**Extended Reactome Analysis tool**

# **Introduction**

## Definitions

A *protein* is identified by a standard protein identifier such as UniProt [1] accession or Ensembl[2].

Proteins can have multiple possible amino acid sequences, each posibility is called *isoform* and they are represented by appending a «-#» (dash and an integer number) stating the number of the isoform according to UniProt.

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

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 [3]. No other types are supported.

The PTM *coordinate* is an integer number indicating the position in the full protein amino acid sequence where the post-translational modification occur. The *coordinate* can also be called *site.* The full sequence may vary depending on the isoform of the protein. Sometimes the coordinate is not known or it can differ a few positions from one reference database to another, depending on the experimental source backing up the annotation.

Proteins are commonly cleaved or modified in different ways to perform their biological purpose. They can be processed to keep only a subsequence of peptides from the original sequence and remove the rest. These subsequences can be represented by a protein accession and a pair of coordinates.

## Project overview

We intend to extend the analysis capabilities of Reactome[4], to include not only protein or gene identifiers but also fine grained data such as protein subsequences, isoforms and post-translational modifications.

# Methods

## Knowledge annotated in Reactome

The representation of proteins and proteoforms involve multiple data classes in the Reactome Model (<https://reactome.org/documentation/data-model>).

*ReferenceEntity* objects represent the invariant attributes of a molecule, such as a protein. For the case of proteins, they have the following attributes among others:

* Identifier: the UniProt accession
* referenceDatabase: «UniProt»

*EntityWithAccessionedSequence* objects represent proteins and nucleic acids with known sequences.

|  |  |
| --- | --- |
| **Real Object** | **Reactome data class** |
| Protein | ReferenceEntity |
| UniProt accession number | + identifier |
| Database name: UniProt | + databaseName |
| Protein subsequences | EntityWithAccessionedSequence |
| Subsequence start coordinate | + startCoordinate |
| Subsequence end coordinate | + endCoordinate |
| Proteins in a subcellular location | EntityWithAccessionedSequence |
| Subcellular location | EntityCompartment |
|  | + name |
| Proteins in a modified form | EntityWithAccessionedSequence |
| Protein Isoform | ReferenceIsoform |
|  | + variantIdentifier |
| Post-translational modification | TranslationalModification |
|  | + psiMod |
|  | + coordinate |
| PsiMod term | PsiMod |
|  | + databaseName |
|  | + displayName |

### 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 omitted. 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 one 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

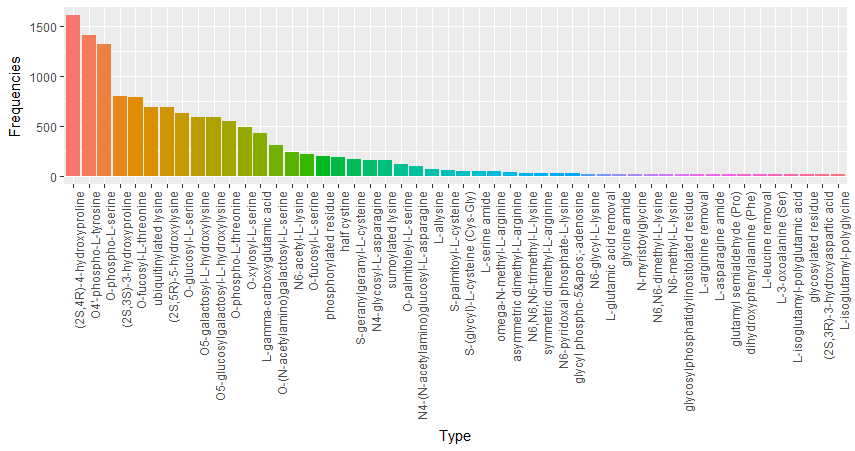
corrections needed to be made in order to keep the information matched.

A unkown coordinate is annotated by a ‘?’, a null field, an empty field or as -1.

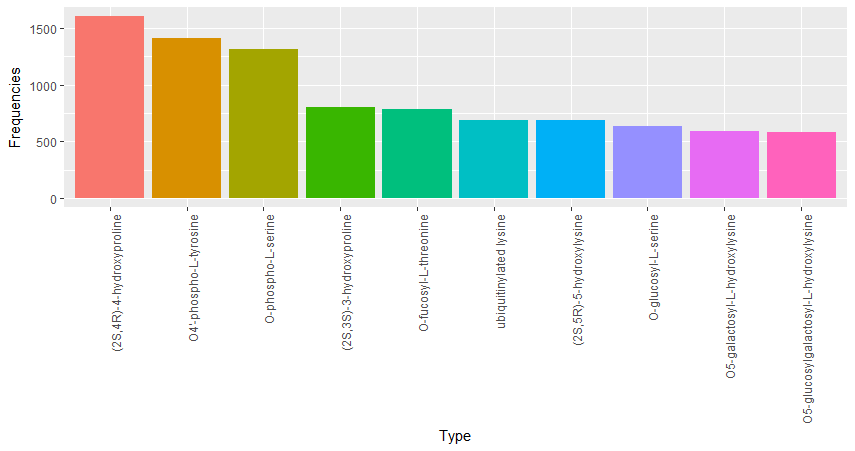
### How are PTMs annotated in Reactome

The number of different annotated protein modifications types from PSI-MOD in Reactome is 155.

Frequencies of protein modification types in humans:



The most common types are:



## Matching input data to reference data

To search for reactions and pathways in the reference database it is necessary to decide if each entity in the input is equivalent to an entity in the database. That is the case for proteins, small molecules, gene names etc.

For our purposes, we will only focus on sample sets related to proteins.

We call *matching* the process of deciding if an input protein/proteoform is equivalent to a protein/proteoform in the database. We say an input proteoform *matches* a reference proteoform if they fulfill certain conditions described in the following approaches.

We propose to implement a new approach that takes different forms of proteins to filter out the results and make them more specific. Several options are possible. We explain some of them briefly.

### Protein only

The previously implemented approach in Reactome that does not consider proteoforms altogether. It uses only the protein accession numbers to decide if the protein is annotated in the database or not. Then it performs a search for reactions and pathways where the matched proteins are participant.

### Full proteoform match

The input contains not only protein identifiers but proteoforms in any format. Proteoforms are matched with the ones contained in the reference database so that the reaction and pathway search is performed using proteoforms instead of proteins.

The conditions needed to say an input proteoform is the same as a reference proteoform:

* The Uniprot Accession is found, and it is the same
* The Isoform:
  + If provided it is the same
  + If not provided it should be missing in the reference too.
* The subsequence range:
  + If provided the coordinates should be the same.
  + If not provided, then the reference proteoforms can have any coordinates.
* The PTMs:
  + The input and reference have the same amount of PTMs.
  + All PTMs have the same PSI-MOD type.
  + All PTMs have the same coordinates.

### Approach 2 – Partial proteoform match

The conditions needed to say an input proteoform is the same as a reference proteoform:

* The UniProt Accession is the same
* The Isoform:
  + If provided it is the same.
  + If not provided it should be missing in the reference too.
* The subsequence range:
  + If not provided, then the reference proteoforms can have any range.
  + If provided the coordinates are the same or fall within a parameterized range of positions away.
* The PTMs:
  + The input and reference have the same amount of PTMs.
  + All input PSI-MOD types are the same or a super type (higher in the PSI-MOD hierarchy) of the annotated ones. Table 1 shows examples of possible cases using the values displayed in Figure 1.
  + Coordinates:
    - Cases when the PTMs have matching coordinates:
      * They are equal.
      * They are different but fall in the closed interval defined by adding or substracting the margin.
      * The input coordinate is unknown.
    - Cases when the PTMs coordinates do not match:
      * They are different and the provided coordinate falls outside the range of the margin interval.
      * A coordinate was provided and the annotated coordinate is unknown.
      * The provided coordinate is negative or zero.

Table 2 show examples of possible cases.

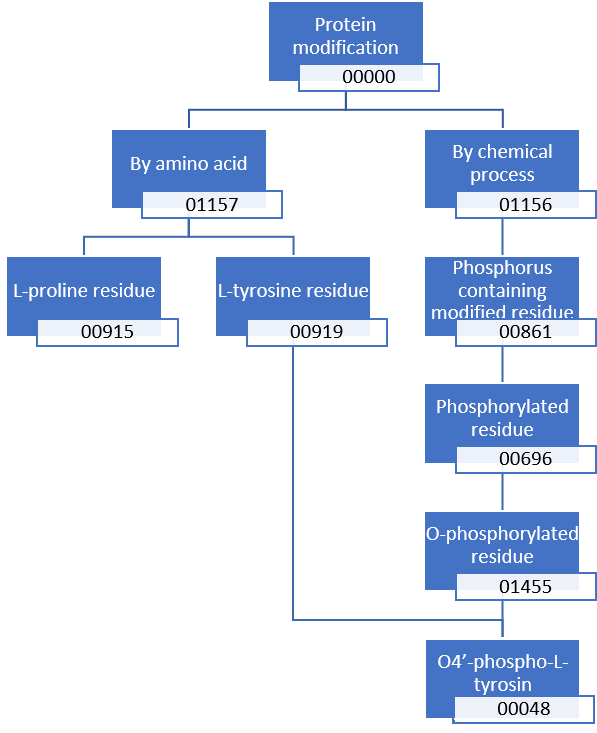


Figure 1 Phosphorylation PSI-MOD modification heirarchy

Table 1 PSI-MOD PTM types matching examples

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Type** | **Annotated Type** | **Matched** | **Comment** |
| 00000 | 00000 | Yes | They are equal |
| 00000 | 01156 | Yes | Input is less specific |
| 00000 | 00048 | Yes | Input is less specific |
| 00861 | 00000 | No | Input is too specific |
| 00861 | 00861 | Yes | They are equal |
| 00861 | 00048 | Yes | Input is less specific |
| 00861 | 00919 | No | They are Different |
| 00919 | 00048 | Yes | Input is less specific |
| 00048 | 00000 | No | Input is too specific |
| 00048 | 00919 | No | Input is too specific |
| 00048 | 00048 | Yes | They are equal |
| 11111 | Any | No | Does not exist |

Table 2 Possible cases when comparing input PTM coordinates with annotated PTM coordinates

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Annotated** | **Margin** | **Matched** | **Comment** |
| 17 | 17 | 0 | Yes | Equal |
| 16 | 17 | 0 | No | Out of margin |
| 18 | 17 | 0 | No | Out of margin |
| 7 | 13 | 5 | No | Out of margin |
| 8 | 13 | 5 | Yes | In margin |
| 9 | 13 | 5 | Yes | In margin |
| 17 | 13 | 5 | Yes | In margin |
| 18 | 13 | 5 | Yes | In margin |
| 19 | 13 | 5 | No | Out of margin |
| ? | 13 | 0 | Yes | Input is less specific |
| ? | 15 | 4 | Yes | Input is less specific |
| 13 | ? | 2 | No | Input is too specific |
| [empty] | [empty] | 5 | Yes | Equally unspecific |
| ? | ? | 5 | Yes | Equally unspecific |
| null | null | 5 | Yes | Equally unspecific |
| -3 | 1 | 5 | No | Input is negative |
| -3 | ? | 0 | No | Input is negative |
| 0 | ? | 0 | No | Input is zero |
| 0 | 1 | 3 | No | Input is zero |

### Approach 3 – 50% Partial proteoform match

Follows the same conditions as the Partial Match with one difference:

* The amount of PTMs matched in the reference proteoform are at least 50% of the total number of PTMs provided in the input.

### Approach 4 – 75% Partial proteoform match

Follows the same conditions as the Partial Match with one difference:

* The amount of PTMs matched in the reference proteoform are at least 50% of the total number of PTMs provided in the input.

### Approach comparison

Here is a comparison between the current approach of mapping with the proposed possible approaches. Table 3 shows the conditions are checked in each approach, whether the conditions are fulfilled or not. Table 4 show which conditions are necessary to be fulfilled to consider a protein/proteoform matched for each of the considered approaches.

Table 3 Overview of conditions checked to match a protein with another in the database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute is checked?** | **Protein only match** | **Full proteoform match** | **Partial proteoform match** | **75% Partial proteoform match** | **50% Partial proteoform match** |
| Protein Identifier | Yes | Yes | Yes | Yes | Yes |
| Isoform | No | Yes | Yes | Yes | Yes |
| Exact subsequence coordinates | No | Yes | Yes | Yes | Yes |
| Subsequence coordinates in margin | No | No | Yes | Yes | Yes |
| Exact PTM Coordinates | No | Yes | Yes | Yes | Yes |
| PTM Coordinates in Range | No | No | Yes | Yes | Yes |
| Exact PTM Types | No | Yes | Yes | Yes | Yes |
| PTM Types equal or less specific | No | No | Yes | Yes | Yes |
| 100% of PTMs | No | Yes | Yes | Yes | Yes |
| >= 75% of PTMs | No | No | No | Yes | Yes |
| >= 50% of PTMs | No | No | No | No | Yes |

Table 4 Overview of the conditions needed to match

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Needs to match?** | **Protein only match** | **Full proteoform match** | **Partial proteoform match** | **75% Partial proteoform match** | **50% Partial proteoform match** |
| Protein Identifier | Yes | Yes | Yes | Yes | Yes |
| Isoform | Yes | Yes | Yes | Yes | Yes |
| Exact subsequence coordinates | No | Yes | No | No | No |
| Subsequence coordinates in margin | No | Yes | Yes | Yes | Yes |
| Exact PTM Coordinates | No | Yes | No | No | No |
| PTM Coordinates in Range | No | Yes | Yes | Yes | Yes |
| Exact PTM Types | No | Yes | No | No | No |
| PTM Types equal or less specific | No | Yes | Yes | Yes | Yes |
| 100% of PTMs | No | Yes | Yes | No | No |
| >= 75% of PTMs | No | Yes | Yes | Yes | No |
| >= 50% of PTMs | No | Yes | Yes | Yes | Yes |

By taking each of the described approaches the number of entities that are not equivalent to any other in the search space reduces as we choose softer matching criteria. The largest search spaces is with the Full proteoform match where each proteoform. The smallest search spaces comes from the Protein only match approach where all the attributes of the proteoforms are ignored to have a generic view of the protein. A *non-redundant* entity is the one that has no other entity matching in the search space of possibilities. We call *modified proteoforms* the ones containing at least one PTM.

Table 5 shows measurements about the different entities (proteins/proteoforms) in the reference database for each matching approach.

Table 5 Statistics about entities for each matching approach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Protein only match** | **Full proteoform match** | **Partial proteoform match** | **50% Partial proteoform match** | **75% Partial proteoform match** |
| Total possible entities | **10353** | **12200** |  |  |  |
| Avg. reactions | **9.297** | **6.96** |  |  |  |
| Avg. pathways |  |  |  |  |  |
| Avg. reactions for modified proteoforms |  |  |  |  |  |
| Avg. pathways for modified proteoforms |  |  |  |  |  |
| Number of non-redundant entities |  |  |  |  |  |
| Percentage of non-redundant entities |  |  |  |  |  |
| Number of redundant entities |  |  |  |  |  |
| Percentage of redundant entities |  |  |  |  |  |
| Average size of equivalent entity groups |  |  |  |  |  |

## Object implementation

Protein coordinates are implemented as «Long» object classes. They can hold only integer numbers or «null». The empty fields, ‘?’ and -1 are converted to «null».

## Matching implementation

## Tools

### Proteoform format converter tool

### Peptide extractor too

## Testing

### Unit testing

### Performance testing

## 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[5] 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[6] 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

# Future additions

We plan to add support for peptides as input. A first stage considers all the available tryptic and nontryptic peptides available at the PRIDE[7] repository of ProteomeTools[8]. The set

# References

1. The UniProt, C., *UniProt: the universal protein knowledgebase.* Nucleic Acids Res, 2017. **45**(D1): p. D158-D169.

2. Aken, B.L., et al., *The Ensembl gene annotation system.* Database, 2016. **2016**: p. baw093-baw093.

3. Montecchi-Palazzi, L., et al., *The PSI-MOD community standard for representation of protein modification data.* Nat Biotechnol, 2008. **26**(8): p. 864-6.

4. Fabregat, A., et al., *The Reactome pathway Knowledgebase.* Nucleic Acids Research, 2016. **44**(Database issue): p. D481-D487.

5. 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.

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

7. Vizcaíno, J.A., et al., *2016 update of the PRIDE database and its related tools.* Nucleic Acids Research, 2016. **44**(D1): p. D447-D456.

8. Zolg, D.P., et al., *Building ProteomeTools based on a complete synthetic human proteome.* Nat Meth, 2017. **14**(3): p. 259-262.