RUHR-UNIVERSITÄT BOCHUM

Medical Faculty
Medical Proteome-Center
Medical Bioinformatics

F953N-19 FoRUM

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung FKZ 031 A 534A

ProtGraph, a graph approach for representing proteins and peptides

Dominik Lux, Katrin Marcus, Martin Eisenacher and Julian Uszkoreit

Abstract

We present ProtGraph[1], a novel approach to represent proteins and digested peptides with feature information, by utilizing a graph structure. It is generated by utilizing the SwissProt-EMBL-Format provided by UniProt, which gives various information about the protein and its features such as isoforms, variants and/or signal peptides. ProtGraph maps the canonical sequence to a path in a directed acyclic graph extending it with feature- and digestion-information by adding specific nodes and edges. All possible paths can then be retrieved by traversing from a dedicated start and end node yielding digested peptides.
Via a dynamic programming approach different statistics can be retrieved in an efficient manner. We show that the search space in peptide identification increases exponentially when using isoform and variant-features. With ProtGraph we provide a fast tool for generating a graph structure from proteins and offer multiple export functionalities, such as FASTAs for search engines, to be used with other tools. As a next step, we would like to explore more sophisticated algorithms on this graph structure to further investigate the search space and to be able to query the graph directly and efficiently for weights.

1. ProtGraph is available on GitHub: https://github.com/mpc-bioinformatics/ProtGraph

Motivation

In today's proteomics workflows FASTA-files are a key parameter for database based peptide search engines to identify spectra generated by mass spectrometers. Summarized, a FASTA-file usually contains header information of a protein with its corresponding canonical amino acids. The FASTA-files are sometimes extended by isoform sequences of proteins.
Our goal is to add further information into the FASTA-files, by utilizing already available information, such as signal peptides, variants, mutagens, conflicts and more which can be publicly retrieved from UniProt.

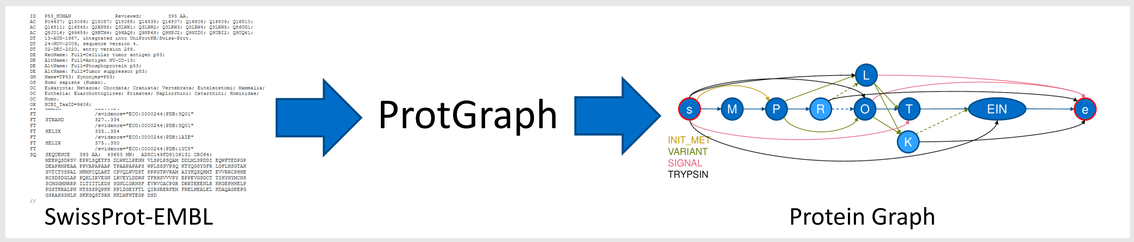


Figure 1: Generation of protein graphs with ProtGraph. *A small workflow to illustrate the in- and output of ProtGraph. The SwissProt-EMBL entry of a protein contains all needed information which ProtGraph utilizes to generate a protein graph.*

To achieve this goal, we used a graph structure, hence creating ProtGraph, allowing us to generate protein graphs. With this data structure, we are able to retrieve statistical information of proteins, showing that it may not be feasible in some cases to generate such FASTA-files from specific proteins

Graph Generation

Figure 2a):

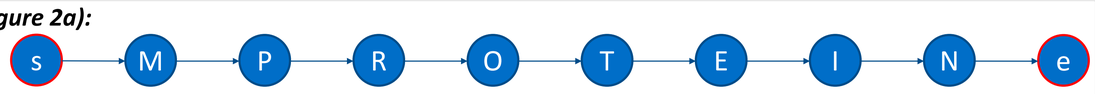


Figure 2b):

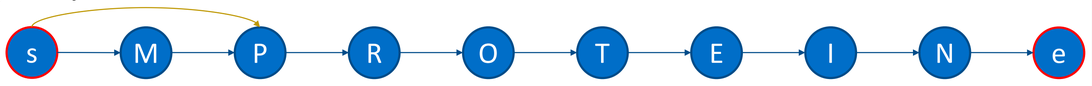


Figure 2c):

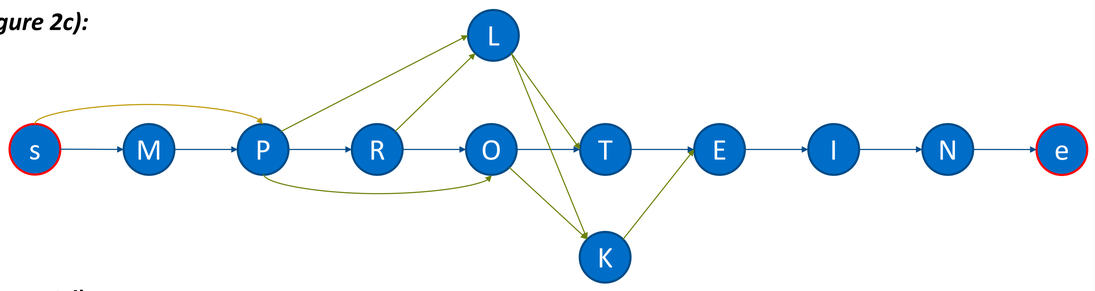


Figure 2d):

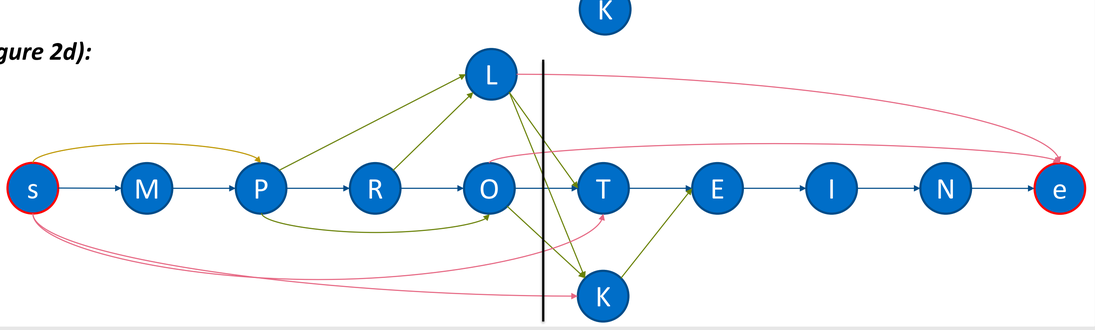


Figure 2: Creation of directed acyclic graphs. *2a) shows the initial graph ProtGraph generates after reading the canonical sequence. 2b) illustrates that one additional edge needs to be introduced while reading the feature entry INIT_MET indicating that the initiator methionine might be skipped. The graph can then successively increase in complexity, the more features are applied. 2c) shows the graph after parsing variant information and 2d) shows the graph after parsing the signal peptide information.*

ProtGraph always generates directed acyclic graphs from proteins with dedicated start and end nodes. The initial graph from Figure 2a) can then be extended by applying specific rules depending on the features. In Figure 2d) it can be observed that the graph gets cleaved due to the applied feature: signal peptide (black line). Additionally, ProtGraph allows the user to choose the features which should then be applied on the initial graph.

Graph Digestion and Optimization

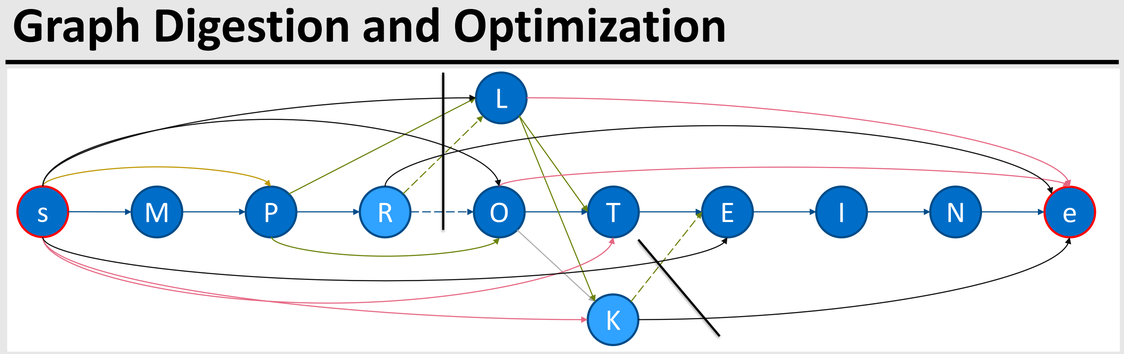


Figure 3: Illustration of the digestion via the enzyme Trypsin. *The protein graph after applying the digestion enzyme. Black lines indicate the cleavage points where additional out- and ingoing edges are added to the nodes on the left and right side of the line, respectively.*

We implemented a digestion step as well as a graph optimization step. Both steps keep the graph directed and acyclic. We observed that the size of most protein graphs reduces drastically after the optimization step due to containing long chains of nodes.

Figure 4a):

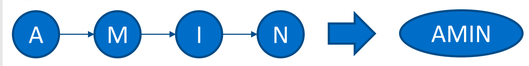


Figure 4b):

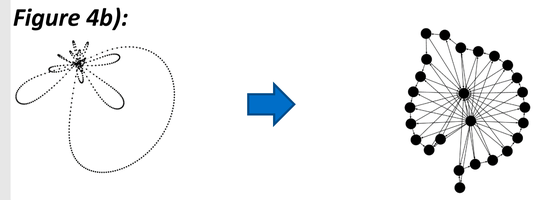


Figure 4: Illustration of the graph optimization step. *The Figure 4a) shows the optimization step abstractly, concatenating chains of nodes into one node. Figure 4b) shows the optimization step on an actual protein graph reducing its size.*

Information Retrieval with the Topological Order

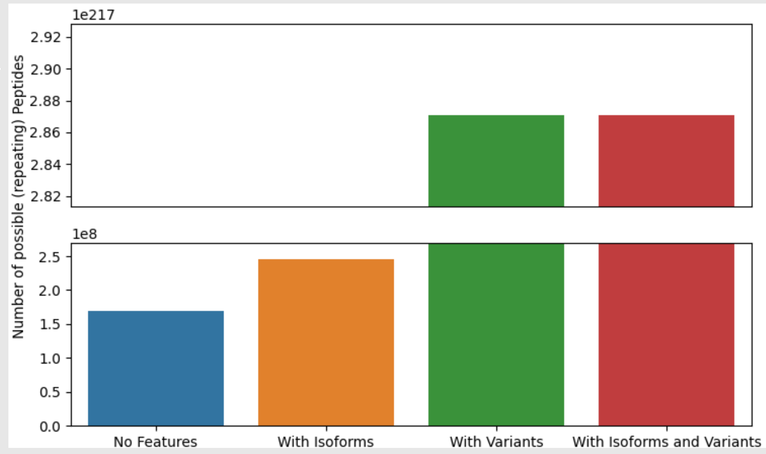


Figure 5: Number of possible peptides[2] of protein graphs *The plot shows the number of possible peptides[2] on the tryptically digested human-proteome dataset. The number of possible peptides is low if no or only isoform features are applied on the protein graphs. The number of possible peptides explodes if applying variant features. Consequently, the number of possible peptides is high applying both, the isoform and variant features.*

Table 1: Number of possible peptides by miscleavages[2]. *This table provides absolute numbers of possible peptides[2] by miscleavages retrieved from protein graphs with no applied features. The values were summed for the complete UniProt database and may vary slightly when recalculating due to different versions of UniProt.*

#Mssclvgs	#Possible Peptides[2]
0	7 488 643 278
1	7 278 922 167
2	7 069 457 392
3	6 860 625 343
4	6 652 821 545
5	6 446 737 427
6	6 243 206 955
7	6 042 551 084
8	5 844 964 760
9	5 650 834 085

A topological order can be generated from protein graphs. In combination with dynamic programming, interesting statistics can be retrieved. Figure 5 shows the number of possible peptides which were retrieved with ProtGraph. Additionally, the number of possible peptides[2] can be further broken down by the number of miscleavages (Table 1) as well as by the peptide-length (not shown).

2. Peptides with the same sequence can occur multiple times in a protein graph.

Outlook and Future Work

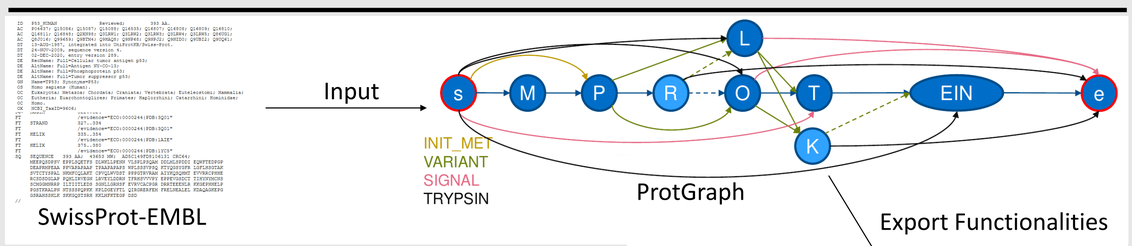
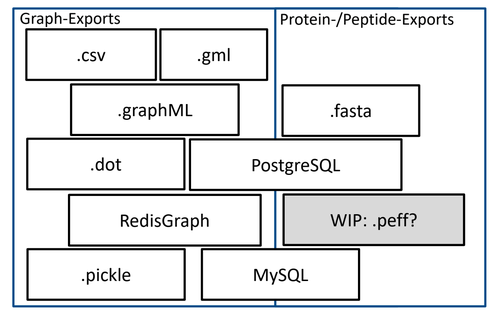


Figure 7: Overview of export functionalities. *This figure illustrates that the generated protein graphs from ProtGraph can be exported into various formats. ProtGraph supports many different graph exports, so that other tools can reuse the generated graphs for other purposes. Additionally, protein- and peptide exports are supported, in which ProtGraph saves the corresponding proteins/peptides which are retrieved by traversing the graph from the dedicated start- and end-node.*



With ProtGraph, we can generate protein graphs quickly and efficiently, even allowing various configurations, such as defining the applied features/applied digestion enzymes. Multiple various export into graphs and/or proteins/peptides are also possible allowing the output to be used on powerful graph algorithms as well as in identification workflows.

In future we aim to utilize the graph structure directly. We are especially interested on querying such graphs by weight, retrieving the corresponding proteins/peptides. First experimental results hint that such a retrieval may be possible on such a graph structure.