Contents

I. SUPPLEMENTARY NOTE 1: OVERVIEW OF OPENMS

A. OpenMS Software

OpenMS uses modern software engineering concepts more commonly found in industry settings than in academic environments. The project places great emphasis on modularity, reusability and extensive testing (using continuous integration) leading to high code quality. The modular architecture of OpenMS tries to build upon existing standard libraries as much as possible, relying on them for sequence analysis, XML parsing, numerical computations and statistics.

The library itself contains over 1300 C++ classes representing core concepts in mass spectrometry and the corresponding ontologies defined by the Human Proteome Organization Proteomics Standard Initiative (HUPO-PSI) [1]. Modern object-oriented, template-based C++ is used exclusively throughout the code base, encapsulating raw data structures and discouraging manual heap-based memory management, thus providing robust and error-tolerant code. Coding conventions are enforced and extensive English documentation is available for several thousand C++ functions part of the public API.

All development is performed in the open, using a public source code repository and ticketing system. Stringent code reviews and continuous integration, running a multitude of functional, unit and black-box tests, ensure continued support, robustness and correctness of the code. Strict abstraction is used to hide implementation details from the user and support classes tailored to specific algorithms.

B. Improvements since OpenMS 1.0

When the first version of OpenMS was released in 2007, it was a pure C++ software library [2] and thus targeting software developers exclusively. Since then, the project has fundamentally changed in many respects and has grown far beyond a software library. In 2016, OpenMS has become a flexible collection of tools, workflows and algorithms with a strong focus on end-users and the community of MS practitioners. The recent 2.0 release of OpenMS consists of over 180 executable end-user tools fully integrated into graphical workflow managers (such as Galaxy or KNIME). The resulting workflows can be executed on computing infrastructure as diverse as laptop computers and high-performance computing

clusters. Over the last decade, the OpenMS project has also expanded in scope and functionality, adding algorithms for high-resolution shotgun proteomics, targeted proteomics, isotopic labelling, cross-linking, metaproteomics, and metabolomics analysis while providing a new Python interface to gain access to additional developers. In comparison to the initial C++ software library of limited scope, OpenMS is now a community-driven initiative which additionally provides teaching outreach, scientific collaboration and dissemination of open source software and is a major driver for the adaption of open standards and transparent, open-source software in computational mass spectrometry. The OpenMS project has thus matured into an algorithmic cornerstone for a large number of data analysis workflows in the field and has been recognized by the community as an easy-to-use, flexible tool for all mass spectrometric data analysis needs.

C. OpenMS resources

The OpenMS project provides an extensive set of resources which can obtained online. Table I displays an overview over these resources, starting with the OpenMS webpage, the binary downloads of the software and the user tutorials which explain the major features of OpenMS to a new user.

D. OpenMS pre-built workflows

The OpenMS project contains several pre-built workflows which can be accessed at `http://www.openms.de/workflows` and are further described in Table II.

TABLE I: Available resources from the OpenMS project

Resource	Description	URL
Web page	OpenMS home page	http://open-ms.de/
Download	Download page to obtain OpenMS binaries	http://open-ms.de/downloads/
Tutorials	User-oriented tutorials for OpenMS	https://github.com/Tutorials
Workflows	Workflows for OpenMS	http://www.openms.de/workflows
Mailing List	Public mailing list for questions relating to the use and features of OpenMS	https://lists.sourceforge.net/lists/listinfo/open-ms-general/
Source code	Complete source code of the OpenMS project	https://github.com/OpenMS/OpenMS
Documentation	Complete documentation of all public classes and functions of OpenMS, including build instructions, developer guide	http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/index.html
Bug tracker	List of currently known bugs and missing features (open to the public and preferred way of reporting issues with OpenMS)	https://github.com/OpenMS/OpenMS/issues
KNIME Forum	OpenMS section of the KNIME Forum which gives advice to issues encountered with the KNIME nodes	https://tech.knime.org/forum/openms

TABLE II: Available workflows in OpenMS

Workflow	Engine	Description	Ref.
Peptide Identification	KNIME	Basic peptide identification	
Peptide Identification	KNIME	Consensus peptide identification with multiple search engines	
Labelfree	KNIME	Label free quantification workflow suitable for large scale experiments	[3]
Labelfree	KNIME	Label free quantification and identification workflow	[3]
Protein Inference	KNIME	Label free quantification, identification and protein inference	
OpenSWATH	KNIME	SWATH MS analysis workflow	[4]
Metabolomics	KNIME	Small molecule identification and quantification	[5]
Isobaric label	TOPPAS	iTRAQ quantification and identification	

E. OpenMS supported data formats

The OpenMS project supports a large set of over 25 different data formats, which can be read by OpenMS tools and converted to more common data formats (see Table III). The supported formats range from a large number of data formats for raw spectral data to spectral library data formats to HUPO PSI standard formats for identification and quantification information. This allows OpenMS to support a large number of different workflows, integrate with different tools or act as intermediate tools to convert from a specific format to a desired data format.

In addition, OpenMS supports over 15 different external tools directly which allows the user in many cases to execute a specific external tool directly from within the OpenMS workflow environment. OpenMS is capable of writing input data and reading output format for the majority of these tools, as indicated in Table IV.

TABLE III: Supported data formats in OpenMS

Data format	Support	Tool support[a]
`.mzML`	Full	Tools reading and writing raw data
`.featureXML`	Full	Tools reading and writing feature data
`.consensusXML`	Full	Tools reading and writing feature data
`.ini`	Full	All tools
`.toppas`	Full	TOPPAS workflow manager
`.trafoXML`	Full	All relevant tools
`.idXML`	Full	All tools reading and writing identification data
`.mzQuantML`	Full	IsobaricAnalyzer, FeatureFinder tools
`.TraML`	Full	All OpenSWATH tools, conversion tools
`.qcML`	Full	QC tools
`.mzTab`	Write	All metabolite tools, MzTabExporter
`.mzIdentML`	Read and Write	IDFileConverter
`.pep.xml`	Read and Write	IDFileConverter and various adapters
`.prot.xml`	Read	IDFileConverter
`.peplist`	Read	FileConverter
MSPFile	Read and Write	SpecLibCreator, SpecLibSearcher
`.mzXML`	Read and Write	FileConverter
`.mzData`	Read and Write	FileConverter
`.mgf`	Read and Write	FileConverter
`.dta`	Read and Write	FileConverter
`.dta2d`	Read and Write	FileConverter
`.edta`	Read and Write	FileConverter
`.fasta`	Read and Write	multiple
`.obo`	Read	CVInspector, SemanticValidator
CV Mapping (`.xml`)	Read	CVInspector, SemanticValidator

[a]Some file formats can only be used in OpenMS after conversion to a suitable format, e.g. `.mzXML` has to be converted to the current standard format `.mzML` before further processing.

TABLE IV: Supported software and associated formats in OpenMS

Software tool	Support	Direct Execution[a]	Tool support
Fido	Input and Output	Yes	FidoAdapter
InsPecT	Input and Output	Yes	InspectAdapter
Luciphor	Input and Output	Yes	LuciphorAdapter
Mascot	Input and Output	Yes	MascotAdapter, MascotAdatpterOnline
MSGFPlus	Input and Output	Yes	MSGFPlusAdapter
MyriMatch	Input and Output	Yes	MyriMatchAdapter
OMSSA	Input and Output	Yes	OMSSAAdapter
PepNovo	Input and Output	Yes	PepNovoAdapter
XTandem	Input and Output	Yes	XTandemAdapter
Percolator	Input and Output	Yes	TopPerc
TPP	Input and Output	No	various (pep and prot xml)
SpetraST	Read	No	ConvertTSVToTraML (MRM file format)
Hardkloer	Read	No	FileConverter (Kroenik file format)
UniMod XML	Read	N/A	multiple
SEQUEST	Input and Output	No	IDFileConverter

[a]Direct execution means that these tools can be directly incorporated into an OpenMS workflow. For all other cases, OpenMS is able to read the output of the tool after separate execution.

II. SUPPLEMENTARY NOTE 2: AVAILABLE TOOLS

The following listing provides an overview of the available tools in OpenMS with a short description of their function. In total, over 180 individual tools are available to the user, please see the online listing of the TOPP tools and the UTILS for and up-to-date list of available tools.

A. Graphical Tools

- TOPPView - A viewer for mass spectrometry data.
- TOPPAS - An assistant for GUI-driven TOPP workflow design.
- INIFileEditor - An editor for OpenMS configuration files.

B. File Handling

- DTAExtractor - Extracts spectra of an MS run file to several files in DTA format.
- FileConverter - Converts between different MS file formats.
- FileFilter - Extracts or manipulates portions of data from peak, feature or consensus feature files.
- FileInfo - Shows basic information about the file, such as data ranges and file type.
- FileMerger - Merges several MS files into one file.
- IDMerger - Merges several protein/peptide identification files into one file.
- IDRipper - Splits protein/peptide identifications according their file-origin.
- IDFileConverter - Converts identification engine file formats.
- MapStatistics - Extract extended statistics on the features of a map for quality control.
- TextExporter - Exports various XML formats to a text file.
- MzTabExporter - Exports various XML formats to an mzTab file

C. Signal Processing and Preprocessing

- BaselineFilter - Removes the baseline from profile spectra using a top-hat filter.
- InternalCalibration - Applies an internal calibration.
- MapNormalizer - Normalizes peak intensities in an MS run.
- MassTraceExtractor - Annotates mass traces in centroided LC-MS maps.
- NoiseFilterGaussian - Removes noise from profile spectra by using different smoothing techniques.
- NoiseFilterSGolay - Removes noise from profile spectra by using different smoothing techniques.
- PeakPickerHiRes - Finds mass spectrometric peaks in profile mass spectra.
- PeakPickerWavelet - Finds mass spectrometric peaks in profile mass spectra.
- PrecursorMassCorrector - Correct the precursor entries of tandem MS scans.
- HighResPrecursorMassCorrector - Correct the precursor entries of tandem MS scans.
- Resampler - Transforms an LC-MS map into an equally-spaced (in RT and m/z) map.
- SpectraFilterBernNorm - Applies a filter to peak spectra.
- SpectraFilterMarkerMower - Applies a filter to peak spectra.
- SpectraFilterNLargest - Applies a filter to peak spectra.
- SpectraFilterNormalizer - Applies a filter to peak spectra.
- SpectraFilterParentPeakMower - Applies a filter to peak spectra.
- SpectraFilterScaler - Applies a filter to peak spectra.
- SpectraFilterSqrtMower - Applies a filter to peak spectra.
- SpectraFilterThresholdMower - Applies a filter to peak spectra.

- SpectraFilterWindowMower - Applies a filter to peak spectra.
- SpectraMerger - Merges spectra from an LC-MS map, either by precursor or by RT blocks
- TOFCalibration - Applies time of flight calibration.
- RNPxlXICFilter - Remove MS2 spectra from treatment based on the fold change between control and treatment for RNP cross linking experiments.

D. Quantitation

- AdditiveSeries - Computes an additive series to quantify a peptide in a set of samples.
- Decharger - Decharges and merges different feature charge variants of the same chemical entity.
- EICExtractor - Quantifies signals at given positions in (raw or picked) LC-MS maps.
- FeatureFinderCentroided - Detects two-dimensional features in centroided LC-MS data.
- FeatureFinderIdentification - Detects two-dimensional features in MS1 data based on peptide identifications.
- FeatureFinderIsotopeWavelet - Detects two-dimensional features in uncentroided (=raw) LC-MS data.
- FeatureFinderMetabo - Detects two-dimensional features in centroided LC-MS data of metabolites.
- FeatureFinderMRM - Quantifies features LC-MS/MS MRM data.
- FeatureFinderMultiplex - Identifies peptide multiplets (pairs, triplets etc.) and determines their relative abundance.
- IsobaricAnalyzer - Extracts and normalizes TMT and iTRAQ information from a MS experiment.

- ITRAQAnalyzer - Extracts and normalizes iTRAQ information from an MS experiment.
- ProteinQuantifier - Computes protein abundances from annotated feature/consensus maps
- ProteinResolver - A peptide-centric algorithm for protein inference.
- SeedListGenerator - Generates seed lists for feature detection.
- TMTAnalyzer - Extracts and normalizes TMT information from an MS experiment.

E. Map Alignment

- ConsensusMapNormalizer - Normalizes maps of one consensusXML file (after linking).
- MapAlignerIdentification - Corrects retention time distortions between maps based on common peptide identifications.
- MapAlignerPoseClustering - Corrects retention time distortions between maps using a pose clustering approach.
- MapAlignerSpectrum - Corrects retention time distortions between maps by spectrum alignment.
- MapRTTransformer - Applies retention time transformations to maps.
- FeatureLinkerLabeled - Groups corresponding isotope-labeled features in a feature map.
- FeatureLinkerUnlabeled - Groups corresponding features from multiple maps.
- FeatureLinkerUnlabeledQT - Groups corresponding features from multiple maps using a QT clustering approach.

F. Protein/Peptide Identification

- CompNovo - Performs a peptide/protein identification with the CompNovo engine.

- CompNovoCID - Performs a peptide/protein identification with the CompNovo engine in CID mode.
- InspectAdapter - Identifies MS/MS spectra using Inspect (external).
- MascotAdapter - Identifies MS/MS spectra using Mascot (external).
- MascotAdapterOnline - Identifies MS/MS spectra using Mascot (external).
- MSGFPlusAdapter - Identifies MS/MS spectra using MSGFPlus (external).
- MyriMatchAdapter - Identifies MS/MS spectra using MyriMatch (external).
- OMSSAAdapter - Identifies MS/MS spectra using OMSSA (external).
- PepNovoAdapter - Identifies MS/MS spectra using PepNovo (external).
- XTandemAdapter - Identifies MS/MS spectra using XTandem (external).
- SpecLibSearcher - Identifies peptide MS/MS spectra by spectral matching with a searchable spectral library.

G. Protein/Peptide Processing

- ConsensusID - Computes a consensus identification from peptide identifications of several identification engines.
- FalseDiscoveryRate - Estimates the false discovery rate on peptide and protein level using decoy searches.
- FidoAdapter - Runs the protein inference engine Fido.
- IDConflictResolver - Resolves ambiguous annotations of features with peptide identifications.
- IDFilter - Filters results from protein or peptide identification engines based on different criteria.
- IDMapper - Assigns protein/peptide identifications to feature or consensus features.

- IDPosteriorErrorProbability - Estimates posterior error probabilities using a mixture model.
- IDRTCalibration - Can be used to calibrate RTs of peptide hits linearly to standards.
- LuciphorAdapter - Scores potential phosphorylation sites in order to localize the most probable sites.
- PeptideIndexer - Refreshes the protein references for all peptide hits.
- PhosphoScoring - Scores potential phosphorylation sites in order to localize the most probable sites.
- ProteinInference - Infer proteins from a list of (high-confidence) peptides.
- DecoyDatabase - Create decoy peptide databases from normal ones.
- Digestor - Digests a protein database in-silico.
- DigestorMotif - Digests a protein database in-silico (optionally allowing only peptides with a specific motif) and produces statistical data for all peptides.
- IDExtractor - Extracts n peptides randomly or best n from idXML files.
- IDMassAccuracy - Calculates a distribution of the mass error from given mass spectra and IDs.
- IDScoreSwitcher - Switches between different scores of peptide or protein hits in identification data.
- RNPxl - Tool for RNP cross linking experiment analysis.
- SequenceCoverageCalculator - Prints information about idXML files.
- SpecLibCreator - Creates an MSP-formatted spectral library.

H. Targeted Experiments

- InclusionExclusionListCreator - Creates inclusion and/or exclusion lists for LC-MS/MS experiments.

- PrecursorIonSelector - A tool for precursor ion selection based on identification results.
- MRMMapper - MRMMapper maps measured chromatograms (mzML) and the transitions used (TraML)
- OpenSwathDecoyGenerator - Generates decoys according to different models for a specific TraML
- OpenSwathChromatogramExtractor - Extract chromatograms (XIC) from a MS2 map file.
- OpenSwathAnalyzer - Picks peaks and finds features in an SRM experiment.
- OpenSwathRTNormalizer - This tool will align an SRM / SWATH run to a normalized retention time space.
- OpenSwathFeatureXMLToTSV - Converts a featureXML to a tsv.
- OpenSwathConfidenceScoring - Computes confidence scores for OpenSwath results.

I. Peptide Property Prediction

- PTModel - Trains a model for the prediction of proteotypic peptides from a training set.
- PTPredict - Predicts the likelihood of peptides to be proteotypic using a model trained by PTModel.
- RTModel - Trains a model for the retention time prediction of peptides from a training set.
- RTPredict - Predicts retention times for peptides using a model trained by RTModel.

J. File Handling

- CVInspector - A tool for visualization and validation of PSI mapping and CV files.
- ConvertTSVToTraML - Converts a tsv file (tab separated) to TraML.

- ConvertTraMLToTSV - Converts a TraML file to TSV.
- FuzzyDiff - Compares two files, tolerating numeric differences.
- IDSplitter - Splits protein/peptide identifications off of annotated data files.
- MzMLSplitter - Splits an mzML file into multiple parts
- OpenSwathMzMLFileCacher - Caching of large mzML files
- SemanticValidator - SemanticValidator for analysisXML and mzML files.
- XMLValidator - Validates XML files against an XSD schema.

K. Algorithm Evaluation

- FFEval - Evaluation tool for feature detection algorithms.
- IDEvaluator - Evaluation tool, comparing peptide recovery at different q-value thresholds for multiple search engines (e.g., after ConsensusID). For interactive version use the IDEvaluatorGUI tool.
- LabeledEval - Evaluation tool for isotope-labeled quantitation experiments.
- MapAlignmentEvaluation - Evaluates alignment results against a ground truth.
- RTEvaluation - Application that evaluates TPs (true positives), TNs, FPs, and FNs for an idXML file with predicted RTs.
- TransformationEvaluation - Simple evaluation of transformations (e.g. RT transformations produced by a MapAligner tool).

L. Quantitation

- ERPairFinder - Evaluate pair ratios on enhanced resolution (zoom) scans.
- FeatureFinderSuperHirn - Find Features using the SuperHirn Algorithm (it can handle centroided or profile data, see .ini file).

- MRMPairFinder - Evaluate labeled pair ratios on MRM features.
- OpenSwathWorkflow - Complete workflow to run OpenSWATH.

M. Metabolite Identification

- AccurateMassSearch - Find potential HMDB IDs within the given mass error window.

N. Quality Control

- QCCalculator - Calculates basic quality parameters from MS experiments and compiles data for subsequent QC into a qcML file.
- QCEmbedder - This application is used to embed tables or plots generated externally as attachments to existing quality parameters in qcML files.
- QCExporter - Will extract several quality parameter from several run/sets from a qcML file into a tabular (text) format - counterpart to QCImporter.
- QCExtractor - Extracts a table attachment of a given quality parameter from a qcML file as tabular (text) format.
- QCImporter - Will import several quality parameter from a tabular (text) format into a qcML file - counterpart to QCExporter.
- QCMerger - Merges two qcML files together.
- QCShrinker - This application is used to remove extra verbose table attachments from a qcML file that are not needed anymore, e.g. for a final report.

O. Misc

- GenericWrapper - Allows the generic wrapping of external tools.
- ExecutePipeline - Executes workflows created by TOPPAS.
- INIUpdater - Update INI and TOPPAS files from previous versions of OpenMS as parameters and storage method might have changed

- DeMeanderize - Orders the spectra of MALDI spotting plates correctly.
- ImageCreator - Creates images from MS1 data (with MS2 data points indicated as dots).
- MSSimulator - A highly configurable simulator for mass spectrometry experiments.
- MassCalculator - Calculates masses and mass-to-charge ratios of peptide sequences.
- OpenSwathDIAPreScoring - SWATH (data-independent acquisition) pre-scoring.
- OpenSwathRewriteToFeatureXML - Rewrites results from mProphet back into featureXML.
- SvmTheoreticalSpectrumGeneratorTrainer - A trainer for SVM models as input for SvmTheoreticalSpectrumGenerator.

III. SUPPLEMENTARY NOTE 3: CASE STUDIES

A. SWATH Analysis

Large scale proteomics measurements of human blood plasma provide in-depth information about inter- and intra-person variability of protein abundances and can be used to investigate longitudinal trends during aging. In a recent study, Liu et al. [6] used SWATH-MS to acquire over 200 blood plasma samples collected from monozygotic and dizygotic twins in a longitudinal study. For maximal sensitivity, the authors chose a targeted analysis strategy for SWATH-MS data [7] which was developed using the OpenMS software library. In particular, OpenMS enabled the development of an analysis tool using existing algorithms and data structures for file parsing, RT alignment and signal processing. Second, the extensibility of OpenMS allowed the creation of a new OpenSWATH module for the targeted analysis of SWATH-MS data, built on top of the C++ library [4]. This module is now an integral analysis tool in OpenMS that executes all steps of the targeted analysis algorithm (RT alignment, targeted extraction, peak detection and peak scoring). When applied in the study of Liu et al., the high sensitivity of OpenSWATH led to the quantification of over 300 plasma proteins and allowed the authors the decomposition of the observed variance for each protein into heritable, environmental (common and individual) and longitudinal components. Interestingly, the authors found that the heritable component of plasma protein variation ranged from over 60 % to almost zero. In conclusion, the rapid development speed and the power of the OpenMS library allowed the researchers to implement a new algorithm for SWATH-MS data and successfully apply it to a complex human sample.

B. Degradomics

The human genome encodes for over 460 active proteases, which shape proteom composition and functionality [10]. Quantitative proteomics employing stable isotope labeling is widely used to investigate the (patho)-physiological role(s) of proteolytic enzymes. Often, enrichment schemes for N- or C-terminal peptides are used in order to identify proteolytically generated protein neo-termini. In combination with loss- or gain-of-function systems, such strategies are suited to unravel the substrate repertoire of a protease under investigation [10].

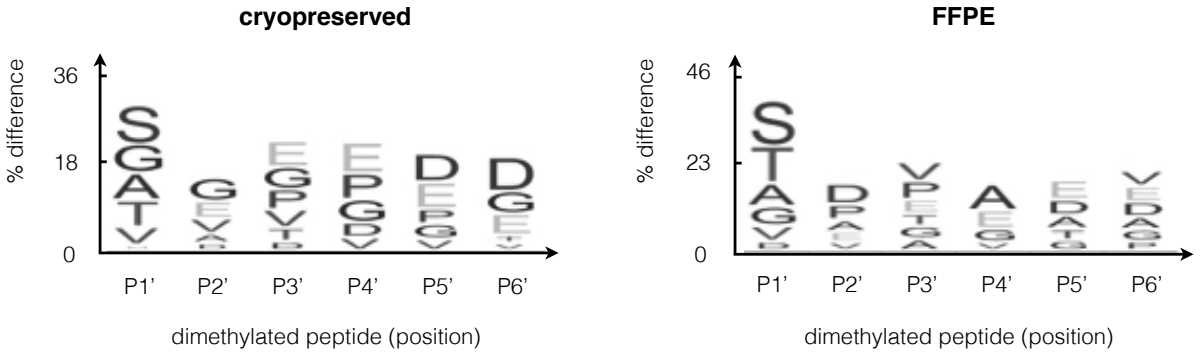

FIG. 1: **Degradomics case study.** N-terminal dimethylated peptides in cryopreserved and FFPE [8]. In both sample types, the same prominent fingerprint of serine, glycine, valine, alanine and threonine at the P1' position were observed.

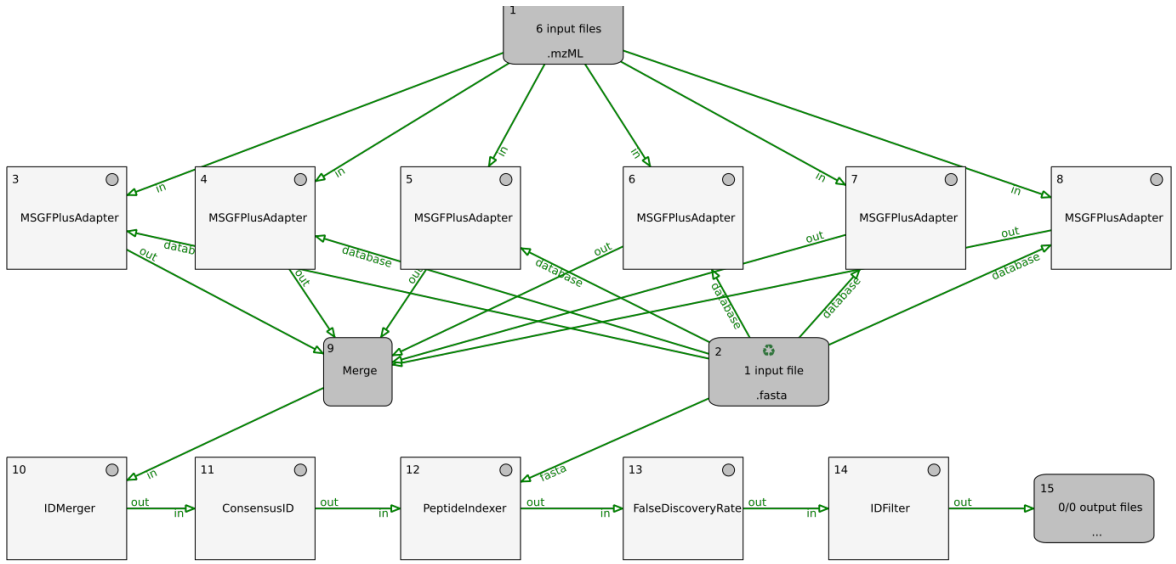

FIG. 2: **Degradomics case study workflow.** Subset of the OpenMS analysis workflow used in (Lai et al. [8]). Peptide sequence searches with six different fixed modifications (light / heavy dimethylation, monomethylation and acetylation) using the external search engine MS-GF+ [9] are subsequently combined.

In a recent study [8], this strategy has been expanded to include formalin-fixed, paraffin-embedded (FFPE) specimens, which represent the vast majority of samples that are stored in clinical archives, see Fig. 1. The aforementioned study highlighted the versatility of

OpenMS. In particular, OpenMS enabled the parallel and unbiased analysis of multiple, prevalent N-terminal modifications in different, isotopically labeled states, Fig. 2. Secondly, OpenMS enabled accurate relative quantitation of stable-isotope-labeled peptides. This included addressing so-called "knock-out" situations, i.e. peptides that occur in only one isotope state and are missing in the corresponding second isotope channel [11]. Typically, this situation is prone to distorted quantitation due to chromatographic background signals [12]; a problem that is resolved by accurate peptide feature detection in OpenMS. The workflow characterized thousands of protein N-termini in FFPE specimens and identified 23 potential substrates for the lysosomal protease cathepsin L [8].

C. Proteogenomics

Proteogenomics is an expanding field of inquiry that uses proteomic data in search of direct evidence of protein expression, in order to refine the annotation of protein-coding regions in genomes. In the case of shotgun tandem mass spectrometry data, the sought-after pieces of evidence are peptides of "unexpected" origin, i.e. peptides that do not match a known protein sequence and that could thus lead to novel annotations if supported by orthogonal evidence such as RNA expression or sequence conservation. For genomes that are already well characterized, particularly the human one, these novel peptides represent needles in a haystack of peptides matching known proteins. Notably, the difficulty in finding genuine novel peptides is exacerbated by the importance of avoiding false positive assignments that could lead to spurious annotations.

Several prerequisites are essential for a successful proteogenomics endeavor: A suitable proteomics dataset, a comprehensive database comprising both known and potential novel protein-coding sequences, and the expertise in genome annotation. ,Moreover, a data analysis workflow is needed that reliably and sensitively identifies peptides and filters them according to rigorous criteria. Such a workflow should operate in a reproducible fashion and allow for high throughput, to enable the analysis of large datasets. OpenMS is particularly well suited for this application, as it facilitates creating complex data analysis pipelines with access to a large variety of available functionality. Once a pipeline is established, it can be configured to account for experimental parameters and applied to large datasets. We recently reported stringent guidelines for proteogenomic data analysis and interpretation,

these analyses were based on OpenMS pipelines to run database searches of spectra followed by a rescoring step [16]. Our recent extension of the capabilities of OpenMS now enable a complete proteogenomic analysis workflow, which encompasses database searches using multiple search engines, rescoring and combining of search results, filtering according to various criteria on PSM, peptide and protein levels, and data export – within OpenMS (see Fig. 3). Processing of over 55 million tandem mass spectra from different publicly available datasets using OpenMS-based pipelines has led to over 40 new protein-coding gene annotations in the GENCODE human reference genome [17]. While this may seem like a low number, our approach for calling novel peptides is intentionally conservative and based on high quality proteomics evidence together with additional orthogonal support. Notably, only a fraction of the novel peptides identified from proteomic data gave rise to new annotations during the manual review process. On the other, the vast majority of the information generated in a proteogenomic analysis pertains to known proteins; this information may be valuable for other purposes, e.g. to elucidate tissue-specific protein expression levels [18]. Overall, the capabilities for creating flexible analysis pipelines, in connection with the wealth of functionality offered by TOPP tools and integrated third-party tools, make OpenMS especially well-suited for large-scale examination of publicly available data, be it with a proteogenomic or applications with a different focus such as differential (quantitative) or phosphoproteomic analyses.

---

[1] L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann, A. D. Pizarro, et al., Molecular & cellular proteomics : MCP **10** (2011).

[2] M. Sturm, A. Bertsch, C. Gröpl, A. Hildebrandt, R. Hussong, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, A. Zerck, K. Reinert, et al., BMC Bioinformatics **9**, 163 (2008).

[3] H. Weisser, S. Nahnsen, J. Grossmann, L. Nilse, A. Quandt, H. Brauer, M. Sturm, E. Kenar, O. Kohlbacher, R. Aebersold, et al., Journal of Proteome Research **12**, 1628 (2013).

[4] H. L. Röst, G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinović, O. T. Schubert, W. Wolski, B. C. Collins, J. Malmström, L. Malmström, et al., Nature Biotechnology **32**, 219 (2014).

[5] E. Kenar, H. Franken, S. Forcisi, K. Wörmann, H.-U. Häring, R. Lehmann, P. Schmitt-

Kopplin, A. Zell, and O. Kohlbacher, Molecular & cellular proteomics: MCP **13**, 348 (2014).

[6] Y. Liu, A. Buil, B. C. Collins, L. C. Gillet, L. C. Blum, L.-Y. Cheng, O. Vitek, J. Mouritsen, G. Lachance, T. D. Spector, et al., Molecular systems biology **11**, 786 (2015).

[7] L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, and R. Aebersold, Molecular & cellular proteomics: MCP **11**, O111.016717 (2012).

[8] Z. W. Lai, J. Weisser, L. Nilse, F. Costa, E. Keller, M. Tholen, J. N. Kizhakkedathu, M. Biniossek, P. Bronsert, and O. Schilling, Molecular & Cellular Proteomics **15**, 2203 (2016).

[9] V. Granholm, S. Kim, J. C. Navarro, E. Sjolund, R. D. Smith, and L. Kall, Journal of proteome research **13**, 890 (2013).

[10] Z. W. Lai, A. Petrera, and O. Schilling, Current opinion in chemical biology **24**, 71 (2015).

[11] L. Nilse, F. C. Sigloch, M. L. Biniossek, and O. Schilling, Proteomics Clinical Applications **9**, 706 (2015).

[12] S. Tholen, M. L. Biniossek, A.-L. Geßler, S. Müller, J. Weißer, J. N. Kizhakkedathu, T. Reinheckel, and O. Schilling, Biological chemistry **392**, 961 (2011).

[13] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, Electrophoresis **20**, 3551 (1999).

[14] S. Kim, N. Gupta, and P. A. Pevzner, Journal of Proteome Research **7**, 3354 (2008).

[15] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, Nature Methods **4**, 923 (2007).

[16] J. C. Wright, J. Mudge, H. Weisser, M. P. Barzine, J. M. Gonzalez, A. Brazma, J. S. Choudhary, and J. Harrow, Nature Communications **7**, 11778+ (2016), ISSN 2041-1723, URL `http://dx.doi.org/10.1038/ncomms11778`.

[17] J. Harrow, F. Denoeud, A. Frankish, A. Reymond, C.-K. Chen, J. Chrast, J. Lagarde, J. G. Gilbert, R. Storey, D. Swarbreck, et al., Genome Biol **7**, S4 (2006).

[18] R. Petryszak, M. Keays, Y. A. Tang, N. A. Fonseca, E. Barrera, T. Burdett, A. Füllgrabe, A. M.-P. Fuentes, S. Jupp, S. Koskinen, et al., Nucleic acids research **44**, D746 (2016).

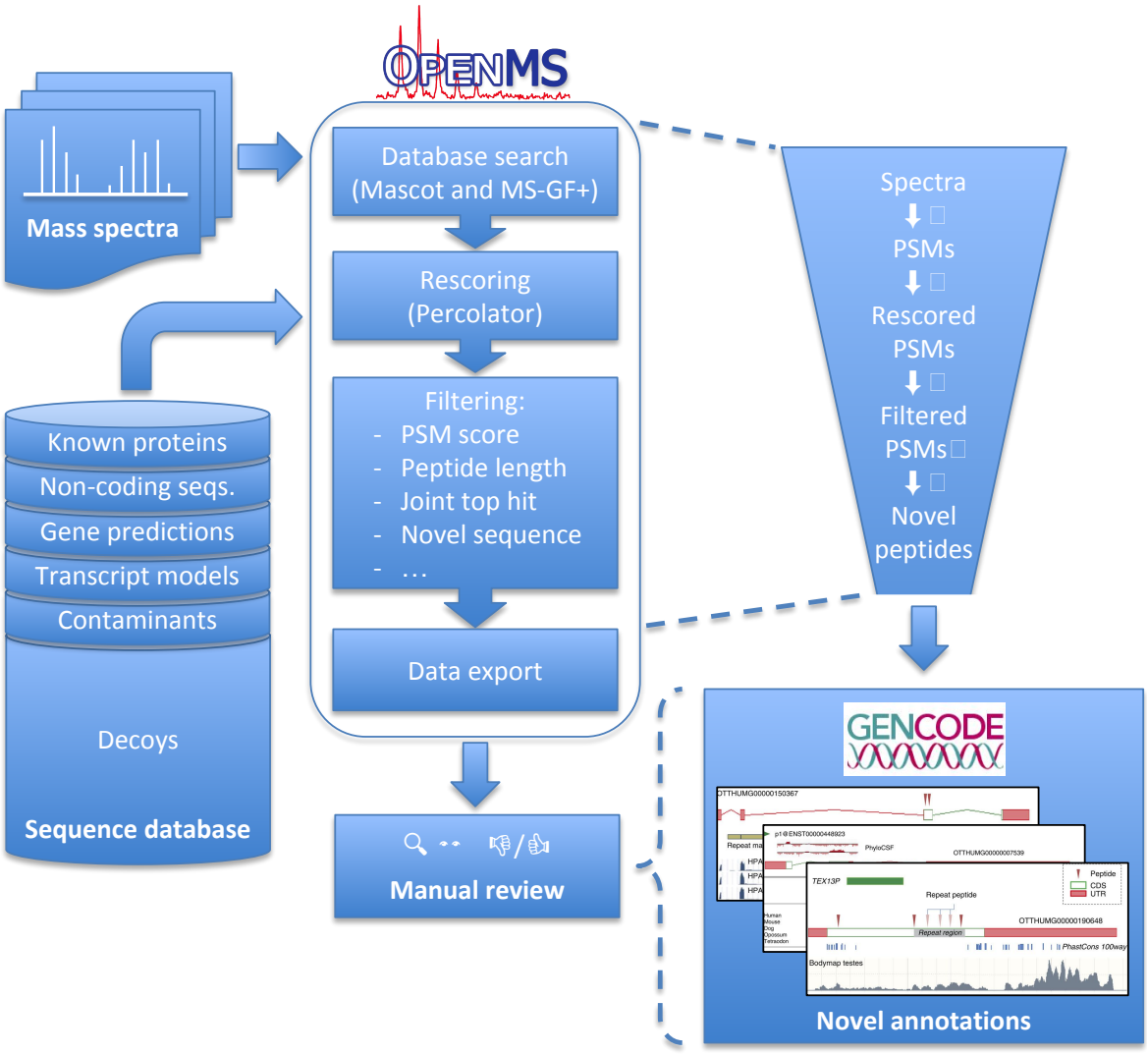

FIG. 3: **Schematic overview of the OpenMS proteogenomics workflow.** Based on a comprehensive sequence database, tandem mass spectra from large proteomic datasets were searched in a competitive target/decoy approach using two search engines, Mascot [13] and MS-GF+ [14]. The search results were rescored using Percolator [15], then filtered in multiple stages according to stringent quality criteria [16]. During this process, starting from a large number of spectra and initial PSMs, the set of retained PSMs was refined further and further, until in the end only high-confidence PSMs from novel peptides remained. These were exported and passed on to the annotators. In a manual review process, novel peptides and other evidence sources were integrated, which in some cases yielding novel genome annotations.