**Extended Reactome Analysis tool**

# **Introduction**

## Definitions

A \_\_**proteoform**\_\_ is a protein with a set of post translational modifications.

A \_\_protein\_\_ is identified by a standard protein identifier such as UniProt accession or Ensembl.

A \_post-translational modification\_(PTM) is uniquely identified by two elements: type and coordinate in the protein sequence.

The PTM \_\_type\_\_ is specified following the PSI-MOD ontology. No other types are supported.

The PTM \_\_coordinate\_\_ is an integer number indicating the position in the protein amino acid sequence where the post-translational modification occur.

\_\_Matching\_\_ is deciding if an input proteoform corresponds to a proteoform in the database.

An input proteoform can be connected to 0 or more stored proteoforms. They can match perfectly or with a margin of error.

A PTM \_\_unkown coordinate\_\_ in the input is specified by an integer 0 or null.

A \_\_perfect PTM match\_\_ is when two PTMs have the exact same coordinate and type.

A \_\_perfect proteoform match\_\_ is when an input proteoform and a stored proteoform fullfill these conditions:

- The protein accession is the same.

- The isoform is the same. If no isoform is specified, the canonical is taken.

- For each PTM in the input, there is perfect PTM match in the stored proteoform.

Having an integer \_n\_. A \_\_range\_\_ for PTM coordinates is the set of protein sequence coordinates that fall in the closed interval [n-1, n+1].

An \_\_imperfect match\_\_ is when an input proteoform and a stored proteoform fullfill these conditions:

- The protein accession is the same.

- The isoform is the same.

- At least one input PTM has no perfect match in the stored proteoform.

For each PTM in the input, there is a PTM in the stored proteoform with:

- The input PTM type is less specific than the stored PTM, using the PSI-MOD hierarchy (the input type is a parent of the stored type).

- The coordinates fulfill at least one of these conditions:

- Either the input PTM coordinate falls in the range of the stored PTM coordinate.

- Or the input PTM coordinate is unknown.

An input proteoform is not matched (not considered found) when at least one of the following conditions happen:

- The identifier is not found in Reactome.

- The specific isoform is not found in Reactome.

- One input PTM has no perfect or imperfect match:

- At least one input PTM is more specific than the stored PTMs. In other words, the input PTM type is a child of the stored PTM type in the ontology hierarchy tree.

- At least one input PTM with the same coordinate was not found in the stored proteoform.

## Project overview

# Methods

## Knowledge annotated in Reactome

### How are subsequence ranges annotated in Reactome

Proteins are commonly processed with cleaving or modifications in order to perform their functions. Very often they do not stay complete the way they are right after translation. For example, the initial residue is removed, because it is just the starting signal of the protein and all of them share that initial methionine. Sometimes, also peptide regions have to be removed so that the protein can actually perform its task. In other situations, only a subsequence of the protein is necessary to perform the task, then it is necessary to specify the range of the subsequence with respect to the full sequence.

By default the sequences refer to the canonical sequence of the proteins. In case they refer to other variants of the sequence then, the isoform is specified.

The subsequences of the proteins are represented by the class "EntityWithAccessionedSequence" (\_ewas\_) in Reactome. It always has "startCoordinate" and "endCoordinate" fields to specify the start and end amino acids in the full sequence of the protein. When the whole protein sequence is meant, then either the start and end are not specified or the coordinates of the start and end positions of the full sequence are annotated. In some cases, either the start or end coordinate are not known, therefore they are left blank. The display name of the \_ewas\_ will contain in parenthesis the subsequence range.

### How are Isoforms annotated in Reactome

The isoform number 1 is the main isoform of the protein, which is selected by UniProt. The isoforms annotated in Reactome are not updated regularly. Therefore, in case the isoform numbering changes in UniProt, the annotated isoforms in Reactome might diverge from the current ones.

When a protein has only one version then there is no need to specify the isoform. The default isoform is "-1", which is often ommited. When an isoform is specified, it means only that isoform can perform the function it is supposed to do. In those cases, even the isoform "-1" will be annotated. Otherwise, there is no need to annotate it. This applies also for the proteins having ine or more isoforms.

// TODO: Create a script to perform a check in Reactome if the coordinates of the subsequence ranges and full sequence coordinates match the ones of the actual sequences in UniProt. Extend to check also the ptm coordinates. In case there are some discrepancies, download the history of the protein from UniProt and check if the protein was really annotated according to the sequence by that time. OUTPUT the list of

corrections needed to be made in order to keep the information matched.

## Tools

### Proteoform format converter tool

### Peptide extractor too

## Testing

### Unit testing

// TODO

### Performance testing

// TODO

## Previous analysis process

### Input

### Search and Matching

### Statistical Analysis

## New analysis process

### Input

#### Gene name list

#### UniProt accession list

#### Gene NCBI / Entrez list

#### Small molecules (ChEBI)

#### Small molecules (KEGG)

#### Simple Proteoforms list

#### Expression values

\* Files can be in one line or multiple lines.

\* File contains multiple columns. The first column in each row contains the identifier. All the other columns must contain

expression values as floating point numbers.

\* Identifiers can be repeated in multiple lines?

Microarray data

#### Metabolomics data

#### Cancer Gene Census

#### Simple proteoforms list with expression values

### Other formats

#### PEFF

#### Protein Ontology

#### GPMDB

#### Simple proteoform list

Each line of the file corresponds to a single proteoform.

A line consists of two fields separated by ';'. First a Uniprot Accession and second a set of PTMs.

The second field is optional. Lacking a PTM set means a proteoform without modifications.

The PTM set specifies presents each PTM separated by a ','.

Each PTM is specified using a modification identifier and an integer coordinate, separated by ':'(semicolon).

The modification identifier is a 5 digit id from the Protein Modification Onthology [\[2\]](#references).

For example: "133:00046, which corresponds to [O-phospho-L-serine](https://www.ebi.ac.uk/ols/ontologies/mod/terms?iri=http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FMOD\_00046) at the coordinate 133.

Single proteoform examples:

- A single protein with no modifications

~~~~

P00519

~~~~

- A protein with one PTM. The two fields are separated by a ';'

~~~~

P16220;133:00046

~~~~

- A protein and a set of PTMs separated by ','. The PTMs can be ordered randomly.

~~~~

P62753;235:00000,236:00000,240:00000

~~~~

In case the PTM type is not known, the modification id used is "00000". For example: "00000:245".

In case the PTM coordinate is not known, the integer used is 0.

<br>File example:

~~~~

P10412-1

P10412

P56524;559:00916

P04637;370:00084,382:00084

P56524;246:00916,467:00916,632:00916

P12345-2;246:00916,467:00916,632:00916

Q1AAA9

O456A1

P4A123

A0A022YWF9

A0A022YWF9;246:00916,467:00916,632:00916

~~~~

#### Protein Ontology

The PRO format[1] is not approved as a formal standard by an international proteomics organization, but it is a convenient way of representing proteoforms.

The format allows representation of proteins with isoforms, subsequence ranges and post translational modifications. Each protein is specified in one line composed of the following blocks:

* Sequence block:
  + Database name: «UniProtKB:»
  + Uniprot accession: «Q08775»
  + Optional Isoform: «-5**»**
  + **Opional Subsequence range. «**1-528»
* Optional modification blocks:
  + N-teminal-most modified amino acid and position: «Ser-203»
  + In case there are more modifications, separator: «/»
  + Additional modified amino acids
  + Modification separator: «,»
  + Modification type: «MOD:00046»
  + In case there are more modification blocks, separator: «|»

The full example looks like this:

UniProtKB: Q08775-5,1-528,Ser-203,MOD:00046| Thr-205/Thr-207,MOD:00046

This format was decided to be fully supported by the input of the new Reactome Analysis.

#### PEFF

The PEFF proposed format[2] intends to allow researchers to represent many different types of proteomics data into a single format. Therefore, it can become very complex to represent some of the proteoforms. The standard is not yet published as a formal standard from HUPO, as a consequence it will not yet be supported by the new Analysis Input of Reactome.

#### GPMDB

### Search and Matching

#### Procedure

First, the proteins are filtered to only those with that UniProt accession.

Second, proteins are filtered by isoform.

Third, proteins are filtered by subsequence ranges.

Fourth, the set of ptms is matched.

The input type is decided using the first 5 lines of the file, without counting headers.

An Ensembl identifier will be mapped to UniProt accession. Ensembl is only allowed in the GPMDB format.

PTMS are stored as radix tree leaves attributes. Save them as a sorted set ordered by mod type and then coordinate. Because PTMs with the same type appear in more than one coordinate, but, theoretically, one coordinate can not contain more than one PTM at once.

Sort the PTMs in the input, by mod and then by coordinate. First search for the protein accession in the radix tree. An unknown PTM coordinate is stored as null, to avoid counting the 0 in the range of near coordinates.

#### Implementation

Intermediate data structure

### Statistical Analysis

#### Procedures

#### Implementation

# Results

Improvements over the previous version

\* Use Google Guava Stopwatch class to measure input preprocessing performance.

It is based on the System.nanoTime(), instead of System.currentTimeMillis(),

which measures the elapsed wall-clock time. In contrast, System.nanoTime() returns the current value of the most

precise available system timer, which is specifically developed to measure elapsed time.

\* Faster reading of expression input.

Traverse the input fewer times. Currently, it is traversing all input at least 3 times using:

split, replaceAll, and StringTokenizer.

\* Added possibility to specify proteoforms.

\* Add support for PEFF format.

\* Added support for Protein Ontology (PRO) format.

\* Added support for a simple custom proteoform format.

# Conclusions

# References

\[1\] [UniProt: the universal protein knowledgebase. Nucleic Acids Res. 45: D158-D169 (2017)](http://dx.doi.org/doi:10.1093/nar/gkw1099) <br>

\[2\] [The PSI-MOD community standard for representation of protein modification data. Nature Biotechnology 26, 864 - 866 (2008)](http://www.nature.com/nbt/journal/v26/n8/full/nbt0808-864.html) <br>

hp

1. Natale, D.A., et al., *Protein Ontology (PRO): enhancing and scaling up the representation of protein entities.* Nucleic Acids Res, 2017. **45**(D1): p. D339-d346.

2. Mayer, G., et al., *The HUPO proteomics standards initiative- mass spectrometry controlled vocabulary.* Database, 2013. **2013**: p. bat009-bat009.