There will be two ways to accept that proteoforms match:

* At least one PTM in common.
* At least all the annotated PTMs in common.

### At least one

* Both have the same UniProt Accession
* The Isoform is the same; either:
  + Both have an isoform specified. Ex: P31749-3
  + Both refer to the default one. Ex: P31749
* The subsequence range will not be taken into account in the regular use case. In case they are used, both Start and End coordinates must match. A pair of coordinates matches if:
  + Both are known (positive integer) coordinates and are the same.
  + Both are known (positive integer) coordinates and they are different, but the absolute difference between the two coordinates is less than or equal to a predefined margin.
  + One of the coordinates is unknown ("null", empty, "?", “-1”).
* The PTMs:
  + The input matches at least ONE of the reference PTMs.
  + A PTM matches if:
    - The types are either the same or one is super type of the other.
    - Same rules as above to compare coordinates.

### At least all the annotated PTMs ("Guilty until proven innocent")

* Both have the same UniProt Accession
* The Isoform is the same; either:
  + Both have an isoform specified. Ex: P31749-3
  + Both refer to the default one. Ex: P31749
* The subsequence range will not be taken into account in the regular use case. In case they are used, both Start and End coordinates must match. A pair of coordinates matches if:
  + Both are known (positive integer) coordinates and are the same.
  + Both are known (positive integer) coordinates and they are different, but the absolute difference between the two coordinates is less than or equal to a predefined margin.
  + One of the coordinates is unknown ("null", empty, "?", “-1”).
* The PTMs:
  + The input contains at least ALL the reference PTMs.
  + A PTM matches if:
    - The types are either the same or one is super type of the other.
    - Same rules as above to compare coordinates.

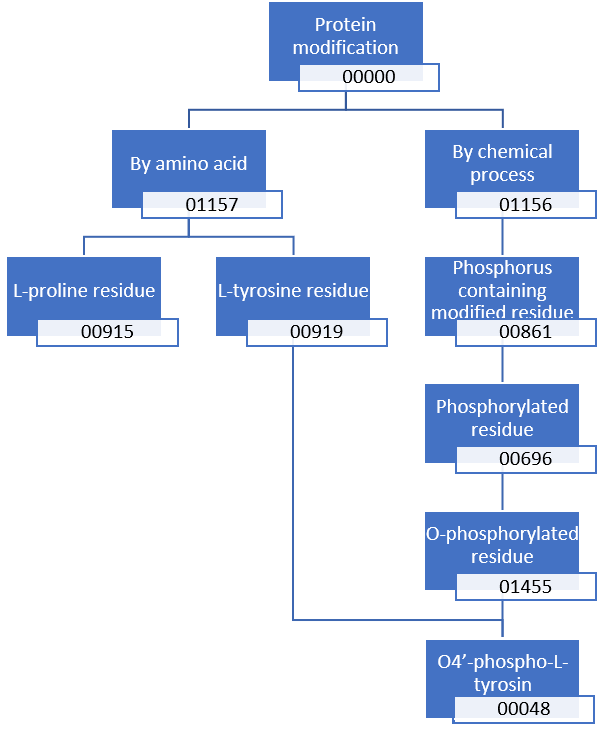
Extra considerations:

* Negative values, zero or floating-point numbers are invalid as sequence coordinates.
* We accept only PSI-MOD ontology modification types.
* The margin to compare the coordinates should be set as an unsigned integer parameter so that different values can be tested.

Tables 1 and 2 show examples to match the PTM types and coordinates respectively. The examples use the types depicted in figure 1, a subset of the PSI-MOD ontology.

*Table 1 Matching PTM types*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input Node type | Input Type | Annotated Type | Matched | Comment |
| Root | 00000 | 00000 | Yes | They are equal |
| Root | 00000 | 01156 | Yes | Input is less specific |
| Root | 00000 | 00048 | Yes | Input is less specific |
| Middle | 00861 | 00000 | Yes | Input is more specific |
| Middle | 00861 | 00861 | Yes | They are equal |
| Middle | 00861 | 00048 | Yes | Input is less specific |
| Middle | 00861 | 00919 | No | They are different |
| Leaf | 00048 | 00000 | Yes | Input is more specific |
| Leaf | 00048 | 00919 | No | Input is more specific |
| Leaf | 00048 | 00048 | Yes | They are equal |
| Non-existent | 11111 | Any | No | Input does not exist |



*Figure 1 Phosphorylation PSI-MOD modification heirarchy*

*Table 2 Matching 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 |
| 0 | 2 | 5 | No | Input in margin but not valid |
| -1 | 2 | 5 | No | Input in margin but negative |
| ?, empty, null | Positive integer | k | Yes | Input is less specific |
| Positive integer | ?, empty, null, -1 | K | Yes | Input is more specific |
| ?, empty, null | ?, empty, null, -1 | k | Yes | Equally unspecific |
| Negative int, zero | Any | k | No | Negative or zero input are invalid |

Note: *k* is a positive integer parameter.