

# Proteograph

Typst proteomics package

v0.2.1      November 11, 2025

<https://codeberg.org/olangella/proteograph>

Olivier Langella

## ABSTRACT

**proteograph** is a package defining [Typst](#) functions dedicated to proteomics.

## CONTENTS

Reference .....	2
XIC (eXtracted Ion Chromatogram) plot .....	2
XIC plot full documentation .....	3
Retention time alignment .....	7
Retention time alignment full documentation .....	8
MS2 spectra .....	10
MS2 spectra full documentation .....	17
Protein sequence .....	20
Protein sequence documentation .....	20
Protein diagram .....	21
Protein diagram documentation .....	21
Isotope pattern plot .....	22

## REFERENCE

### XIC (eXtracted Ion Chromatogram) plot

#### Example :

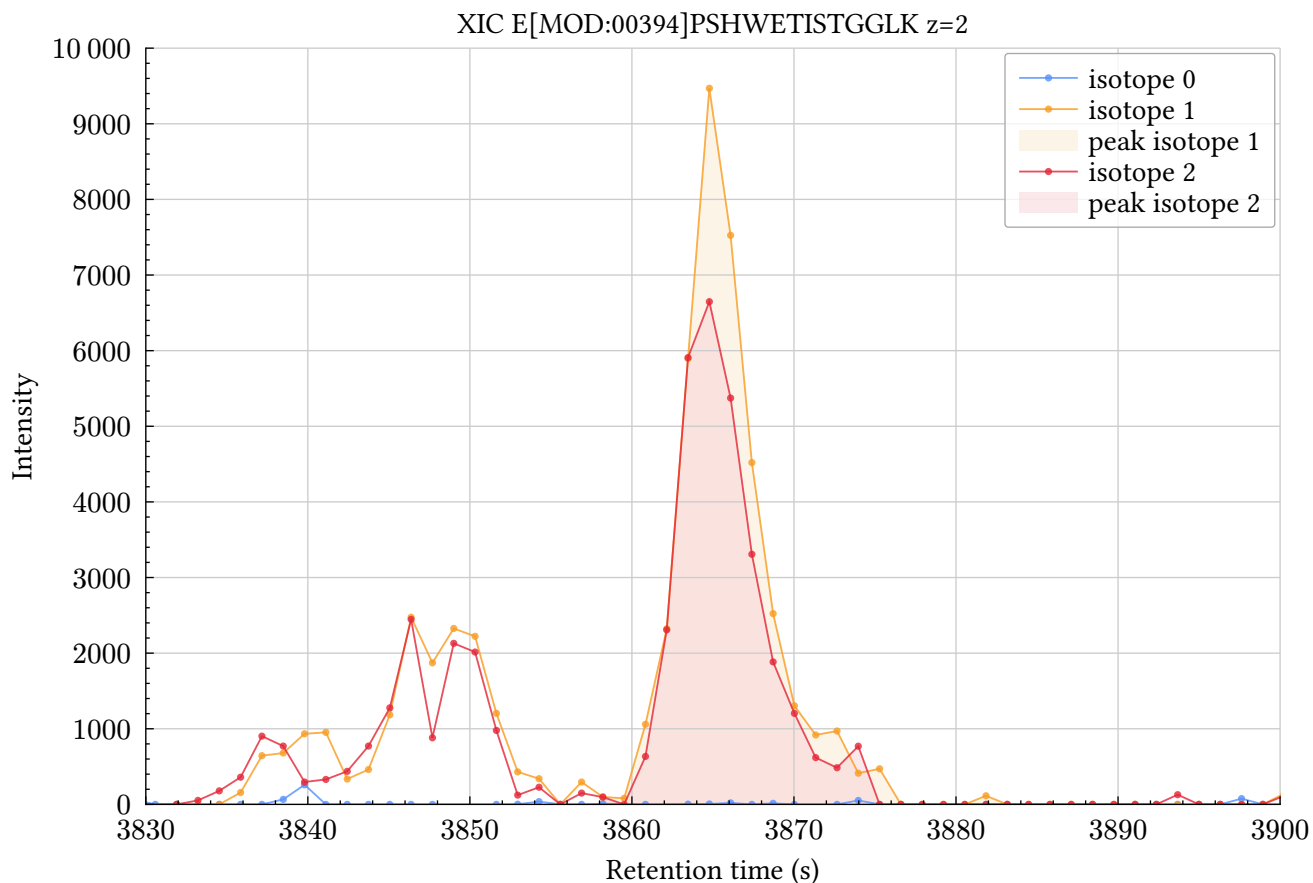
```
#import "@preview/tegraph:0.2.1": *

#let data_json = json("../examples/data/pepalb35.json")

#let xic0 = data_json.found_list_first_pass.first().xics.first().trace
#xic0.insert("title", "isotope 0")
#let xic1 = data_json.found_list_first_pass.first().xics.at(1).trace
#xic1.insert("title", "isotope 1")
#xic1.insert("peak-
begin", data_json.found_list_first_pass.first().xics.at(1).peak.rt.first())
#xic1.insert("peak-
end", data_json.found_list_first_pass.first().xics.at(1).peak.rt.last())
#let xic2 = data_json.found_list_first_pass.first().xics.at(2).trace
#xic2.insert("title", "isotope 2")
#xic2.insert("peak-
begin", data_json.found_list_first_pass.first().xics.at(2).peak.rt.first())
#xic2.insert("peak-
end", data_json.found_list_first_pass.first().xics.at(2).peak.rt.last())

#xic-plot(height: 10cm, title: "XIC E[MOD:00394]PSHWETISTGGLK z=2", rt-range: (3830,
3900), max-intensity: 10000, xic0, xic1, xic2)
```

## Result :



## XIC plot full documentation

- [xic-isotope-plot\(\)](#)
- [xic-plot\(\)](#)

## xic-isotope-plot

Generates a XIC plot separating istope pattern by charge

### Parameters

```
xic-isotope-plot(  
  width: length relative ,  
  height: length relative ,  
  rt-range: auto array ,  
  max-intensity: none float ,  
  title: content ,  
  ..xic-item: dictionary  
) -> content
```

**width**   `length` or `relative`

The width of the diagram. This can be

- A `length` ; in this case, it defines just the width of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the width of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded width, e.g., a page with `width: auto` .

Default: `15cm`

**height**   `length` or `relative`

The height of the diagram. This can be

- A `length` ; in this case, it defines just the height of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the height of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded height, e.g., a page with `height: auto` .

Default: `10cm`

**rt-range**   `auto` or `array`

Data limits along the x-axis (retention time in seconds). Expects auto or a tuple (min, max) where min and max may individually be auto

Default: `auto`

**max-intensity**   `none` or `float`

Maximum intensity to display. **Optional.**

Example: `30000`

Default: `none`

**title**   `content`

Graph title

Default: `none`

**..xic-item** dictionary

dictionary containing XIC data

```
1  /// XIC dictionary structure typc
2  #let xic_item = (
3    /// Array of retention times values
4    "x": (),
5    /// Array of intensity values
6    "y": (),
7    /// Title for this XIC
8    "title": "title for this XIC"
9    /// Detected peak retention time start. *Optional*.
10   "peak-begin": 3500,
11   /// Detected peak retention time end. *Optional*.
12   "peak-end": 3500,
13   /// Charge.
14   "charge": none,
15   /// Isotope number.
16   "isotope": none,
17 )
```

## xic-plot

Generates a XIC plot.

### Parameters

```
xic-plot(
  width: length relative,
  height: length relative,
  rt-range: auto array,
  max-intensity: none float,
  title: content,
  ..xic-item: dictionary
) -> content
```

**width** length or relative

The width of the diagram. This can be

- A `length` ; in this case, it defines just the width of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the width of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded width, e.g., a page with `width: auto` .

Default: `15cm`

**height** `length` or `relative`

The height of the diagram. This can be

- A `length`; in this case, it defines just the height of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the height of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded height, e.g., a page with `height: auto`.

Default: `10cm`

**rt-range** `auto` or `array`

Data limits along the x-axis (retention time in seconds). Expects auto or a tuple (min, max) where min and max may individually be auto

Default: `auto`

**max-intensity** `none` or `float`

Maximum intensity to display. **Optional**.

Example: `30000`

Default: `none`

**title** `content`

Graph title

Default: `none`

**..xic-item** `dictionary`

dictionary containing XIC data

```
1  /// XIC dictionary structure typc
2  #let xic_item = (
3    /// Array of retention times values
4    "x": (),
5    /// Array of intensity values
6    "y": (),
7    /// Title for this XIC
8    "title": "title for this XIC"
9    /// Detected peak retention time start. *Optional*.
10   "peak-begin": 3500,
11   /// Detected peak retention time end. *Optional*.
12   "peak-end": 3500,
13 )
```

## Retention time alignment

The alignment of retention times between MS runs done with MassChroQ3 are represented as follow in the JSON output

*MS run alignment data*

```
{
  "corrections": {
    "msrun1": {
      "aligned": [],
      "ms2_delta_rt": {
        "x": [],
        "y": []
      },
      "ms2_mean": [],
      "ms2_median": [],
      "original": []
    },
    "msrun2": {
      "aligned": [],
      "ms2_delta_rt": {
        "x": [],
        "y": []
      },
      "ms2_mean": [],
      "ms2_median": [],
      "original": []
    },
    "msrun3": {
      "aligned": [],
      "ms2_delta_rt": {
        "x": [],
        "y": []
      },
      "ms2_mean": [],
      "ms2_median": [],
      "original": []
    },
    "msrun_ref": "msrun4"
  }
}
```

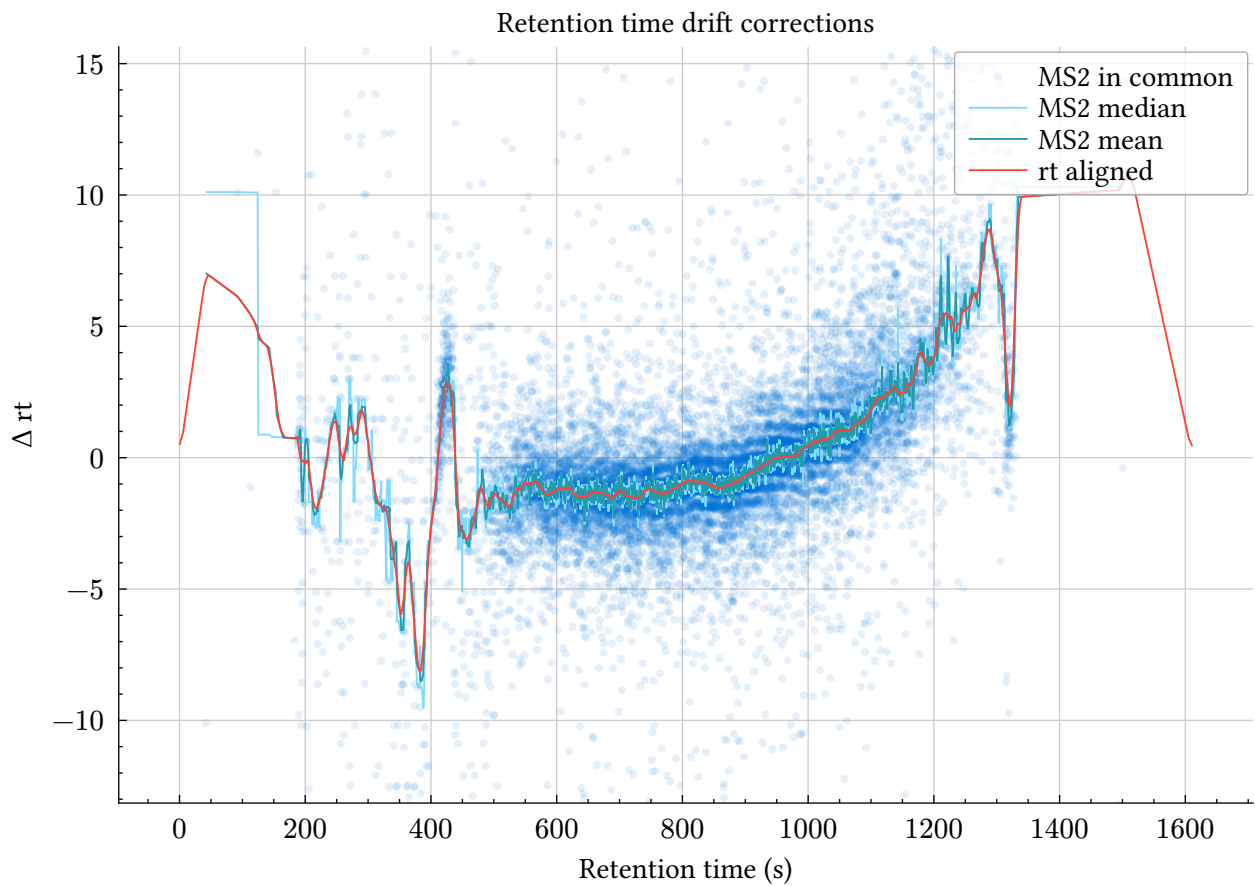
### Example :

```
#import "@preview/teproteograph:0.2.1": *

#let rt_align = json("../examples/data/one_alignment.json")
#rtalign-plot(title: "Retention time drift corrections", ylim: auto,
  ms2-delta-rt: rt_align.ms2_delta_rt,
  aligned: rt_align.aligned,
  ms2-mean: rt_align.ms2_mean,
  ms2-median: rt_align.ms2_median,
```

```
original: rt_align.original)
```

## Result :



## Retention time alignment full documentation

- [rtalign-plot\(\)](#)

### rtalign-plot

retention time delta plot between MS runs plot.

#### Parameters

```
rtalign-plot(  
  width: length relative,  
  height: length relative,  
  xlim: auto array,  
  ylim: auto array,  
  title,  
  original,  
  aligned,  
  ms2-delta-rt,  
  ms2-mean,  
  ms2-median  
) -> content
```



**width**   `length` or `relative`

The width of the diagram. This can be

- A `length` ; in this case, it defines just the width of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the width of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded width, e.g., a page with `width: auto` .

Default: `15cm`

**height**   `length` or `relative`

The height of the diagram. This can be

- A `length` ; in this case, it defines just the height of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the height of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded height, e.g., a page with `height: auto` .

Default: `10cm`

**xlim**   `auto` or `array`

Data limits along the x-axis (retention time in seconds). Expects `auto` or a tuple (min, max) where min and max may individually be `auto`

Default: `auto`

**ylim**   `auto` or `array`

Data limits along the y-axis (retention time delta in seconds). Expects `auto` or a tuple (min, max) where min and max may individually be `auto`

Default: `auto`

## MS2 spectra

### Simple MS2 spectra

The spectra is a dictionary containing 2 arrays of floats : mz and intensity

*spectra dictionary*

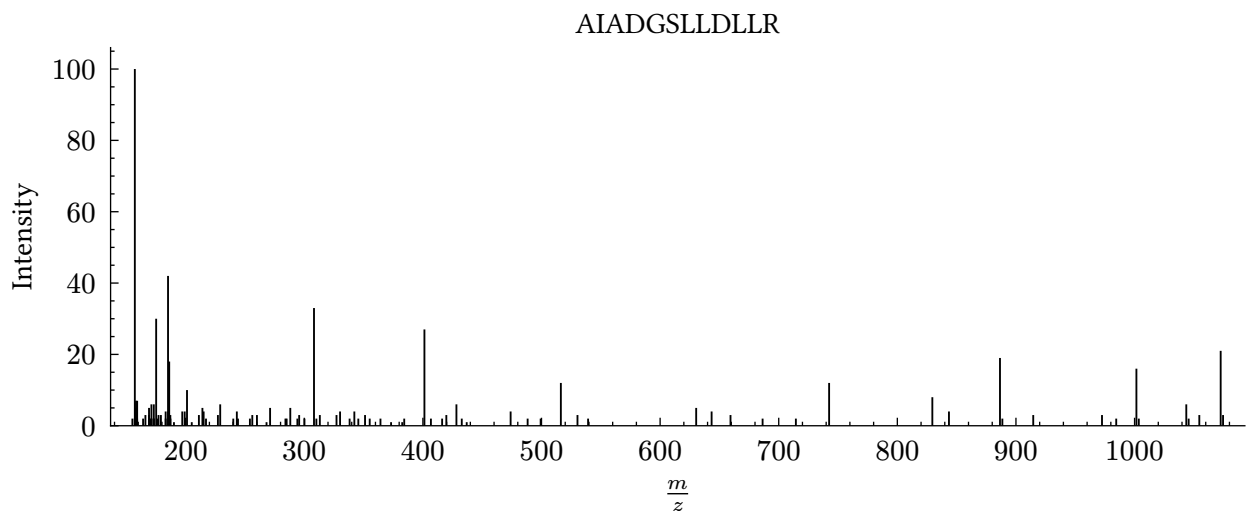
```
{
  "intensity": [
    2,
    2,
    100,
    5,
    7,
    2
  ],
  "mz": [
    155.081,
    157.097,
    157.133,
    158.092,
    158.137,
    159.076
  ]
}
```

### Example :

```
#import "@preview/tegraph:0.2.1": *

#let complete_psm = json("../examples/data/complete_psm.json")
#ms2spectra-plot(height: 5cm, title: "AIADGSLLDLLR", spectra: complete_psm.spectra)
```

### Result :



### MS2 spectra with ion annotations

This example shows an MS2 spectra with ion series annotation

The ion annotations are described in a dictionary :

*ion-series dictionary*

```
{
  "a": [
    {
      "charge": 1,
      "intensity": 100,
      "mz": 157.133,
      "mzth": 157.13353961189497,
      "size": 2
    }
  ],
  "b": [
    {
      "charge": 1,
      "intensity": 42,
      "mz": 185.128,
      "mzth": 185.12845423145498,
      "size": 2
    },
    {
      "charge": 1,
      "intensity": 3,
      "mz": 256.165,
      "mzth": 256.16556801702,
      "size": 3
    }
  ]
}
```

The first key is the ion series name as follow :

y, b, yp, ystar, y\*, yO, x, bstar, b\*, bO, a, astar, aO, c, z, bp

and any ion can be described by :

- [ion-description\(\)](#)

## ion-description

Ion description

### Parameters

```
ion-description(
  "charge": integer,
  "intensity": float,
  "mz": float,
  "mzth": float,
  "size": integer
)
```

**"charge"** integer

Ion charge

Default: 1

**"intensity"** float

Ion intensity

Default: none

**"mz"** float

Ion experimental mass on charge ratio

Default: none

**"mzth"** float

Ion theoretical mass on charge ratio. **Optional.**

Default: none

**"size"** integer

Ion size : number of residues for this fragment

Default: 1

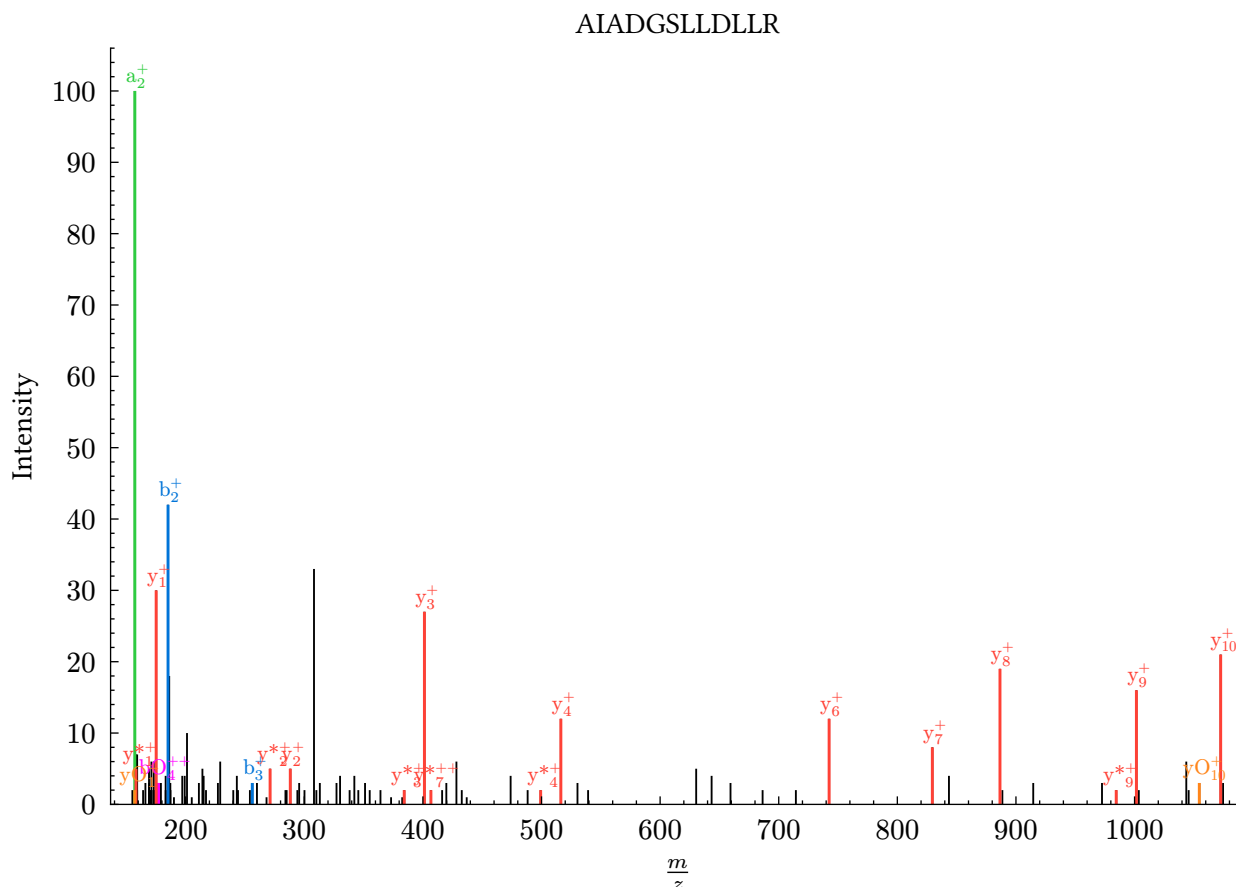
### Example :

```
#import "@preview/tegraph:0.2.1": *
```

```
#let complete_psm = json("../examples/data/complete_psm.json")
```

```
#ms2spectra-plot(title: "AIADGSLLDLLR", spectra: complete_psm.spectra, ion-series: complete_psm.ion-series)
```

## Result :



## MS2 spectra zoom (mz\_range and intensity)

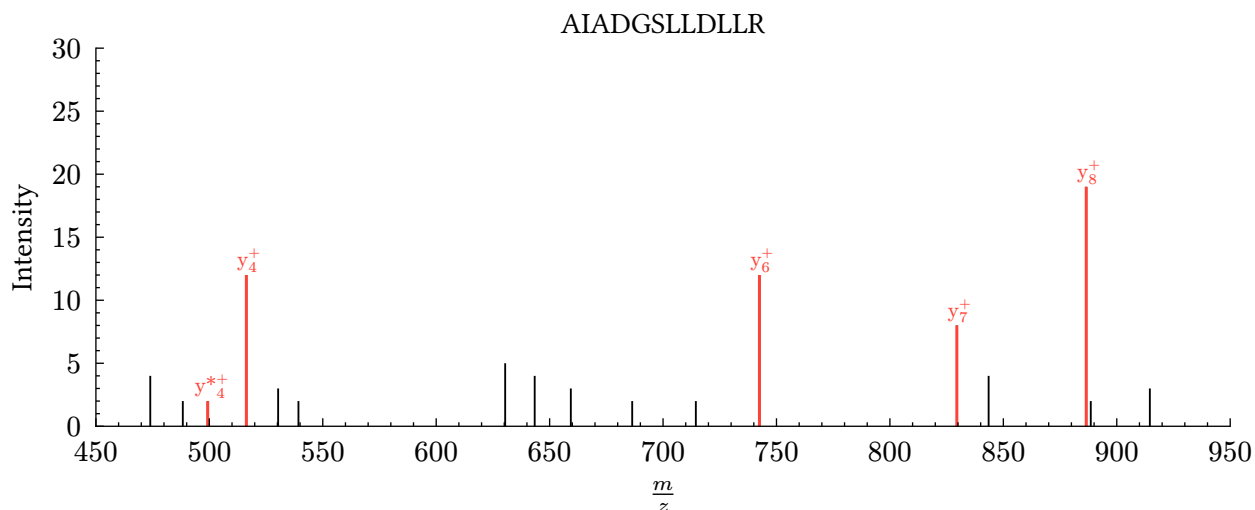
This example shows an MS2 spectra zoomed on a specific m/z range and maximum intensity.

## Example :

```
#import "@preview/teotograph:0.2.1": *

#let complete_psm = json("../examples/data/complete_psm.json")
#ms2spectra-plot(width: 15cm, height: 5cm, title: "AIADGSLLDLLR", spectra:
complete_psm.spectra, ion-series: complete_psm.ion-series, mz-range: (450, 950), max-
intensity: 30)
```

## Result :



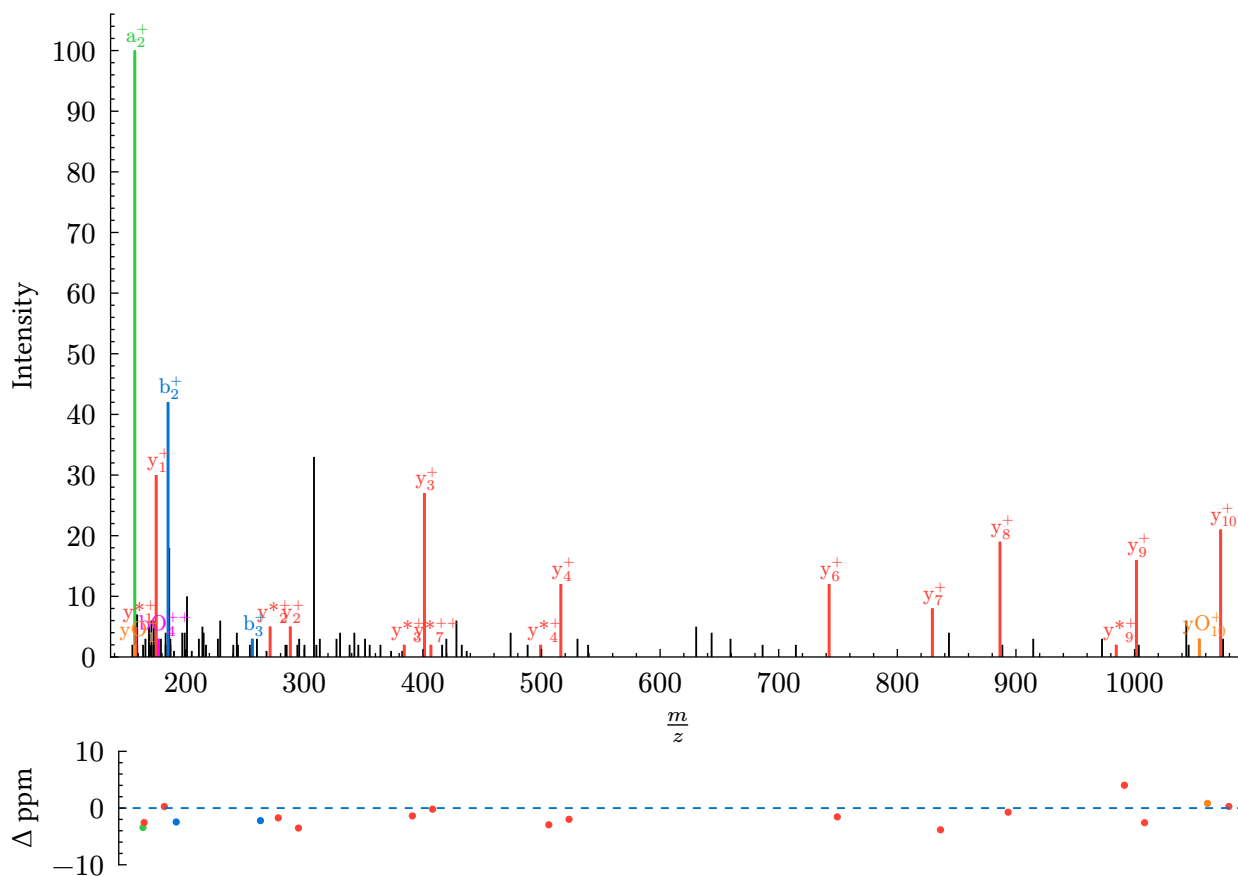
## MS2 spectra with ion annotations and MS2 fragments mass delta

This example shows an MS2 spectra with ion series annotation

### Example :

```
#import "@preview/teotegraph:0.2.1": *  
  
#let complete_psm = json("../examples/data/complete_psm.json")  
#ms2spectra-plot(width: 15cm, height: 10cm, spectra: complete_psm.spectra, ion-series:  
complete_psm.ion-series, delta-fragments: true)
```

**Result :**



## MS2 spectra with ion annotations and highlighted mass delta between peaks

The space between two peaks can be highlighted and manually annotated using an array defined as follow

*Highlight space between peaks array*

```
[
{
    "ion": "y",
    "mz": [401.287, 516.313],
    "level": 1,
    "label": "D"
},
{
    "ion": "y",
    "mz": [742.481, 829.511],
    "level": 1,
    "label": "S"
},
{
    "ion": "y",
    "mz": [829.511, 886.535],
    "level": 1,
    "label": "G"
},
{
```

```

    "ion": "y",
    "mz": [886.535,1001.56],
    "level": 1,
    "label": "D"
  }
]

```

## Example :

```
#import "@preview/proteograph:0.2.1": *
```

```

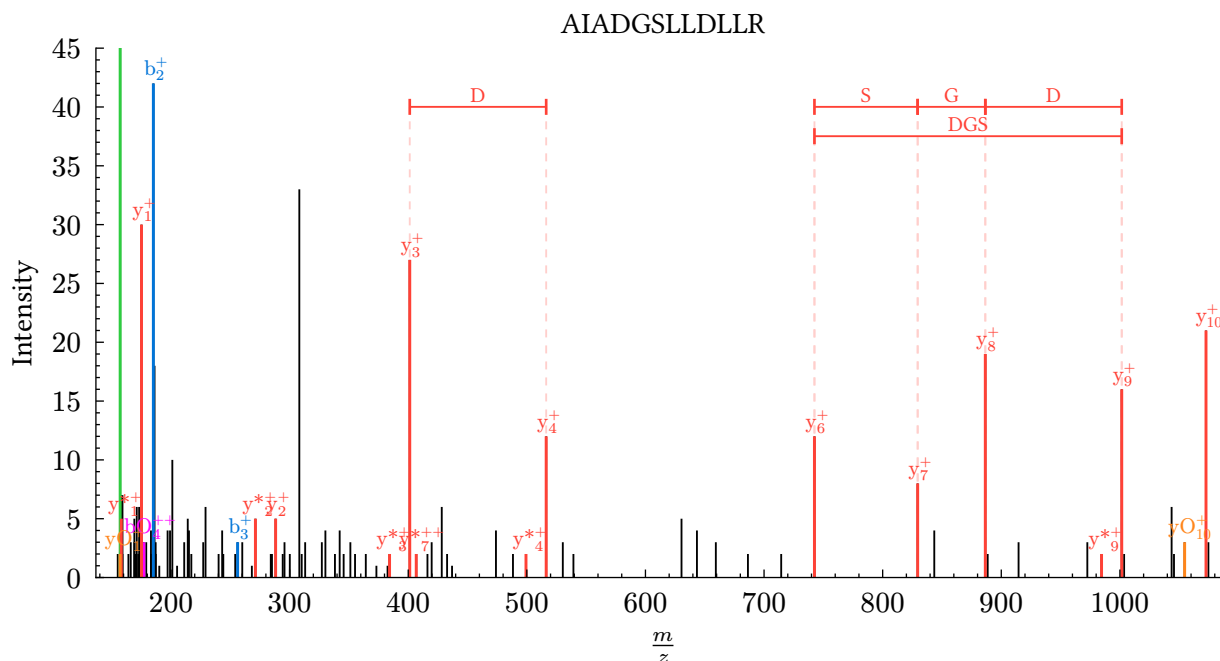
#let delta_arr = (
  (
    "ion": "y",
    "mz": (401.287,516.313),
    "level": 1,
    "label": "D"
  ),
  (
    "ion": "y",
    "mz": (742.481,829.511),
    "level": 1,
    "label": "S"
  ),
  (
    "ion": "y",
    "mz": (829.511,886.535),
    "level": 1,
    "label": "G"
  ),
  (
    "ion": "y",
    "mz": (886.535,1001.56),
    "level": 1,
    "label": "D"
  ),
  (
    "ion": "y",
    "mz": (742.481,1001.56),
    "level": 2,
    "label": "DGS"
  )
)

#let complete_psm = json("../examples/data/complete_psm.json")
#ms2spectra-plot(width: 15cm, height: 7cm, max-intensity: 45, title: "AIADGSLLDLLR",
spectra: complete_psm.spectra, ion-series: complete_psm.ion-series, delta: delta_arr)

```



## Result :



## MS2 spectra full documentation

- [ms2spectra-plot\(\)](#)

### ms2spectra-plot

Generates an annotated MS2 spectra plot.

#### Parameters

```
ms2spectra-plot(  
    width: length relative,  
    height: length relative,  
    title: str content,  
    mz-range: none array,  
    max-intensity: none float,  
    spectra: none dictionary,  
    ion-series,  
    delta,  
    delta-fragments: bool  
) -> content
```

**width**    length or relative

The width of the diagram. This can be

- A `length` ; in this case, it defines just the width of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the width of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded width, e.g., a page with `width: auto` .

Default: `15cm`

**height** `length` or `relative`

The height of the diagram. This can be

- A `length` ; in this case, it defines just the height of the data area, excluding axes, labels, title etc.
- A `ratio` or `relative` where the ratio part is relative to the height of the parent that the diagram is placed in. This is not allowed if the parent has an unbounded height, e.g., a page with `height: auto` .

Default: `10cm`

**title** `str` or `content`

Shows the plot title. **Optional.**

Default: `none`

**mz-range** `none` or `array`

m/z range to display. **Optional.**

Example: (`450`, `950`)

Default: `none`

**max-intensity** `none` or `float`

maximum intensity to display. **Optional.**

Example: `30000`

Default: `none`

**spectra** `none` or `dictionary`

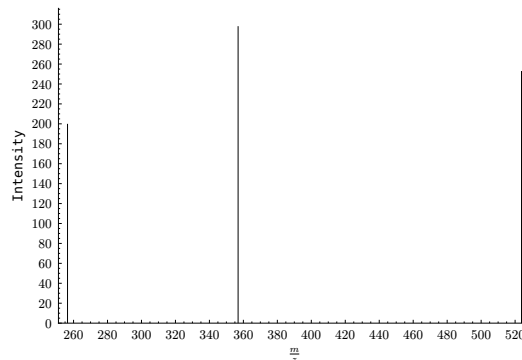
Mass spectra values. **Optional.**

a dictionary with the keys `mz` (array of mass to charge ratios) and `intensity` (array of intensities) with the same length for each array

(e.g., `spectra: (mz: (256.45, 356.89, 523.78), intensity: (200, 298, 253))`)

```
1 #import "@preview/
  proteograph:0.2.1": *
2 #set text(size: 12pt)
  #ms2spectra-plot( spectra: (mz:
3 (256.45, 356.89, 523.78), intensity:
  (200, 298, 253)))
```

typ



Default: `none`

**delta-fragments** `bool`

Whether to clip the matched ion mass delta to the plot. **Optional.**

Default: `false`

## Protein sequence

### Example :

```
#import "@preview/tegraph:0.2.1": *
```

```
#protein-sequence(line-length: 60,
```

```
"MASTKAPGPGEKHHSIDAQLRQLVPGKVSDDKLEIYDALLVDRFLNQLDHLGPSLREFVQECYEVSADYEGKGDITKLGLGAKLTGLAPADAILV
```

### Result :

```
0  MASTKAPGPGEKHHSIDAQLRQLVPGKVSDDKLEIYDALLVDRFLNQLDHLGPSLREF
60  VQECYEVSADYEGKGDITKLGLGAKLTGLAPADAILVASSILHMLNLANLAEVQIAHR
120 RRNSKLKKGFADEGSATTESDIEETLKRVLSEVGKSPPEVFEALKNQTVDLVFTAHTQ
180 SARRSLLQKNARIRNCLTQLNAKDITDDDKQELDEALQREIQAAFRTDEIRRAQPTPQDE
240 MRYGMSYIHETVWKGVPKFLRRVDTALKNIGINERLPYNVSLIRFSSWMGGDRDGNPRVT
300 PEVTRDVCLLARMMAANLYIDQIEELMFELSMWRCNDELRVRAEELHSSSGSKVTKYIE
360 FWKQIPPNEPYRVILGHVRDKLYNTRERARHLLASGVSEISAESSFTSIEEFLEPLELCY
420 KSLCDCGDKAIADGSLLDLLRQVFTFGLSLVKLDIRQESERHTDVIDAITTHLGIGSYRE
480 WPEDKRQEWLLSELRGKRPLLPDLPQTDEIADVIGAFHVLAEPPDSFGPYIISMATAP
540 SDVLAVELLQRECGVRQPLPVVPLFERLADLQSAPASVERLFSVDWYMDRIKQKQVMVG
600 YSDSGKDAGRLSAAWQLYRAQEEMAQVAKRYGVKLTLFHGRGGTVGRGGPHTLAISQP
660 PDTINGSIRVTVQGEVIEFCFGEEHLCFQTLQRFTAATLEHGMHPPVSPKPEWRKLMDEM
720 AVVATEEYRSVVVKEARFVEYFRSATPETEYGRMNIGSRPAKRRPGGGITTLEIPWIFS
780 WTQTRFHLPVWLGVGAFAIDKDVRNFQVLKEMYNEWPFVRVTLDLLEMVFAKGDPGI
840 AGLYDELLVAEELKPFQKLRDKYVETQQLLLQIAGHKDILEGDPFLKQGLVLRNPYITT
900 LNVFQAYTLKRIRDPNFKVTPQPPLSKEFADENKPAGLVKLNPASEYPPGLEDTLILTMK
960 GIAAGMQNTGA
```

## Protein sequence documentation

- [protein-sequence\(\)](#)

### protein-sequence

Display a protein sequence in a block with amino acid numbering

#### Parameters

```
protein-sequence(  
  line-length,  
  protein  
) -> content
```

#### line-length

The maximum length of amino acid to display in a line. **Optional**.

Default: 50

#### protein

Amino acid sequence (one letter code) of the protein to display

## Protein diagram

This figure is based on the [fletcher](#) package features.

### Example :

```
#import "@preview/telescope:0.2.1": *

#let tint(c) = (stroke: c, fill: rgb(..c.components().slice(0,3), 5%), inset: 1pt)

#protein-diag(line-length: 60,
"MASTKAPGPGGEKHHSIDAQLRQLVPGKVSDDKLEIYDALLVDRFLNILQDLHGPSLREFVQECYEVSADYEGKGDTTKLGELGAKLTGLAPADAILV
(pos: (1,5), style: tint(green)),
(pos: (12,17), style: tint(teal)),
(pos: (75,177), style: tint(red)),
)
```

### Result :

```
MASTKAPGPGGEKHHSIDAQLRQLVPGKVSDDKLEIYDALLVDRFLNILQDLHGPSLREF
VQECYEVSADYEGKGDTTKLGELGAKLTGLAPADAILVASSILHMLNLANLAEVQIAHR
RRNSKLLKGGFADEGSATTESDIEETLKRLVSEVGKSPPEVFEALKNOTVDLVFTAHPTQ
SARRSLLQKNARIRNCLTQLNAKDITDDDKQELDEALQREIQAAFRTDEIRRAQPTPQDE
MRYGMSYIHETVWKGVPKFLRRVDTALKNIGINERLPYNVSLIRFSSWMGGDRDGNPRVT
PEVTRDVCLLARMMAANLYIDQIEELMFELSMWRCNDELRVRAEELHSSSGSKVTKYIE
FWKQIPPNEPYRVILGHVRDKLYNTRERARHLLASGVSEISAESSFTSIEEFLEPLELCY
KSLCDCGDKAIADGSLLDLLRQVFTFGLSLVKLDIRQESERHTDVIDAITTHLGIGSYRE
WPEDKRQEWLLSELRGKRPLLPDLPQTDEIADVIGAFHVLAELPPDSFGPYIISMATAP
SDVLAVELLQRECGVRQPLPVVPLFERLADLQSAPASVERLFSVDWYMDRIKQKQVMVG
YSDSGKDAGRLSAAWQLYRAQEEMAQVAKRYGVKLTLFHGRGGTVGRGGPTHLAILSQP
PDTINGSIRVTVQGEVIEFCFGEELHLCFQTLQRFTAATLEHGMHPPVSPKPEWRKLMDEM
AVVATEEYRSVVVKEARFVEYFRSATPETEYGRMNIGSRPAKRRPGGGITTTLRAIPWIFS
WTQTRFHLPVWLGVGAFAFKFAIDKDVRNFQVLKEMYNEWPFVRVTLDLLEMVFAKGDPI
AGLYDELLVAEELKPFQKLRDKYVETQQLLLQIAGHKDILEGDPFLKQGLVLRNPYITT
LNVFQAYTLKRIRDPNFKVTPQPPLSKEFADENKPAGLVKLNPASEYPPGLEDTLILTMK
GIAAGMQNTGA
```

## Protein diagram documentation

- [protein-diag\(\)](#)

### protein-diag

Display a protein sequence in a block with amino acid numbering

#### Parameters

```
protein-diag(
  line-length: int,
  boxes: array,
  protein: str,
  ..box: dictionary
) -> content
```

**line-length** `int`

The maximum length of amino acid to display in a line. **Optional**.

Default: `50`

**boxes** `array`

Array of boxes to draw in the protein sequence **Optional**.

This can be used to visualize the position of one or more peptides

example for 3 peptides :

```
((pos: (1,5), style: tint(green)),(pos: (12,17), style: tint(teal)),(pos: (75,177), style: tint(red)))
```

Default: `()`

**protein** `str`

Amino acid sequence (one letter code) of the protein to display

**..box** `dictionary`

box item to draw in the protein sequence **Optional**.

This can be used to visualize the position of one or more peptides

example :

```
(pos: (1,5), style: tint(green))
```

```
1 #import "@preview/
2 proteograph:0.2.1": *
3 #protein-diag(line-length: 60,
4 "MASTKAPGPGGEKHHSIDAQLRQLV",
5 (pos: (1,5), style: (stroke: red,
6 inset: 1pt)),
7 (pos: (12,17), style: (stroke: teal,
8 inset: 3pt))
9 )
```



## Isotope pattern plot

### Example :

```
#import "@preview/proteograph:0.2.1": *
```

```
#isotope-pattern-plot(
  mz:(792.89009986061, 793.3914387289032, 793.8927444414951),
  intensity:(22000, 18000, 8000),
  th-ratio: ( 0.4182709626684006, 0.3428843140088085, 0.157722041765769),
  color: orange,
)
```

**Result :**

