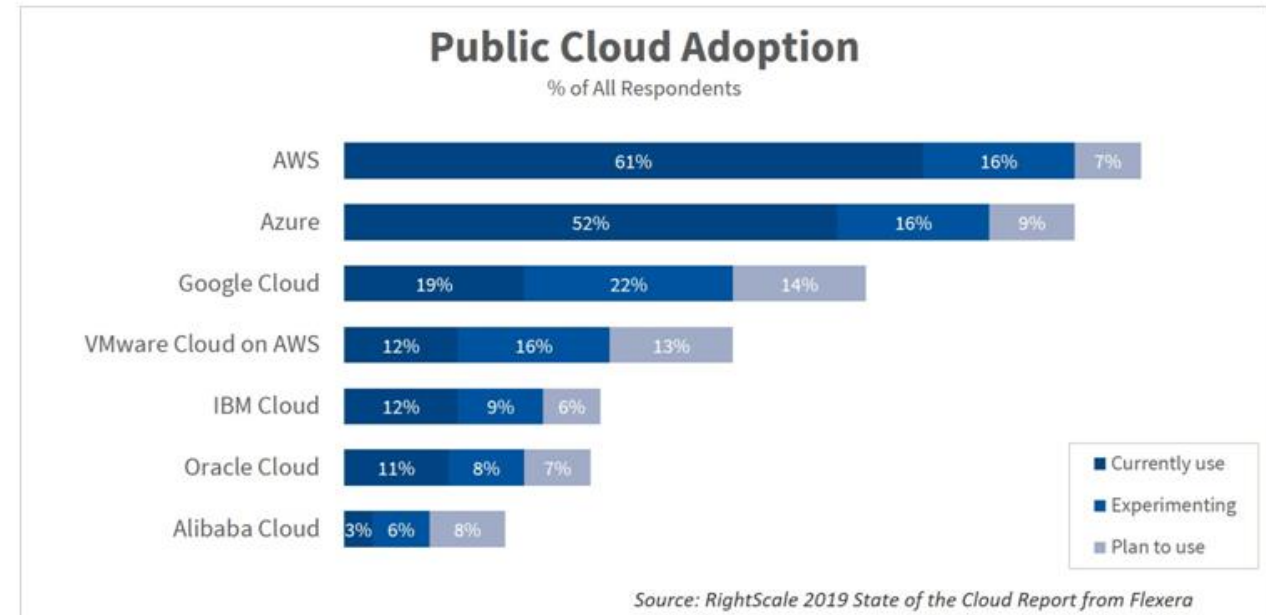
# Cloud Computing Platforms



Figure 1. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide



Source: Gartner (May 2018)



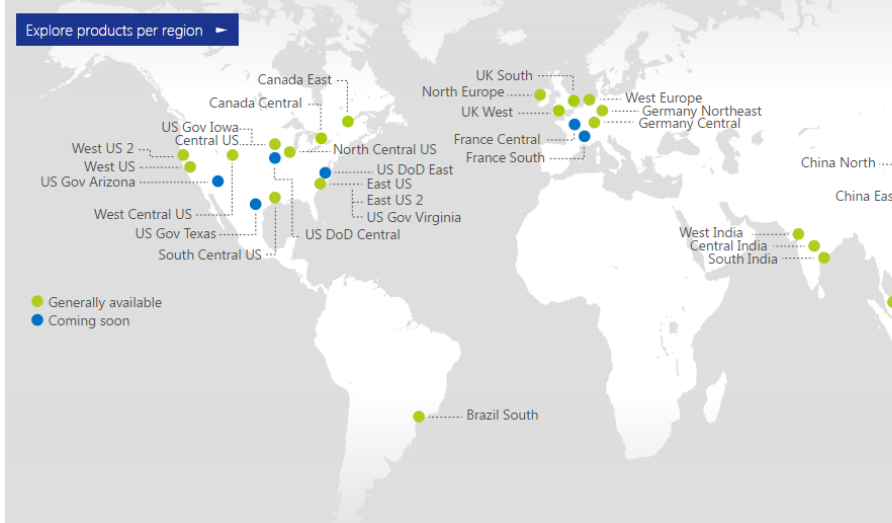
# Microsoft Azure

Microsoft Azure SALES 1-800-967-1389 MY ACCOUNT PORTAL Search

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support [FREE ACCOUNT >](#)

## Azure regions

Azure is generally available in 30 regions around the world, and has announced plans for 8 additional regions. Geographic expansion is a priority for Azure because it enables our customers to achieve higher performance and it support their requirements and preferences regarding data location.



Microsoft Azure Virtual machines > Compute > Ubuntu Server Search resources

### Compute

Filter Search Compute

#### MongoDB with Replication

Bitnami

Scale above two VMs using this Solution Template. MongoDB with Replication gives you the ability to allow individual vertical scaling of nodes, break out configuration into two nodes for replication & backups, and more with just a single click.

Create

#### Recommended

- Windows Server (Microsoft)
- Red Hat Enterprise Linux (RedHat)
- Ubuntu Server (Canonical)**
- SQL Server 2016 SP1 Enterprise on (Microsoft)
- Virtual machine scale set (Microsoft)
- Azure Container Service (Microsoft)

#### Virtual Machine Images

- Dynamics AX 2012 R3 (preview) (Microsoft)
- Drupal (Bitnami)
- VoipNow 3.6.0 (4PSA)
- 4ward365 Azure Client (4ward)
- SiouxApp Business Suite (7iSolutions)
- Ubuntu Server 12.04.5 LTS (Canonical)

#### Ubuntu Server

- Ubuntu Server 12.04.5 LTS Canonical
- Ubuntu Server 14.04 LTS Canonical
- Ubuntu Server 16.04 LTS Canonical
- Ubuntu Server 16.10 Canonical


# TPP on the Cloud



Find tutorials at: <http://tppms.org/tutorials>

article discussion edit history

## TPP 5.2.0: Installing on Ubuntu 18.04 LTS

 Note that this recipe will likely not be exactly applicable to different versions of TPP or different distributions of Linux, but perhaps can be a useful head start. For other versions of TPP, please see the README and INSTALL\_LINUX files found in source code distribution for instructions on how to build TPP for the Linux platform.

**Contents** [hide]

- 1 Before we begin
- 2 Installation
  - 2.1 Installing prerequisite packages
  - 2.2 Creating a suitable place to compile and install
  - 2.3 Pulling the TPP 5.2.0 source code from SourceForge
  - 2.4 Compiling the source code
  - 2.5 Configuring the Apache web server
- 3 Testing the installation with a sample dataset (optional)

### Before we begin

This recipe describes installing TPP release 5.2.0.

First some general notes

- We assume a fresh installation of Ubuntu 18.04 LTS as a starting point (LTS stands for Long Term Support).
- This recipe was tested on a fresh Azure cloud computing platform instance running Ubuntu 18.04 LTS.
- We assume that there is an ordinary user account named 'tpp' that will be used as the user for the installation.
- This installation assumes you are installing to `/usr/local/tpp`. If you want to install to a different location, you should modify the `INSTALL_LINUX` file.
- This installation assumes you will make your data directory `/data`. If you want to make a different directory, you should modify the `INSTALL_LINUX` file.
- This installation assumes you are using Apache 2.4 as your web server.
- The gray blocks are commands to be typed in at the shell directly. You should be able to copy and paste them.

### Installation

#### Installing prerequisite packages

Use the ubuntu package manager to upgrade any outdated packages and then install all the prerequisite packages.

```
sudo apt update
sudo apt --yes upgrade
sudo apt --yes install subversion
sudo apt --yes install build-essential
sudo apt --yes install perl
```

## ISB/SPC Trans Proteomic Pipeline ::

# Petunia

PLEASE LOG IN TO USE TPP AND RELATED TOOLS

User Name:

Password:

Login

TPP v5.2.0 Flammagenitus, Build 201903272055-7901 (Linux-x86\_64)

# TPP on Amazon Web Services



## TPP Web Application (TWA)

- Simple web based launcher to start petunia on a Amazon server
- Starts up an pre-configured TPP instance
- Doesn't require any software installation and is inexpensive to run
- Great tool for just trying out TPP
- Can be used when memory and better CPU is needed for an analysis



## TPP Amazon command line tools (amztp)

- Advanced command line toolset
- Launches parallel searches of files across multiple nodes
- Currently supports X!Tandem, OMSSA, MyriMatch, InsPect
- Manage all aspects of cloud computing including data transfer, scheduling, and instances
- Great for quickly and inexpensively processing large amounts of data



**Direct Cloud support in TPP's User Interface, Petunia**



# Discussion

- Are you currently using Docker to run proteomics tools?
- Are you currently using cloud computing?
- Using the TPP on the cloud?
- Experiences?
- Is there an aspect that we should add?
- Are you writing cloud computing funds into your grant budgets?

# Agenda

- Introduction to TPP and overview of the latest features in 5.2
- Proteogenomics & PEFF applications
- Analyzing PTMs with PTMProphet
- Cross-linking analysis with Kojak 2.0
- Deploying the TPP using Docker containers and cloud computing platforms
- **Open Discussion and Feedback**



# Group presentations at ASMS

For more info:

- Kojak 2.0 (MP 044)
- PEFF in TPP (MP 417)
- PTMPProphet (WP 400)

## Booth #118

<http://www.tppms.org/asms/>