

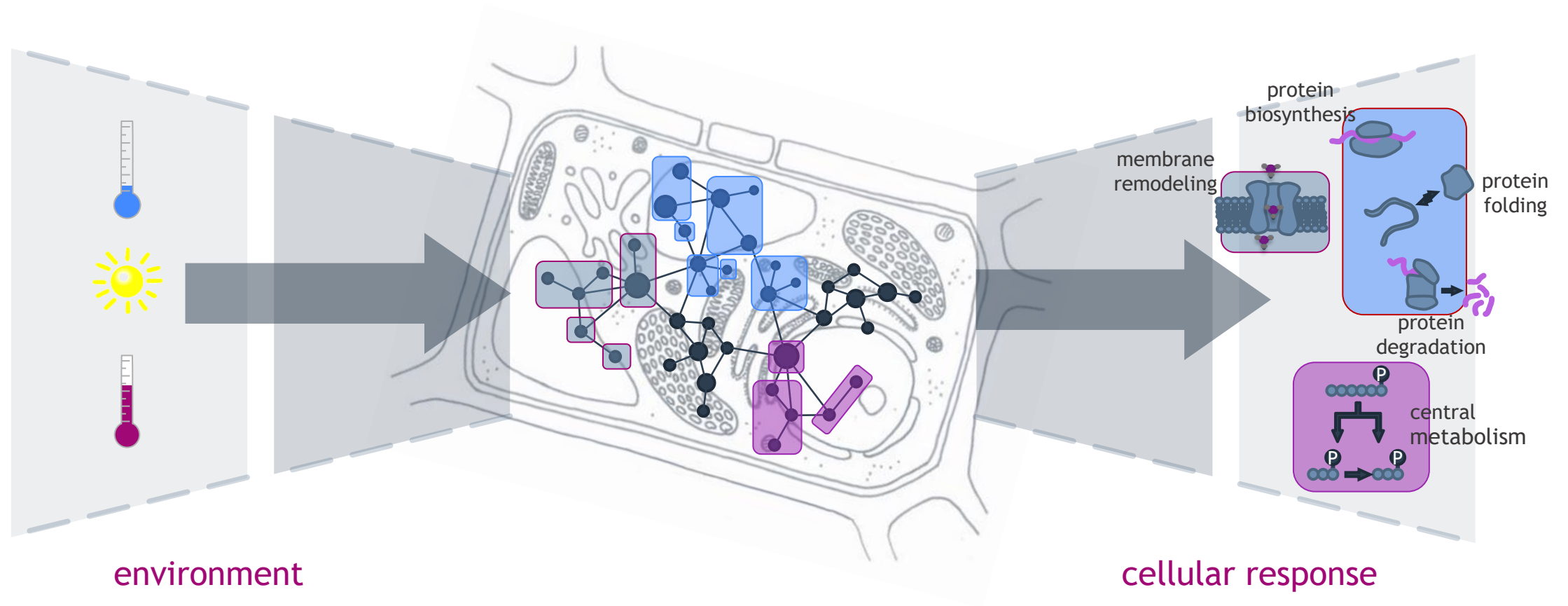
Unleashing the Analytical Power of F#: Empowering Biotechnological Data Science Education and Research

Data Science in F#
Berlin, 28-30 September 2023

Timo Mühlhaus



Computational Systems Biology



Computational Systems Biology Group

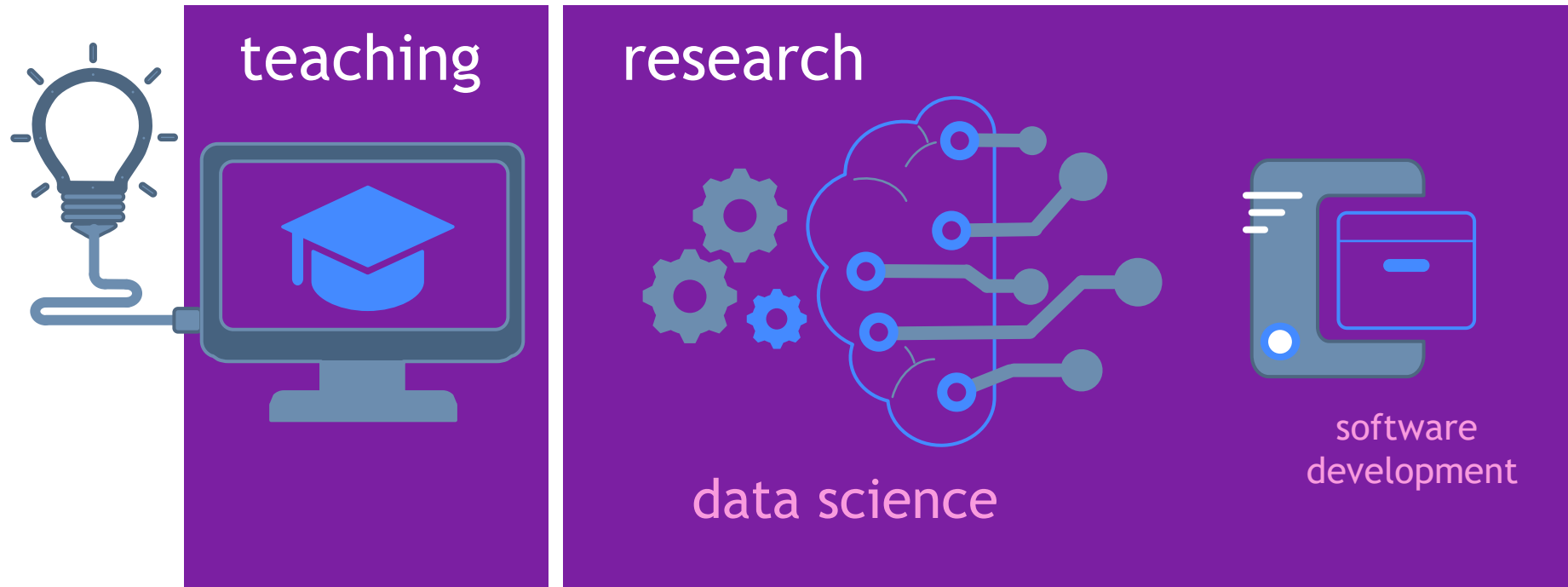


Computational Systems Biology at RPRU

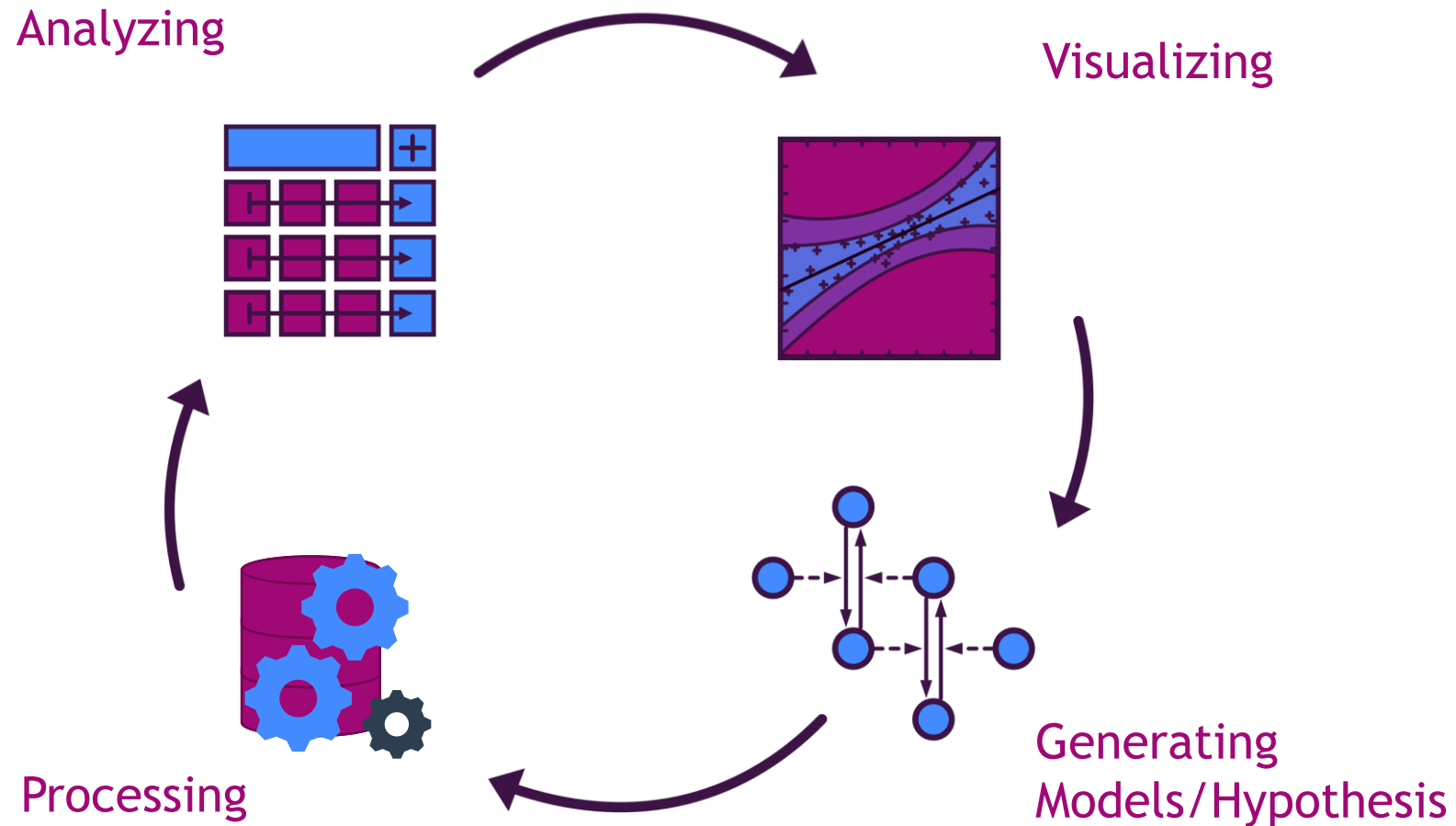
TU Rheinland-Pfälzische
Technische Universität
RP Kaiserslautern
Landau



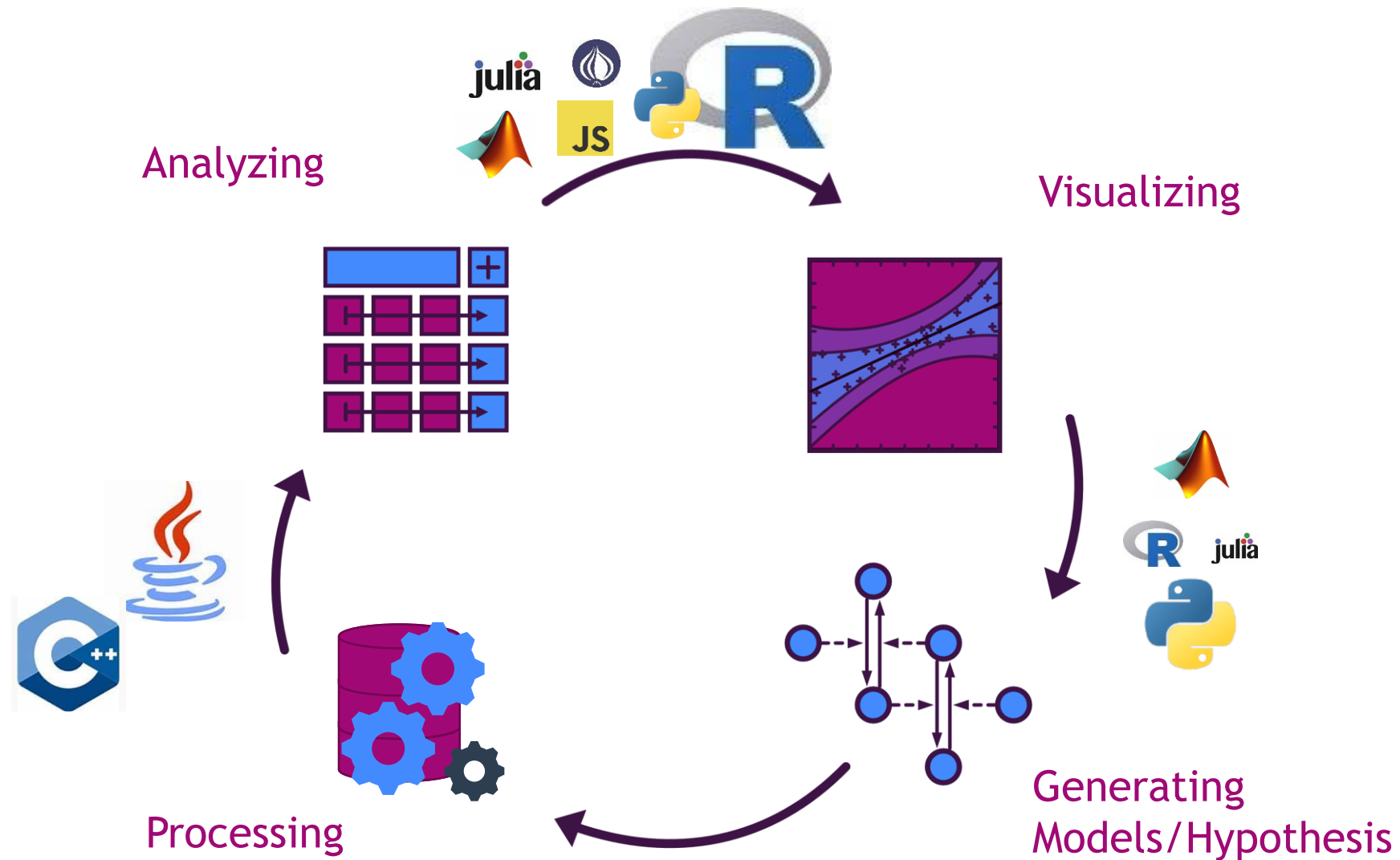
Tasks in Academia



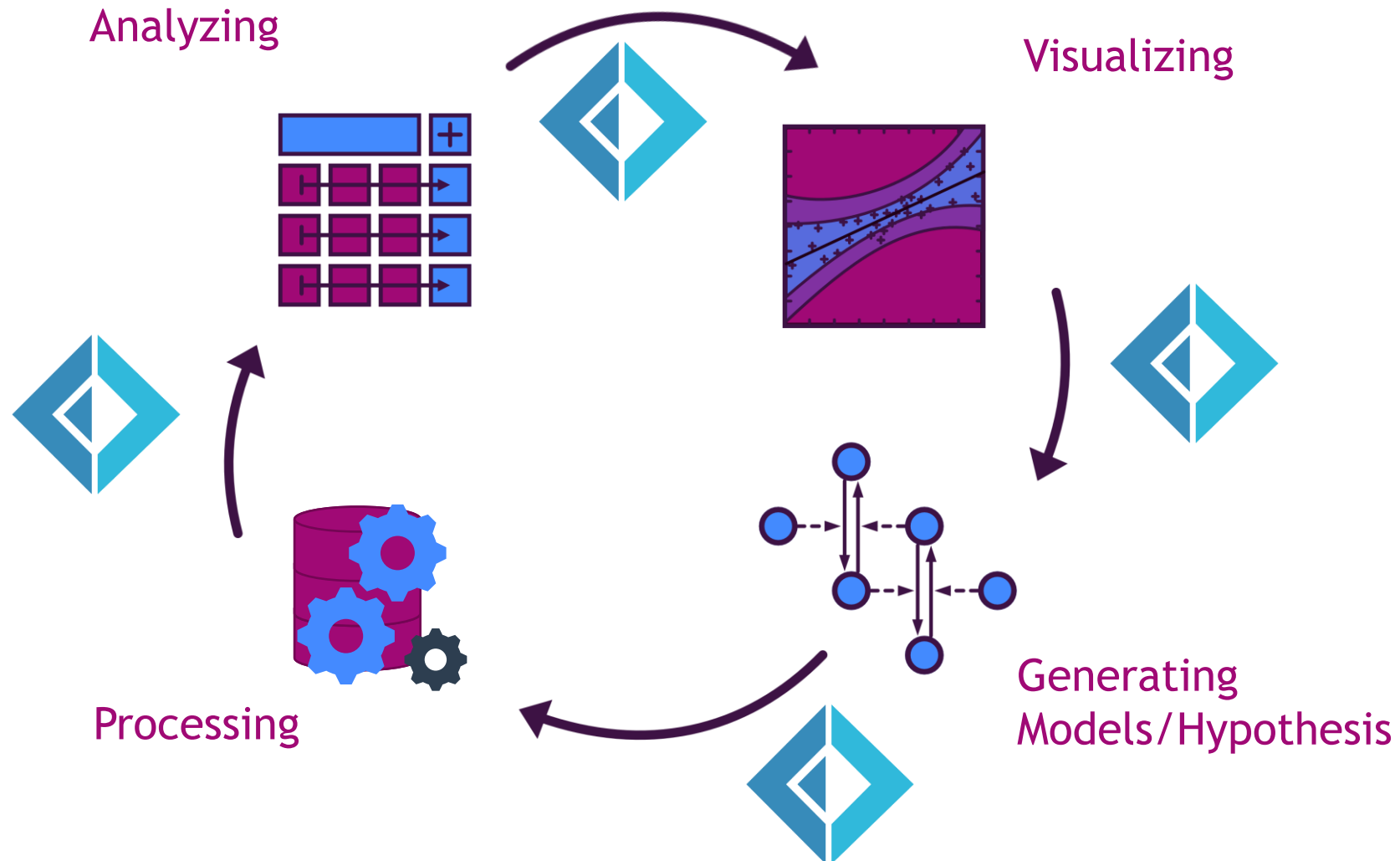
Data Science in Biology



Data Science in Biology

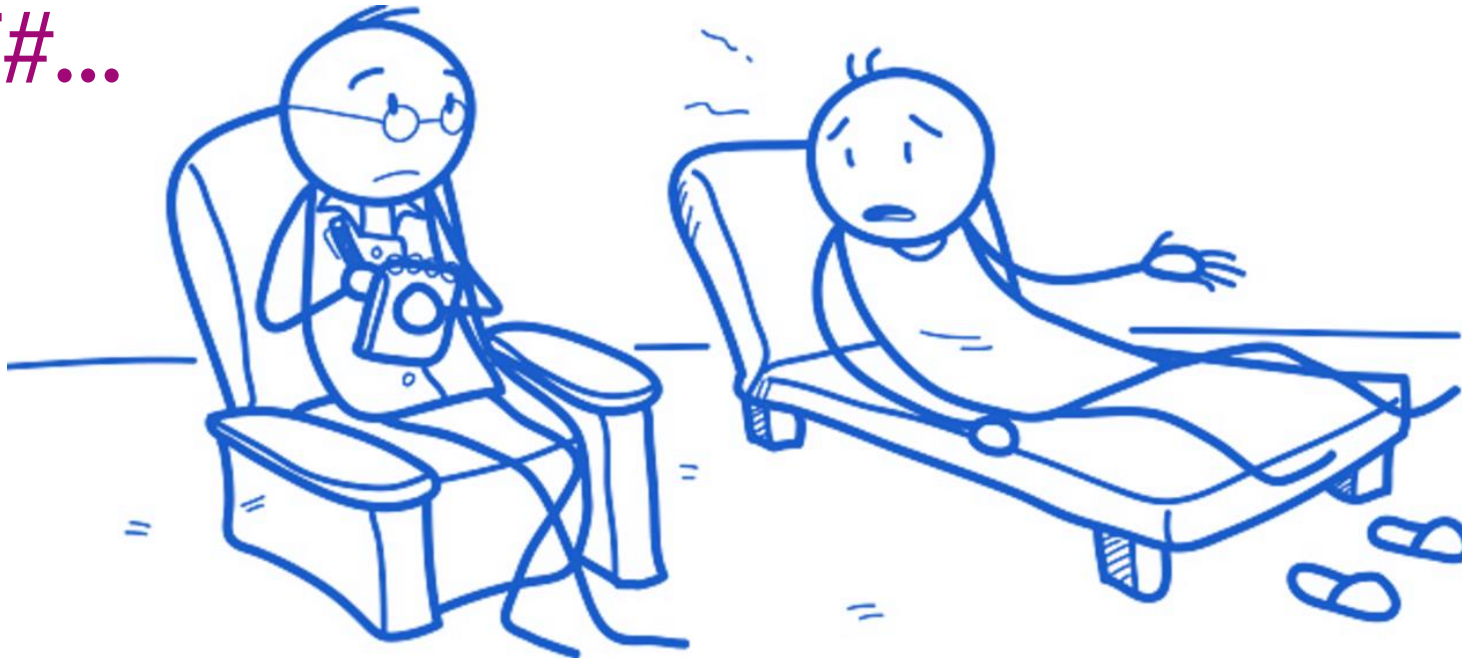


F# to Homogenize the 'Zoo' of Languages



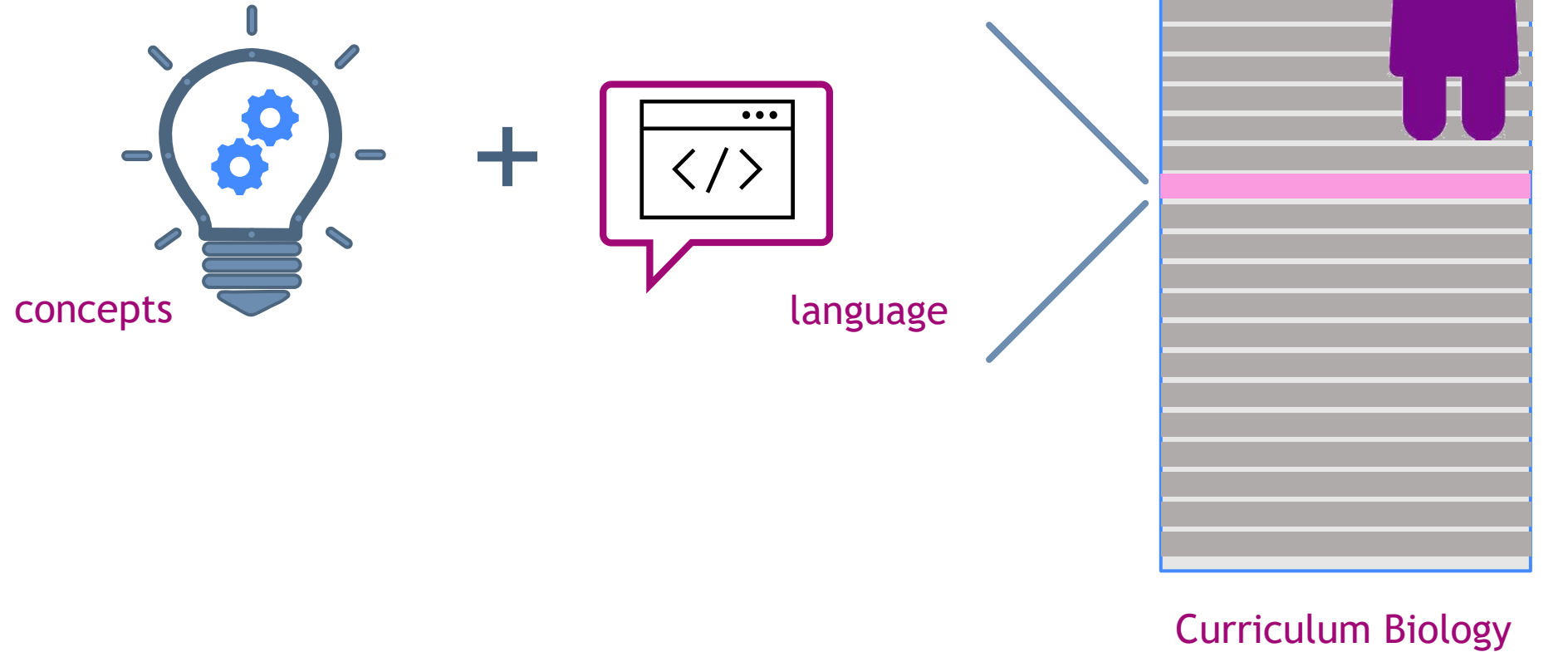
Biggest Question of the Students in Biology

Why F#...



...when all my friends use python?

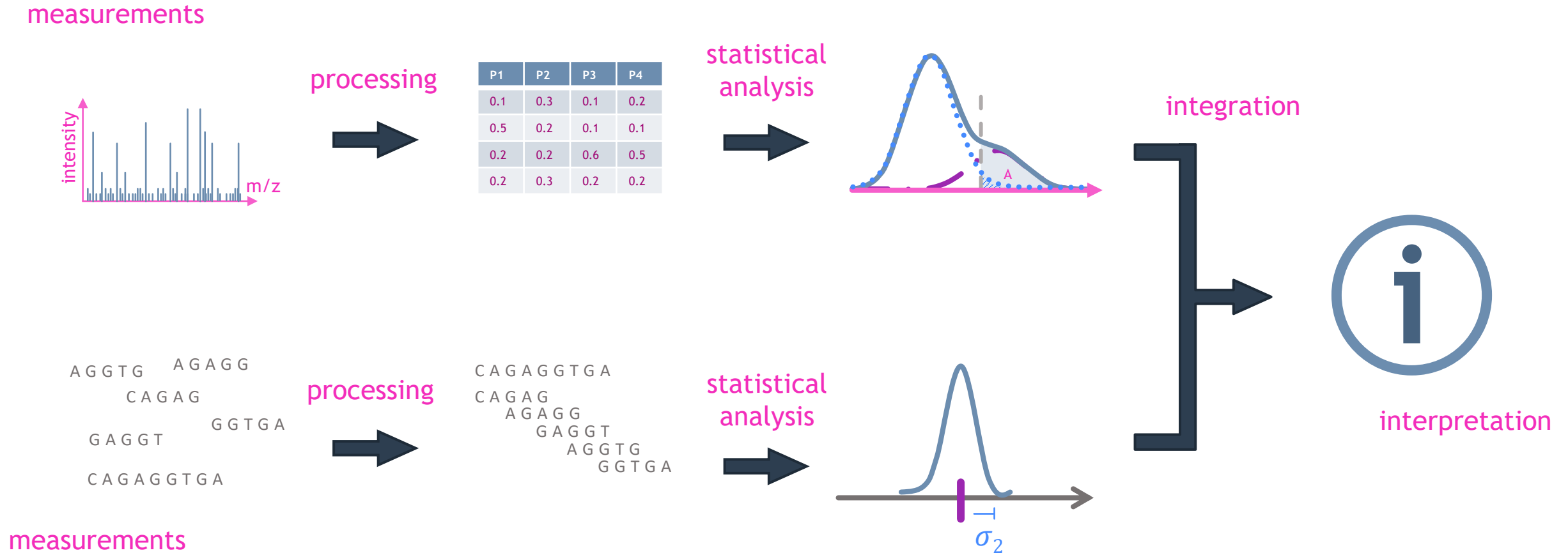
Workload of a Biology Student



...and of course ...

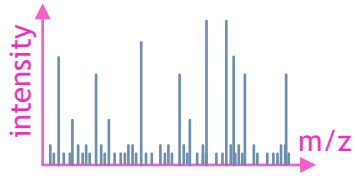


Take a look at a bioinformatics workflow



Take a look at a bioinformatics workflow

measurements

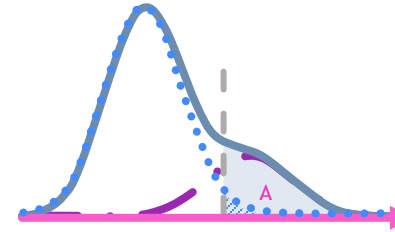


processing



P1	P2	P3	P4
0.1	0.3	0.1	0.2
0.5	0.2	0.1	0.1
0.2	0.2	0.6	0.5
0.2	0.3	0.2	0.2

statistical analysis



integration



interpretation



information

function(s)



information

function(s)



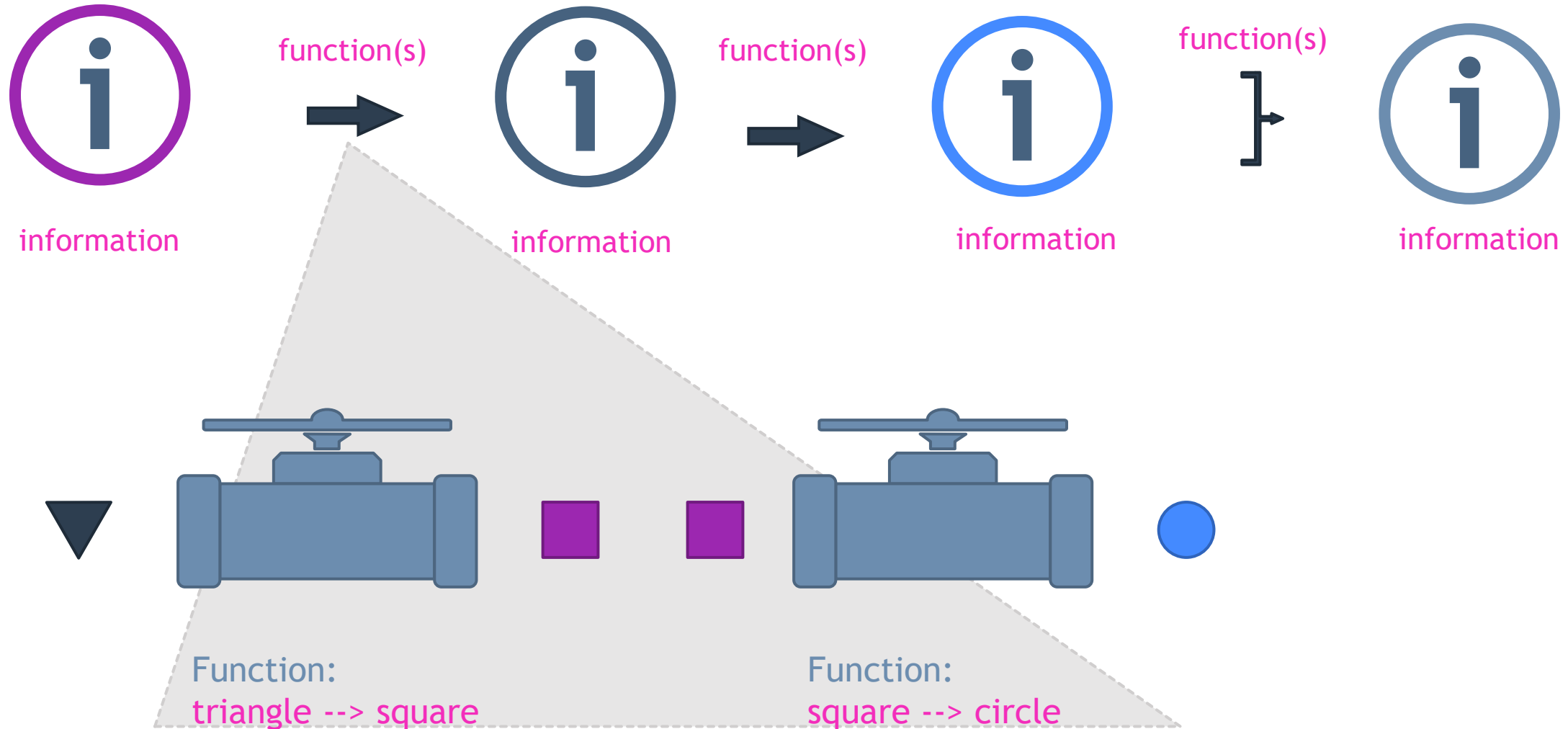
information

function(s)



information

Function Driven Transformation



Lecture Series F# for Biologists

Library > WPB

Title ▾ ▾

06:06

Vorlesung WPB III. 01
3 years ago

14:16

Vorlesung WPB II. 01
3 years ago

18:06

Vorlesung WPB I. 04
3 years ago

Vorlesung WPB IV. 03
Timo Mühlhaus

Funktionen als Ausgabewert

Parameter der "Factory"-Funktion

"Factory"-Funktion

"Innere" Funktion als Rückgabewert

Parameter der "Factory"-Funktion

Function1:
float --> (bool --> float)

Function2:
bool --> float

Signatur: hinter dem letzten "-->" steht die Signatur des Ausgabewertes

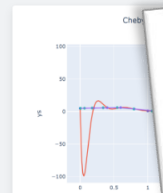
Load more...

Blog Posts as Resources for Students

Data Science Posts

Data science using the FSLab stack

2023-8-31

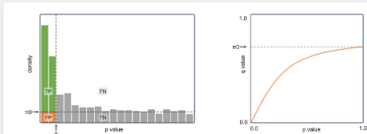


Chebyshev function and

This tutorial demonstrates how to use functions.

Posted on 2023-8-31 by Benedikt

2022-3-20

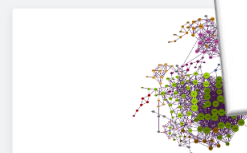


Multiple testing correction: q values

This tutorial explains the key concepts of q values and how to calculate them using FSharp.Stats.

Posted on 2022-3-20 by Benedikt Venn in Data Science

2021-12-7



Correlation network

This tutorial demonstrates how to generate a gene co-expression network

Clustering with FSharp.Stats II: hierarchical clustering

This tutorial demonstrates hierarchical clustering with FSharp.Stats and how to visualize the results with Plotly.NET.

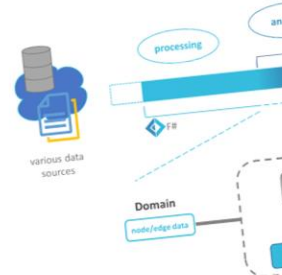
Posted on 2021-7-28 by Benedikt Venn in Data Science

<https://fslab.org/blog/posts/categories/Datascience.html>

Student Projects

FSharpGephiStreamer

FSharpGephiStreamer is intended to close the gap between F# and network visualization power of gephi into any kind of data science workflow. It uses a short Grammar which makes it possible to create a kind of relationship between these objects to edges of a graph independent from specific data structures/types.



Resources

Installation

See how to [setup FSharpGephiStreamer](#)

DeepSTABp: Predict protein thermal stability

Protein thermal stability is an important parameter for understanding protein function, designing stability of enzymes used in industrial processes. This web interface allows you to utilize the melting point of a protein of your choice!

Only three more steps are needed to get going:

- Provision a protein sequence or protein database in the FASTA format
- Specify a growth temperature
- Set the 'lysate' or 'cell' flag.

Input

Insert amino acid sequence(s) in FASTA format (with or without header)

Or upload a file

Start Plot

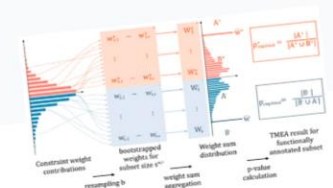
Growth temperature: 32

Select environment: ☒ Lysate ☐ Cell

Functionally Annotated Set (FAS) weight distributions

- `plotFASWeightDistribution` is an exploratory plot that visualizes the overall weight distributions of the given TMEA Characterizations, and adds detailed weight distributions of the FAS of interest on top of that. Additionally, annotations on the respective subplots show useful information about the FAS characterization.

```
tmeaRes  
|> TMEAResult.plotFASWeightDistribution  
true //use style presets  
0.05 //significance threshold for (corrected!) p values  
[1;2;3] //constraints to plot  
"signalling.light" //name of the FAS
```



Potential Time Course:

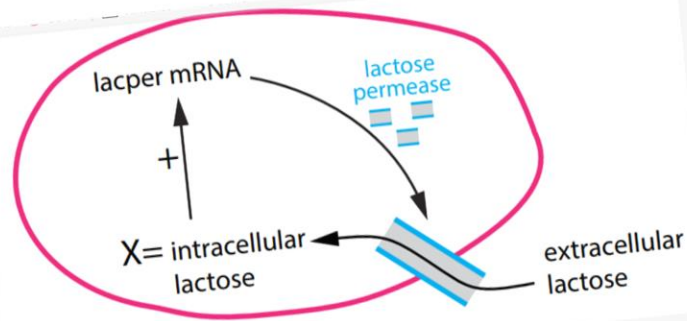
- `plotConstraintTimecourses` plots the constraint potential time courses of the given TMEA result:

```
tmeaRes  
|> TMEAResult.plotConstraintTimecourses true //true -> will use style presets
```

- `plotPotentialHeatmap` is a more visually pleasing version of above plot (it omits the baseline state per default):

```
tmeaRes  
|> TMEAResult.plotPotentialHeatmap true
```

Notebooks



Die Differenzialgleichung für die Konzentration von intrazellulärer Lactose kann somit folgendermaßen beschrieben werden. Es gibt einen unregulierten geringen Lactoseimport, außerdem einen regulierten Lactoseimport, dieser wird β genannt analog zur Proteinproduktion von β – Galactosid-Permease die nötig ist damit dieser regulierte Transport intrazelluläre Lactose zu mehr Import führt, kann hier von einer positiven Autoregulation gesprochen werden. Abschließend wird Lactose natürlich auch von der Zelle verbraucht, die genutzt wird wieder von der Lactosekonzentration abhängt. Die Differenzialgleichung ist somit

$$\text{Lactosekonzentration}_{\text{intrazellulär}}' = \frac{d\text{Lactosekonzentration}_{\text{intrazellulär}}}{dt} = \alpha + \beta * \frac{\text{Lactosekonzentration}_{\text{intrazellulär}}}{K_D + \text{Lactosekonzentration}_{\text{intrazellulär}}} - \gamma * \text{Lactosekonzentration}_{\text{intrazellulär}}$$

Da die Simulationen nur auf die richtige Differenzialgleichung achten, aber nicht auf die genutzten Namen der Parameter, verkürzen wir die Gleichung zu:

$$\text{inlac}' = \frac{d\text{inlac}}{dt} = \alpha + \beta * \frac{\text{inlac}}{K_D + \text{inlac}} - \gamma * \text{inlac}$$

Aufgabe 4.1

Setzen Sie die Gleichung für die intrazelluläre Lactosekonzentration in die nachfolgende Simulation ein (Bitte beachten Sie, dass Sie die Lactosekonzentration angeben müssen, ohne " ". Beachten Sie ebenfalls das α , β und γ als "alpha", "beta" und "gamma", ohne "", eingesetzt werden müssen und nicht als Symbole, und das wird).

```
//Konstanten
let alpha = 0.5 // Konstante für die Zunahme der Lactosekonzentration (unreguliert)
let beta = 10.0 // Konstante für die Zunahme der Lactosekonzentration (reguliert)
let gamma = 1.0 // Konstante für die Verringerung der Lactosekonzentration
let K_d = 5.0 // Dissoziationskonstante
```

```
// unsere DGL als Modell (Model)
let dP_dt : Model =
  fun P t ->
```

The notebook contains the following F# code:

```
let berechneEuler x0 y0 f h n =
  [|x0 .. h .. x0 + (n / (1. / h))|]
  > Array.scan (
    fun acc x ->
      acc + h * (f acc)
  ) y0

berechneEuler t0 y0 (fun t -> r * t) h n
```

Hier das Ganze als Diagramm visualisiert:

```
// eine Zwischenspeicherung unseres Ergebnisses:
let eulerWerte = berechneEuler t0 y0 (fun t -> r * t) h n

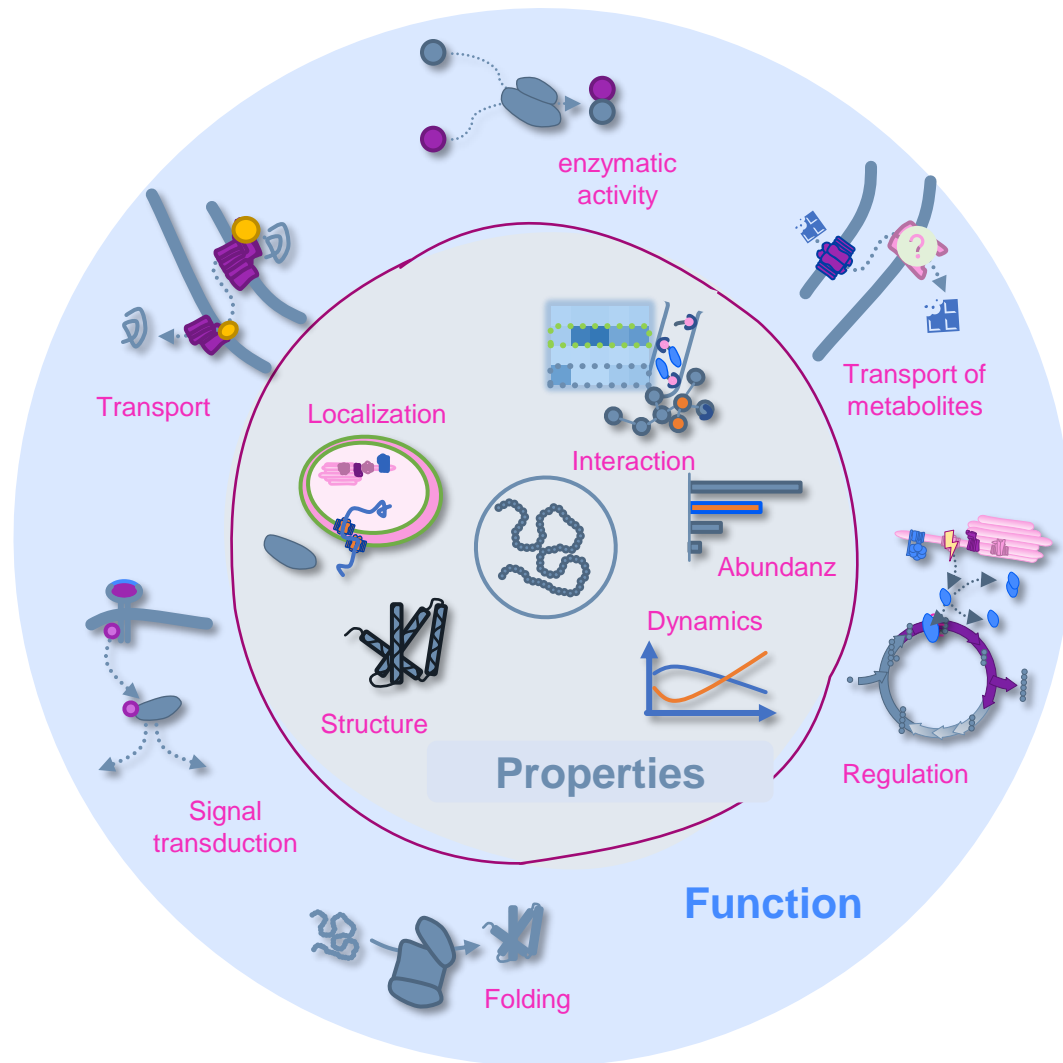
eulerWerte
|> Array.indexed
|> Chart.Point
|> Chart.withTitle "Euler-Verfahren, Schrittweite = 1, Anzahl Punkte = 10"
|> Chart.withXAxisStyle "t [min]"
|> Chart.withYAxisStyle "Bakteriendichte"
```

Aufgabe 1.3: Wie könnten Sie als Biologin einen solchen Startpunkt P experimentell bestimmen?

Antwort:

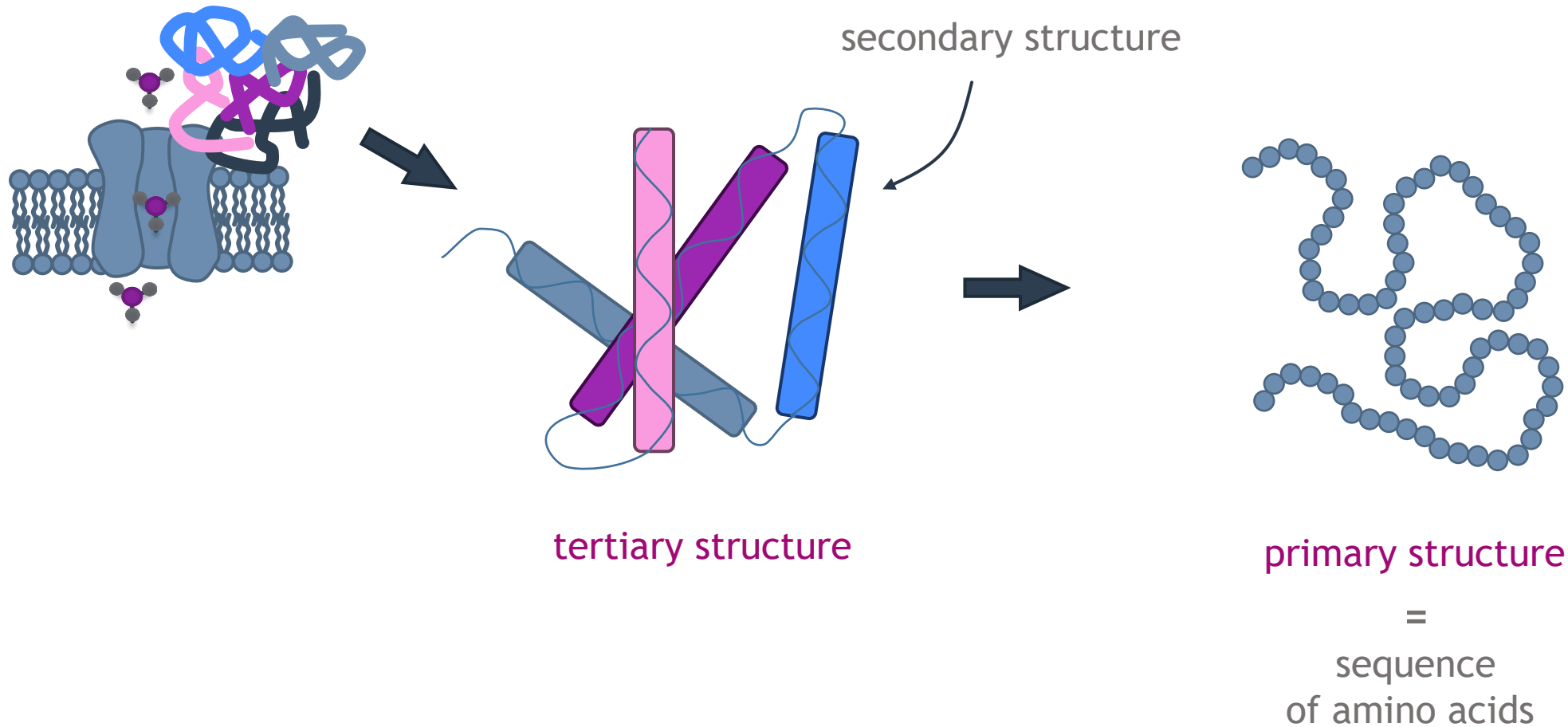
Betrachten Sie folgende Abbildung aus der Vorlesung:

Proteins: Key Players in Living Systems

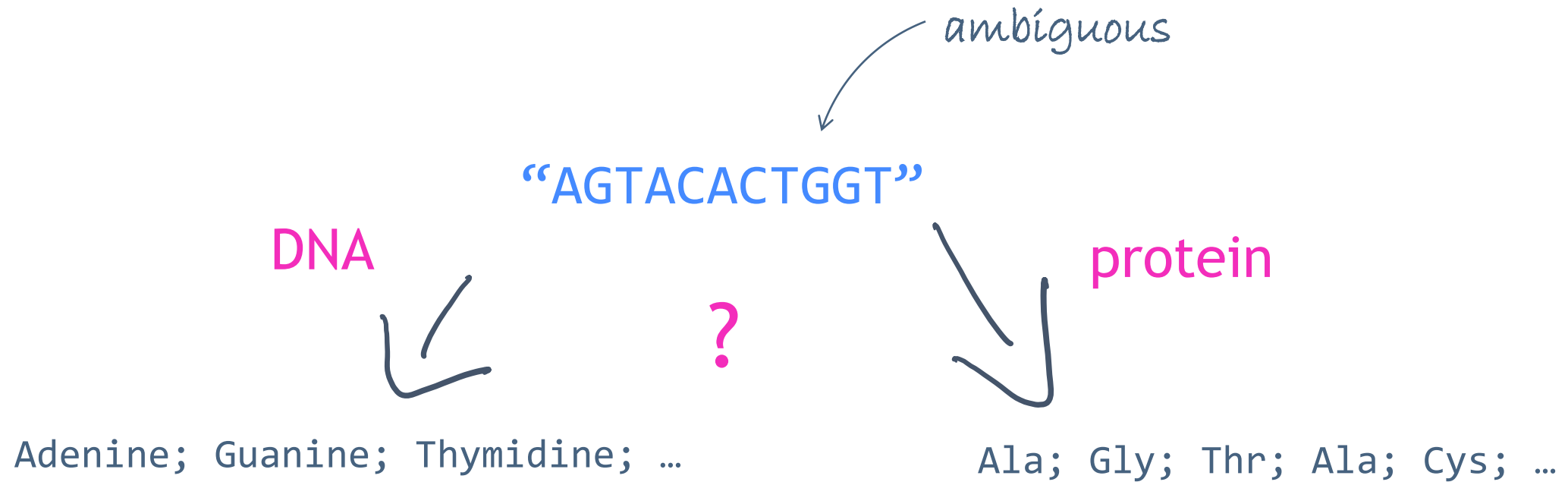


- ▶ Proteins are crucial for the **biotechnological production** of high-quality biomaterials
- ▶ Proteins are important targets for the **diagnosis and treatment** of diseases caused by protein malfunctions

All Information are Encoded in Biological Sequences

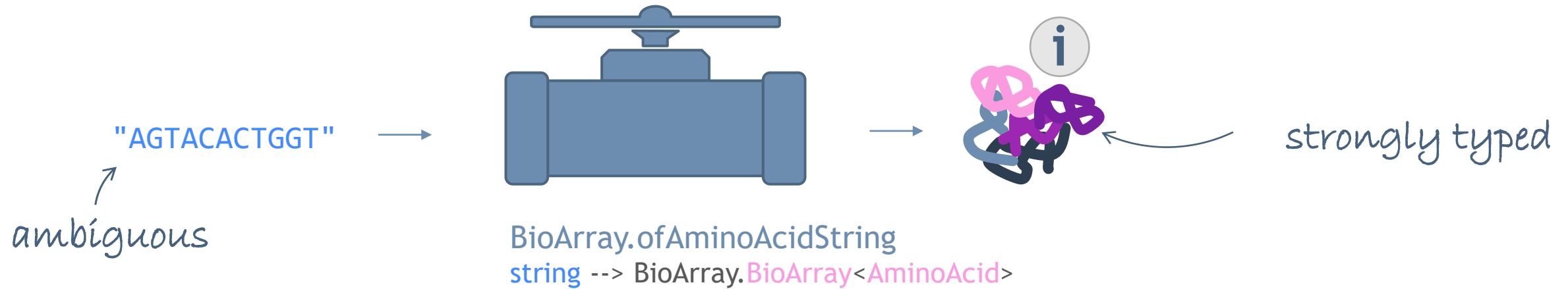


Working with Biological Sequences



- The representation of biological sequences is basically a sequence of characters.

Working with Biological Sequences



```
BioArray.ofAminoAcidString "AGTACACTGGT"
```

F#

```
val it : BioArray.BioArray<AminoAcid> =  
[|Ala; Gly; Thr; Ala; Cys; ...|]
```


BioFSharp - Computational Biology in F#

BioFSharp

fsharp.org [github page](#)

BioFSharp

BioFSharp aims to be a user-friendly library for Bioinformatics written in F#. It contains the basic data structures for common biological objects like amino acids and nucleotides based on chemical formulas and chemical elements.

Example

This example demonstrates using a function defined in BioFSharp library.

```
1: #r "BioFSharp.dll"
2: open BioFSharp
3:
4: /// Creates a BioSeq of the given peptide string
5: BioSeq.ofAminoAcidString "REYAHMIGMEYDTVQK"
6:
7: /// Creates a BioArray of the given peptide string
8: BioArray.ofAminoAcidString "REYAHMIGMEYDTVQK"
9:
10: /// Creates a BioList of the given peptide string
11: BioList.ofAminoAcidString "REYAHMIGMEYDTVQK"
```

Samples & documentation

The library comes with comprehensible documentation. It can include tutorials automatically generated from `*.fsx` files in [the content folder][content]. The API reference is automatically generated from Markdown comments in the library implementation.

- [Tutorial](#) contains a further explanation of this sample library.
- [API Reference](#) contains automatically generated documentation for all types, modules and functions in the library. This includes additional brief samples on using most of the functions.



BIOFSHARP

[Home page](#)

[Get Library via NuGet](#)

[Source Code on GitHub](#)

[License](#)

[Release Notes](#)

GETTING STARTED

[BioSequence](#)

[Spectrum centroidization](#)

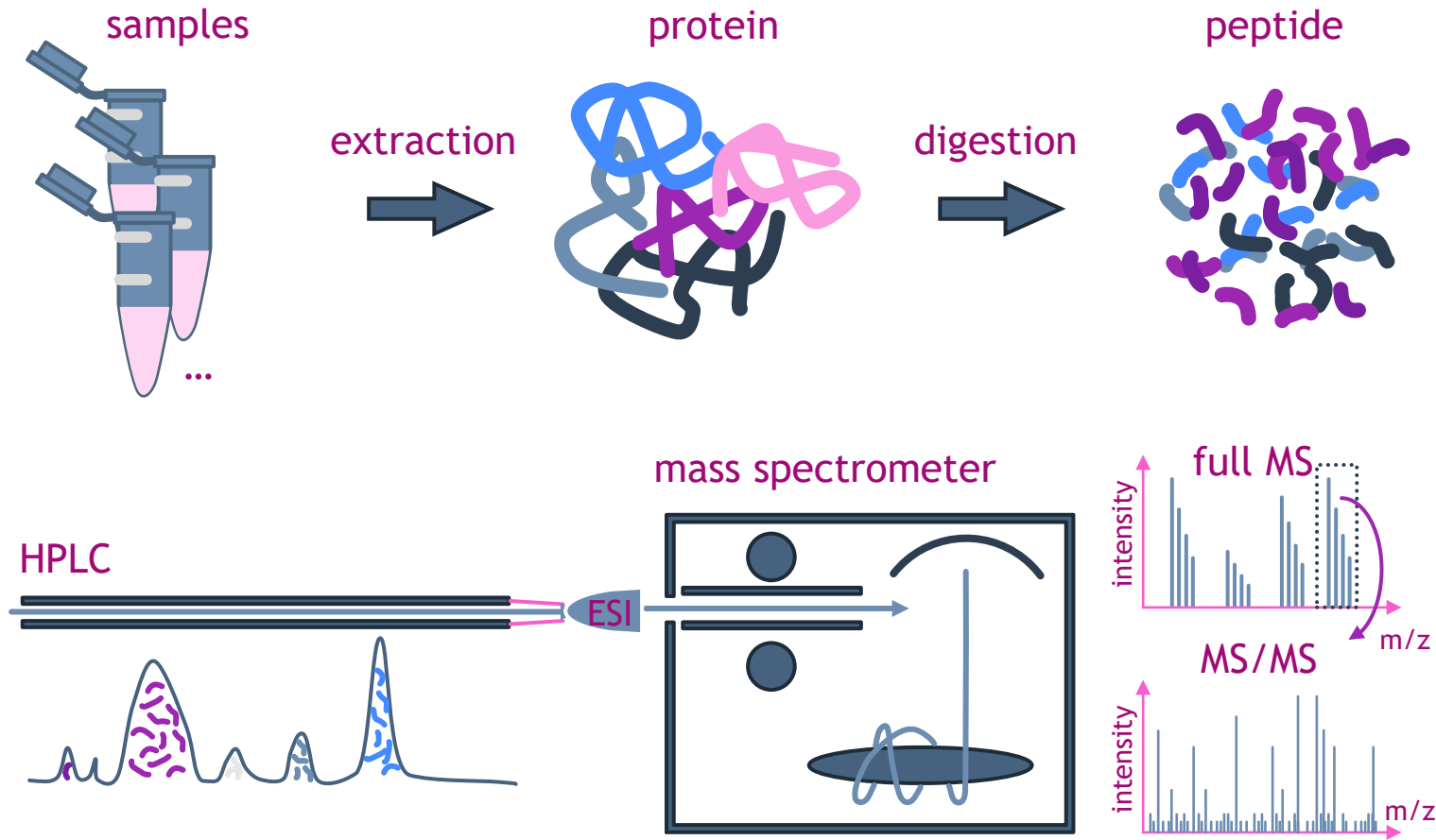
[Charge state determination](#)

[Peptide look up](#)

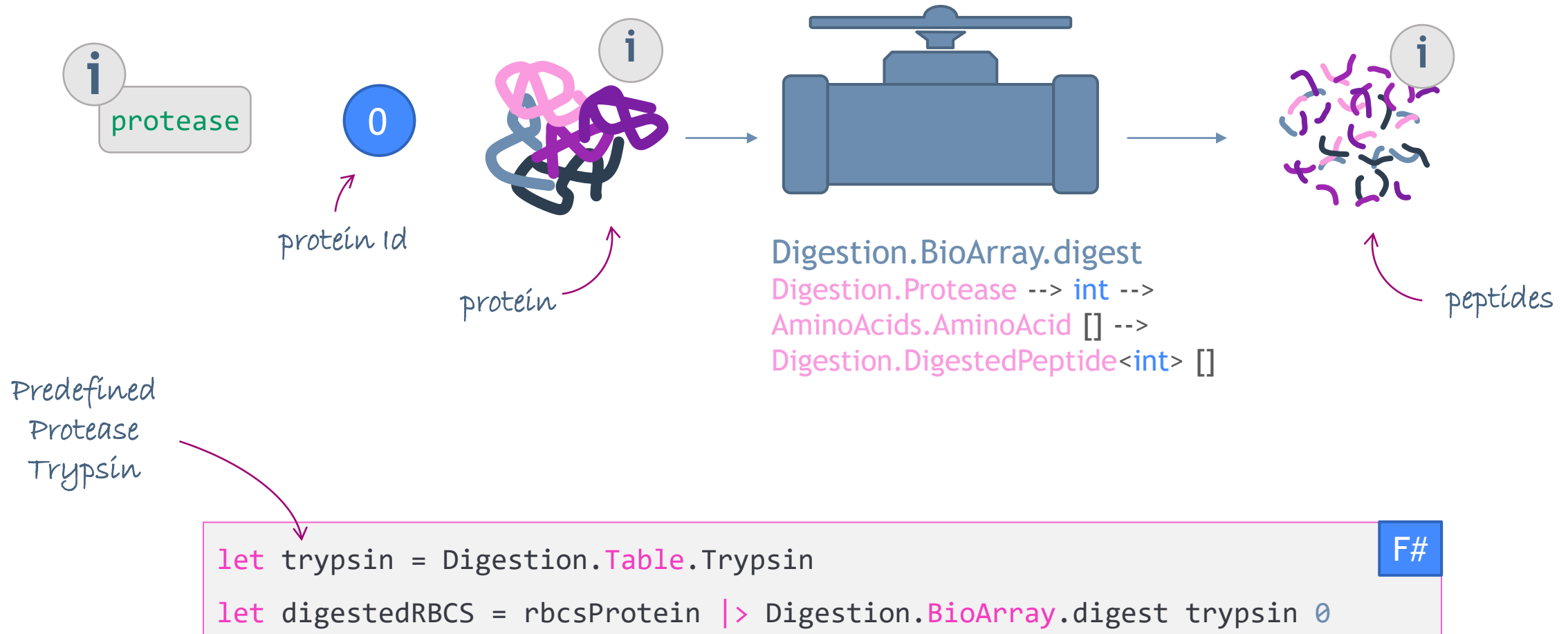
[Peptide identification](#)



Proteomics workflow



Step1: Proteolytic Digestion of Proteins



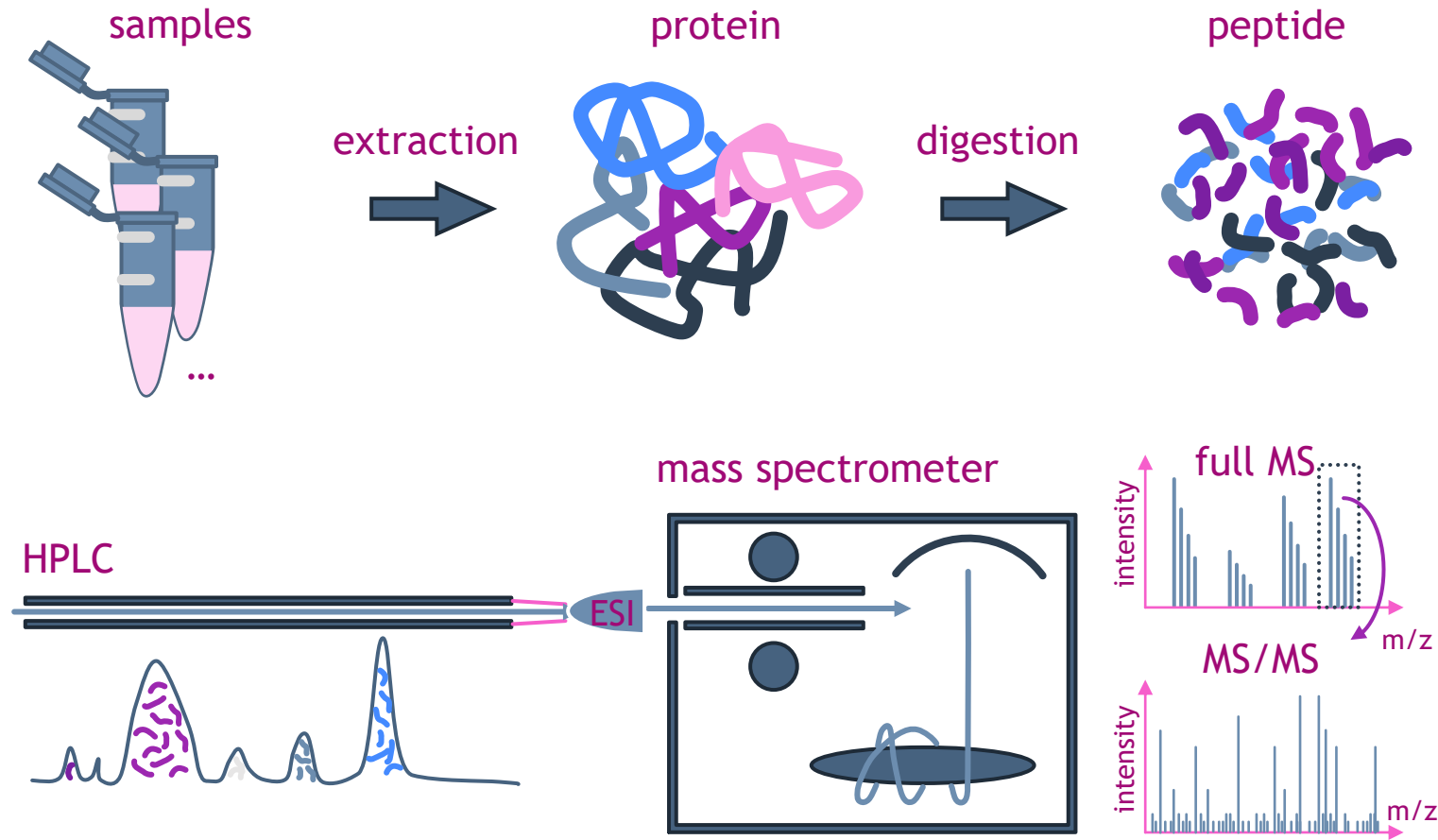
Proteolytic Digestion

```
let trypsin = Digestion.Table.Trypsin
let digestedRBCS = rbcProtein |> Digestion.BioArray.digest trypsin 0
```

F#

```
val digestedRBCS : Digestion.DigestedPeptide<int> [] =
[|
  { ProteinID = 0
    MissCleavages = 0
    CleavageStart = 0
    CleavageEnd = 9
    PepSequence = [Met; Met; Val; Trp; Thr; Pro; Val; Asn; Asn; Lys]
  };
  ...
|]
```

Proteomics Workflow

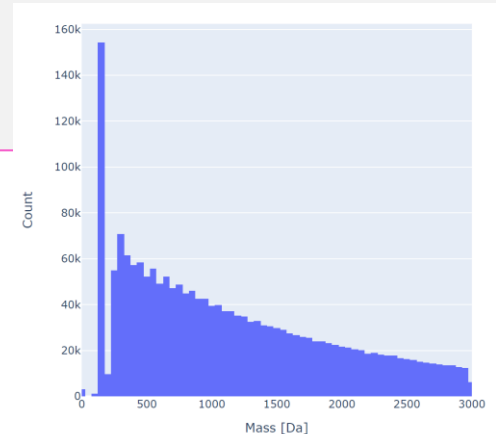


Calculating Peptide Masses

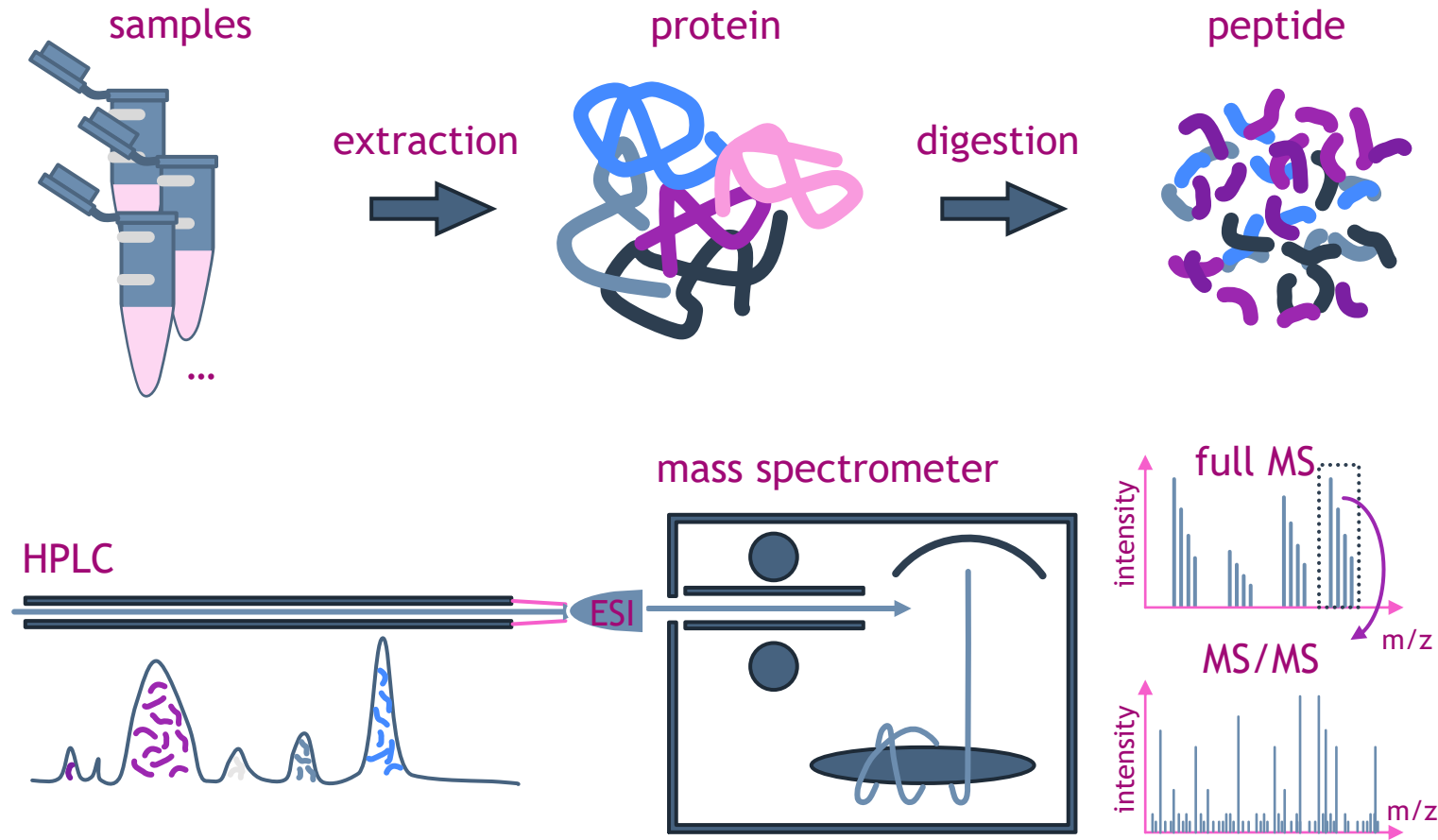
F#

digestedProteins

```
|> Array.map (fun peptide ->  
    // calculate mass for each peptide  
    BioSeq.toMonoisotopicMassWith  
        (BioItem.monoisoMass ModificationInfo.Table.H20) peptide  
    )  
  
|> Array.filter (fun x -> x < 3000.)  
// visualize distribution of all peptide masses < 3000 Da  
|> fun masses -> Chart.Histogram(data = masses, orientation = Vertical, NBinsX = 100)  
|> Chart.withXAxisStyle (TitleText = "Mass [Da]", MinMax = (0., 3000.))  
|> Chart.withYAxisStyle "Count"
```

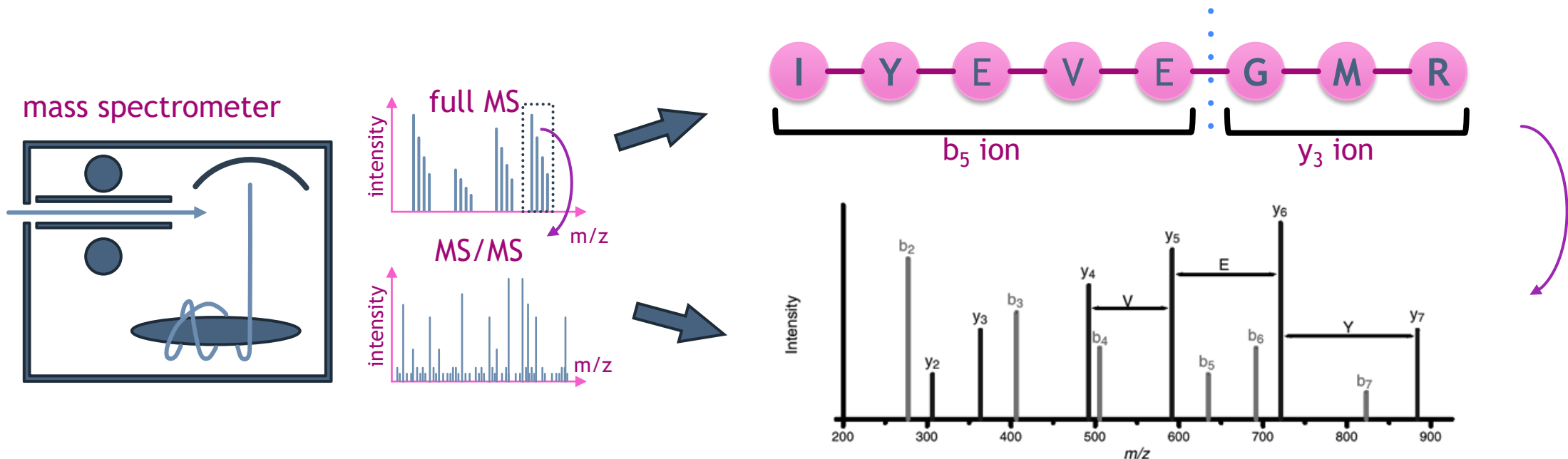


Proteomics Workflow



Pepide Identification by Tandem MS

- ▶ Sequence consists of the **same 20 building blocks** (amino acids)
- ▶ CID: peptide breaks preferentially along the **backbone**
- ▶ Peptide **fragment ions correspond to prefixes and suffixes** of the whole peptide sequences
- ▶ Complete ion series (ladders) reveal the sequence via mass differences of adjacent fragment ions



Simulation of MS2 Fragmentation

preselecting peptides of interest

peptideAndMasses

F#

```
|> Array.filter (fun (sequence,mass) -> mass > 1020.52 && mass < 1020.53)
```

```
let predictFromSequence peptide =
```

```
[
```

```
    peptide
```

```
    |> Mz.Fragmentation.Series.yOfBioList BioItem.initMonoisoMassWithMemP
```

```
    peptide
```

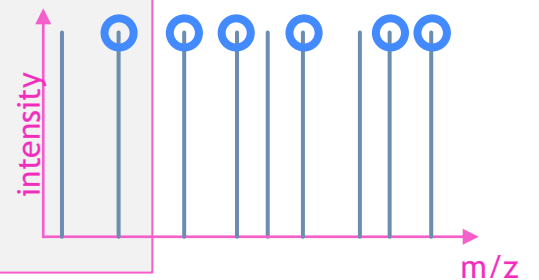
```
    |> Mz.Fragmentation.Series.bOfBioList BioItem.initMonoisoMassWithMemP
```

```
]
```

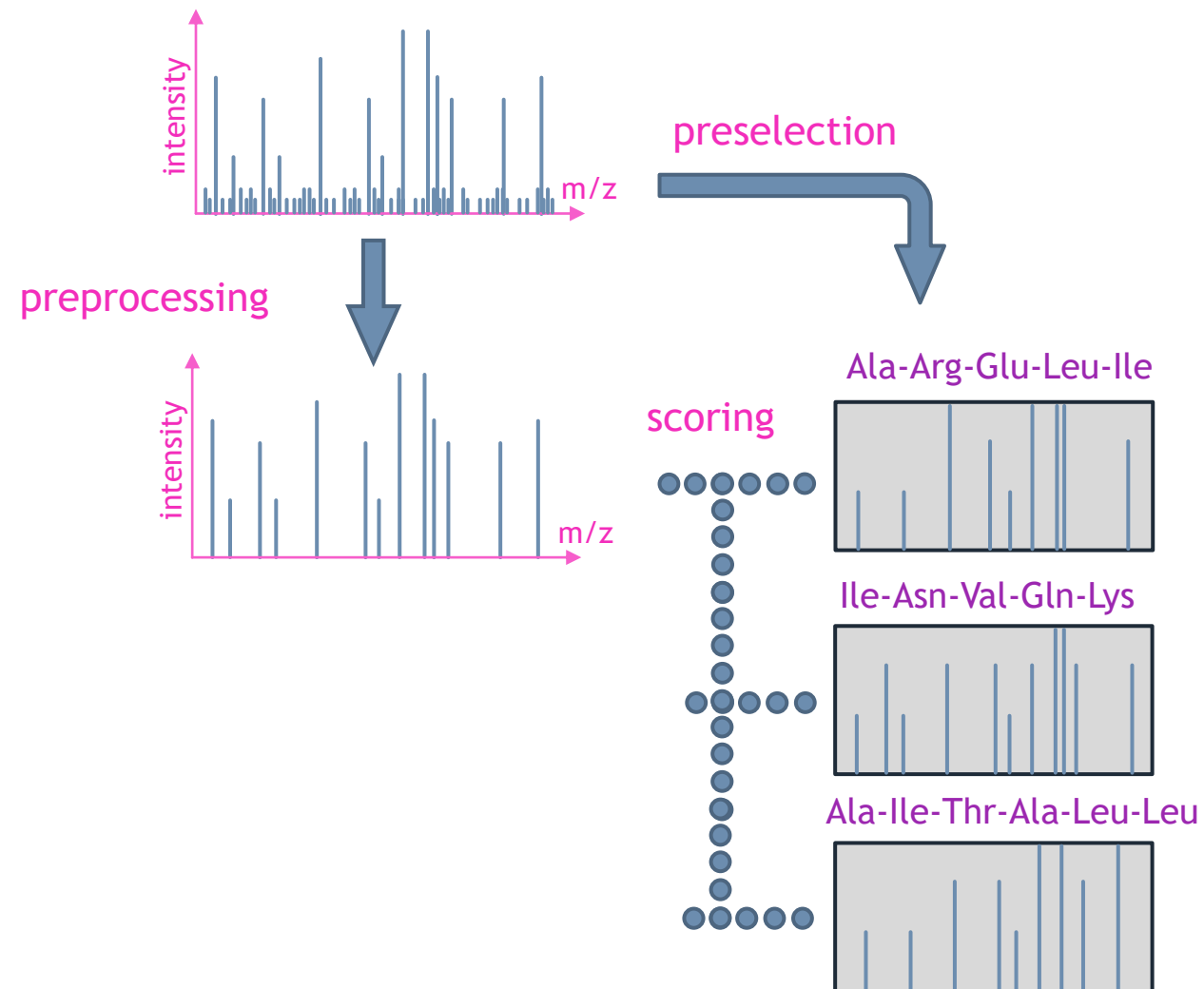
```
|> List.concat
```

```
|> Mz.SequestLike.predictOf (lowerScanLimit,upperScanLimit) 2.
```

hypothetical spectrum
(y/b ions)



Basic Steps of Peptide Identification



Spectrum Preprocessing

```
let ms2 = BioFSharp.IO.Mgf.readMgf „filename“
```

```
let lowerScanLimit = 150.
```

```
let upperScanLimit = 1000.
```

```
let preprocessedIntesities =
```

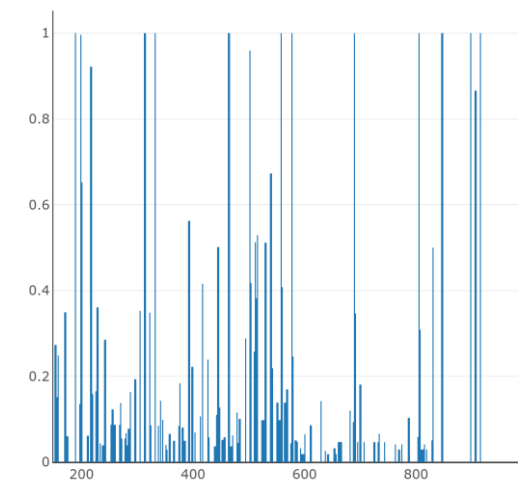
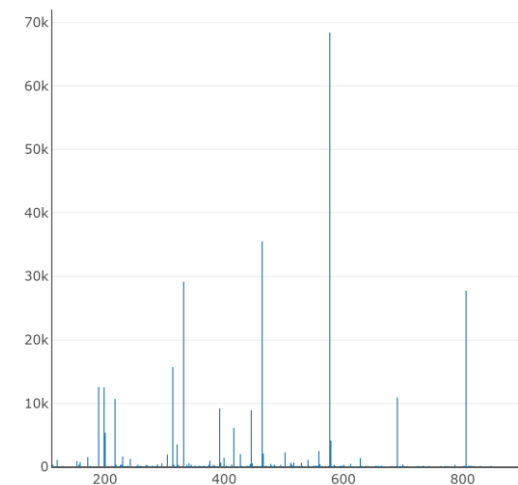
```
  Mz.PeakArray.zip ms2.Mass ms2.Intensity
```

```
  |> (fun pa -> Mz.PeakArray.peaksToNearestUnitDaltonBinVector  
                pa lowerScanLimit upperScanLimit)
```

```
  |> (fun pa -> Mz.SequestLike.windowNormalizeIntensities pa 10)
```

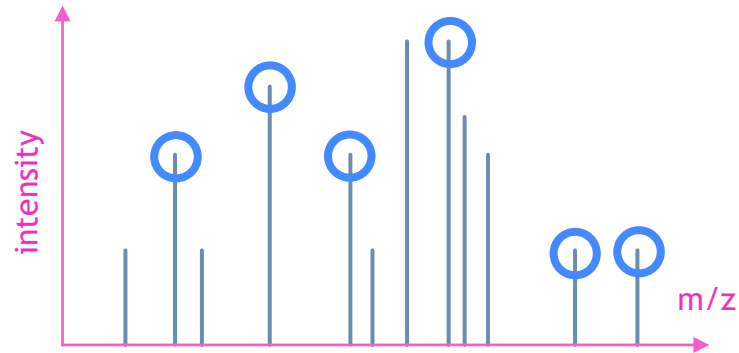
```
Chart.Column(preprocessedIntesities, [lowerScanLimit .. upperScanLimit])
```

```
|> Chart.withTemplate ChartTemplates.light
```

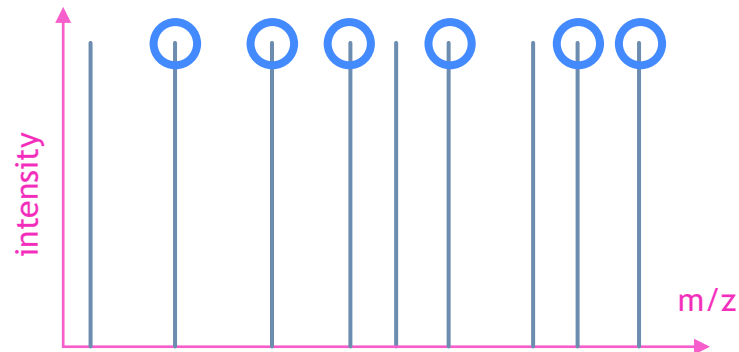


Scoring by Auto-Correlation (SEQUEST)

measured spectrum



Simulated spectrum



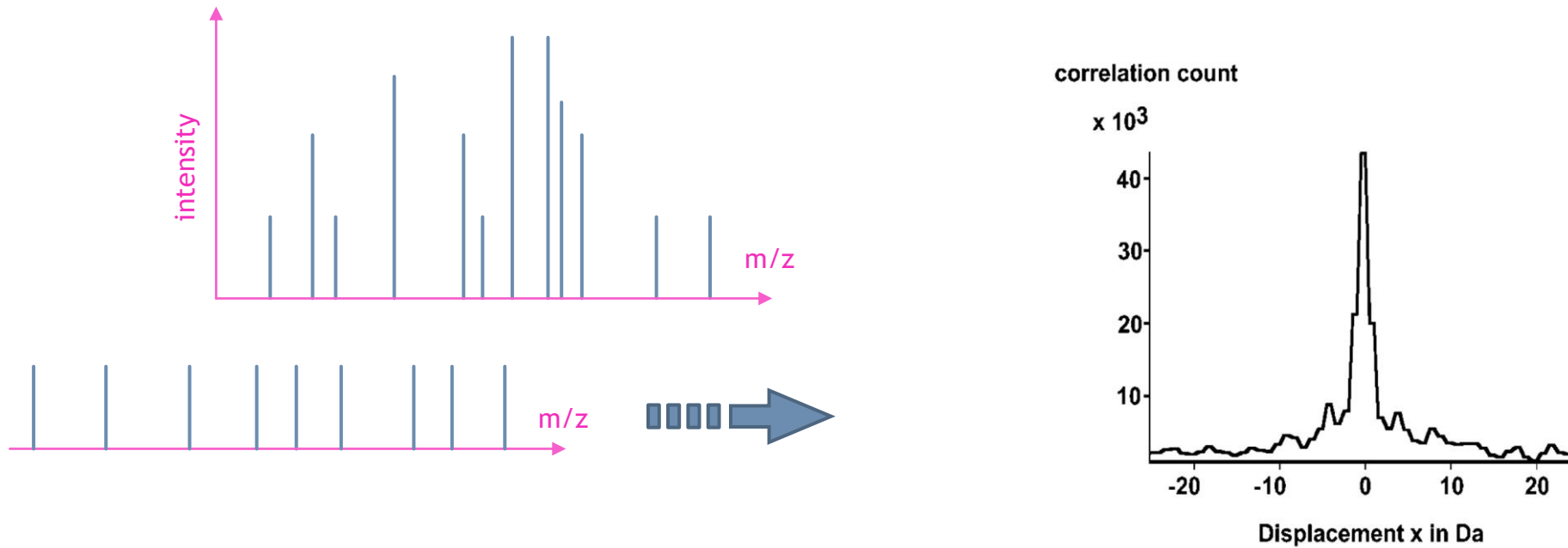
acquired spectrum



cross correlation
sum of all peaks in overlap

hypothetical spectrum
(y/b ions)

Scoring by Auto-Correlation (SEQUEST)



- The peaks that overlap upon spectra shifting are used to calculate the autocorrelation

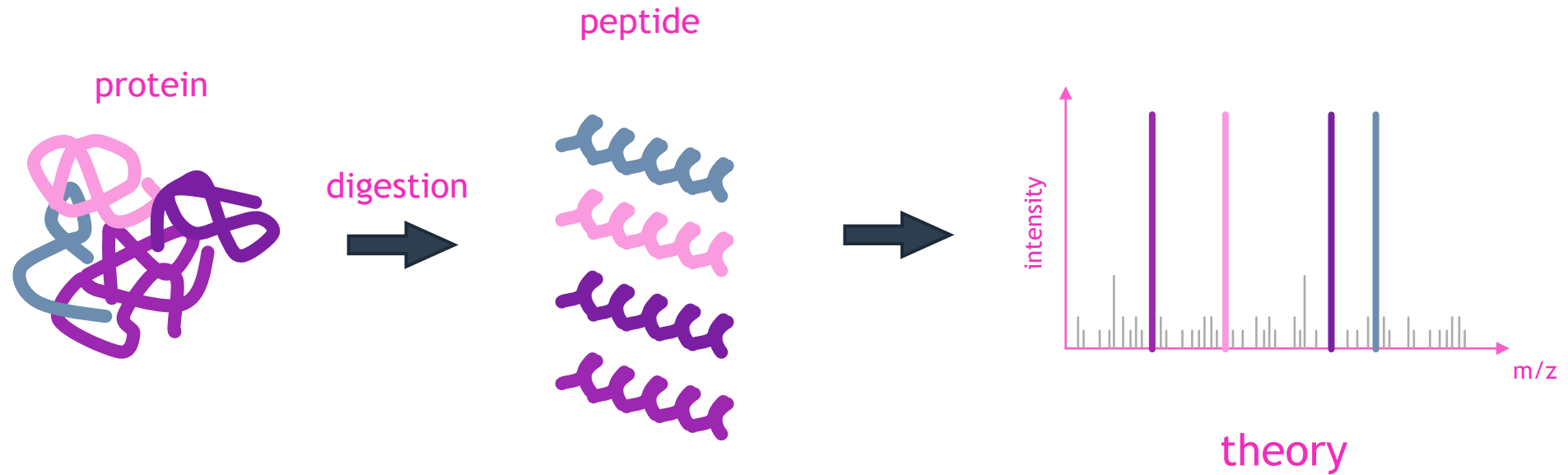
Matching and Scoring

```
let sortedScores =  
    peptideAndMasses  
    |> Array.filter (fun (sequence,mass) ->  
        mass > 1020.52 && mass < 1020.53  
    )  
    |> Array.map (fun (sequence,mass) ->  
        sequence,predictFromSequence sequence  
    )  
    |> Array.map (fun (sequence,theoSpectrum) ->  
        sequence, BioFSharp.Mz.SequestLike.scoreSingle  
            theoSpectrum preprocessedIntesities  
    )  
    |> Array.sortByDescending (fun (sequence,score) -> score)  
  
sortedScores
```

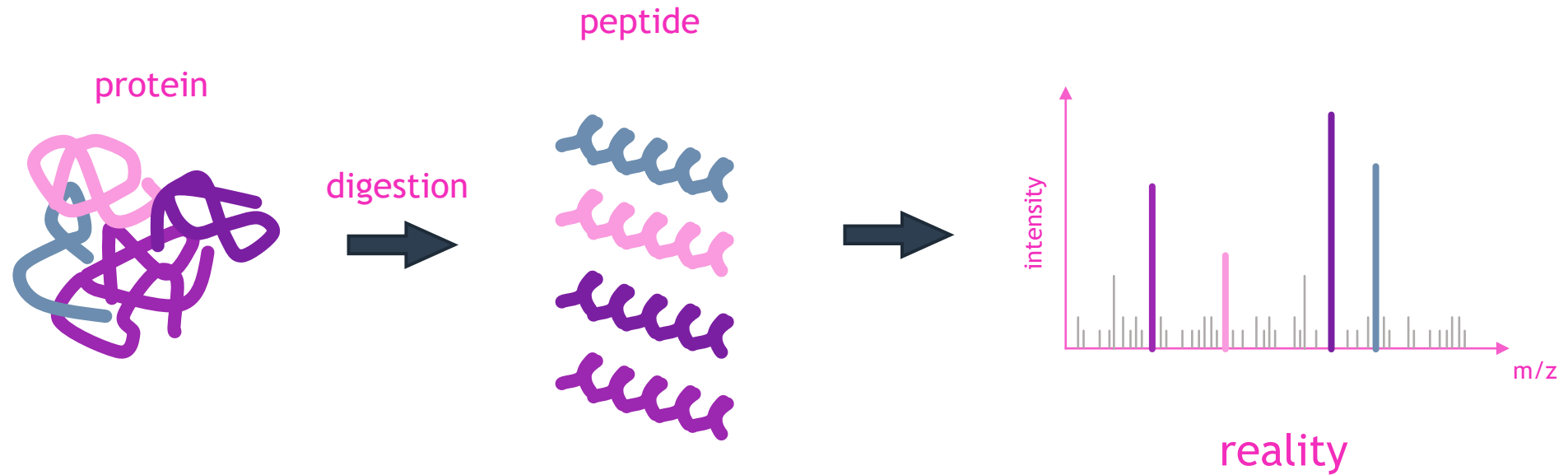
F#

index	value
0	▶ ([Asp; Thr; Asp; ...], 11.489317931131332)
1	▶ ([Val; Val; Asp; ...], 4.800498735800271)
2	▶ ([Asn; Tyr; Val; ...], 3.547344985238297)
3	▶ ([Ser; Leu; Ile; ...], 3.0626381054538503)
4	▶ ([Gly; Glu; Glu; ...], 2.6674782152072827)
5	▶ ([Asp; Phe; Val; ...], 2.5897868421493198)
6	▶ ([Leu; Ser; Ser; ...], 2.5532097910662954)
7	▶ ([Leu; Ser; Ser; ...], 2.5532097910662954)
8	▶ ([Ala; Val; Gln; ...], 2.5501482345520454)
9	▶ ([Glu; Tyr; Gln; ...], 2.222608555919107)
10	▶ ([Thr; Thr; Pro; ...], 1.9836285155579028)
11	▶ ([Leu; Glu; Gly; ...], 1.9032621441376565)
12	▶ ([Asp; Val; Leu; ...], 1.7938301080315675)
13	▶ ([Phe; Leu; Asp; ...], 1.7615553265303323)
14	▶ ([Val; Asp; Phe; ...], 1.7576420105210504)
15	▶ ([Ala; Glu; Ala; ...], 1.6090022149128864)
16	▶ ([Leu; Gly; Ala; ...], 1.5177524771864663)
17	▶ ([Gln; Ile; Ile; ...], 1.5004855067669207)
18	▶ ([Trp; Pro; Gly; ...], 1.4586761816897038)
19	▶ ([Val; Ser; Thr; ...], 1.1391486393077925)

Detectability Problem for Peak Intensities



Detectability Problem for Peak Intensities

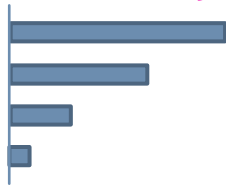


d::pPop algorithm

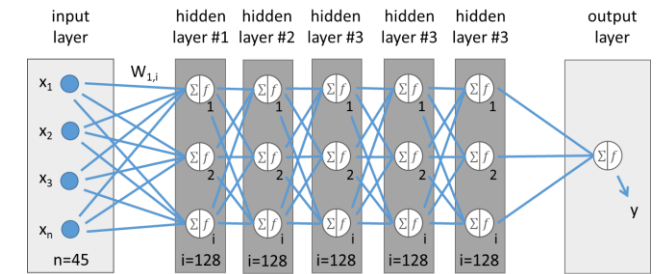
protein A

LGREWELSFRK
SGTFNFGDK
VALTALTAK
NKQDVLFFIDNIK

intensity



0.4	0.6	0.2	0.3
0.7	0.3	0.9	0.1
0.1	0.2	0.1	0.1
0.3	0.3	0.6	0.8



* ranking

* feature vectors

* machine learning

d::pPop website


d::pPop

deep peptide observability predictor

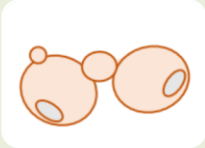
d::pPop uses a deep neural network to predict proteotypic peptides for proteins of interest.

1. Model selection

Select the model that is the closest to the organism for which you intend to predict proteotypic peptides.



Plant model



Non-Plant model

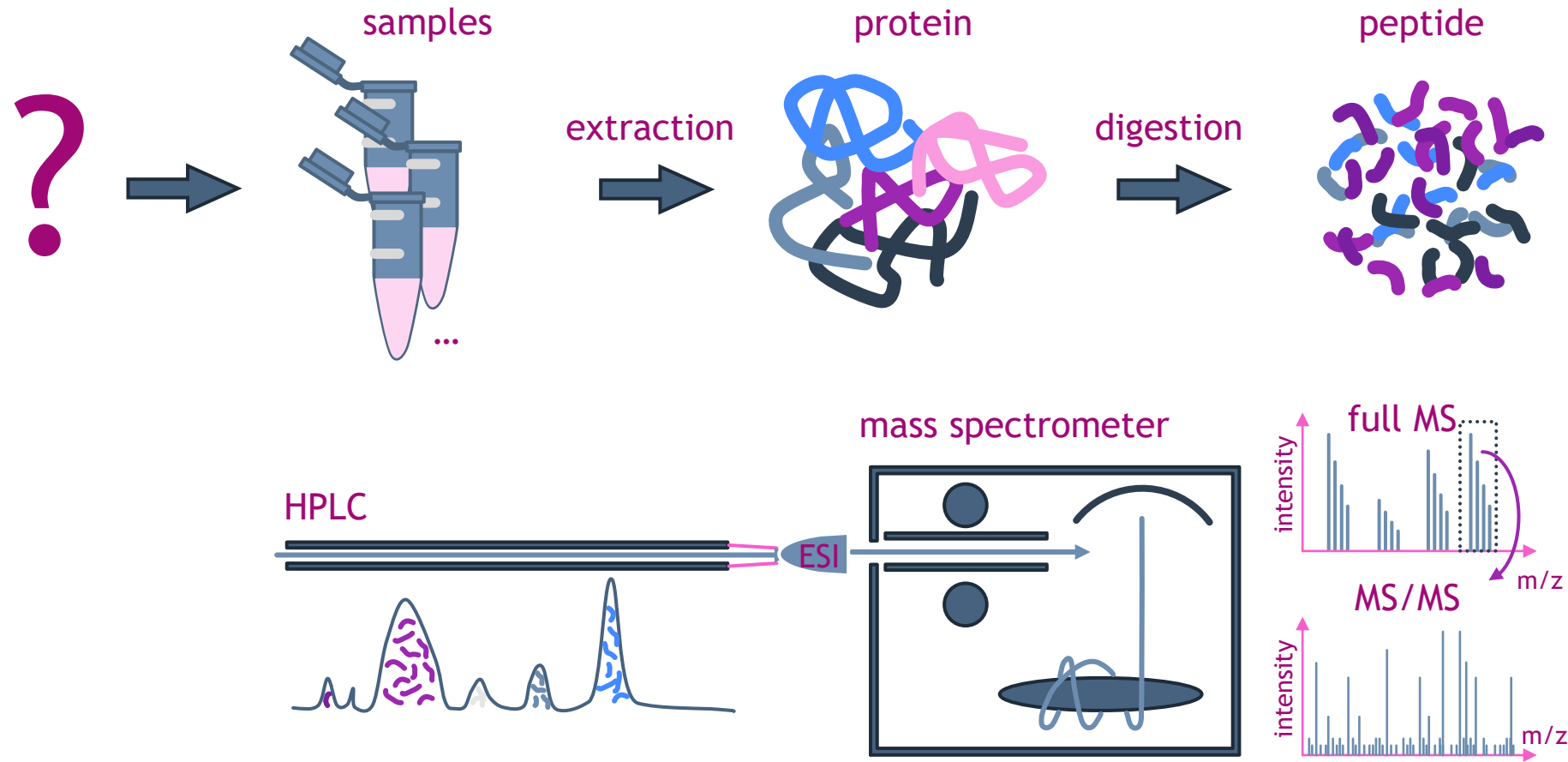
Model description

Generalized plant model. Provide the genome of your organism of interest in FASTA format and choose the proteins of interest by selecting their corresponding headers.

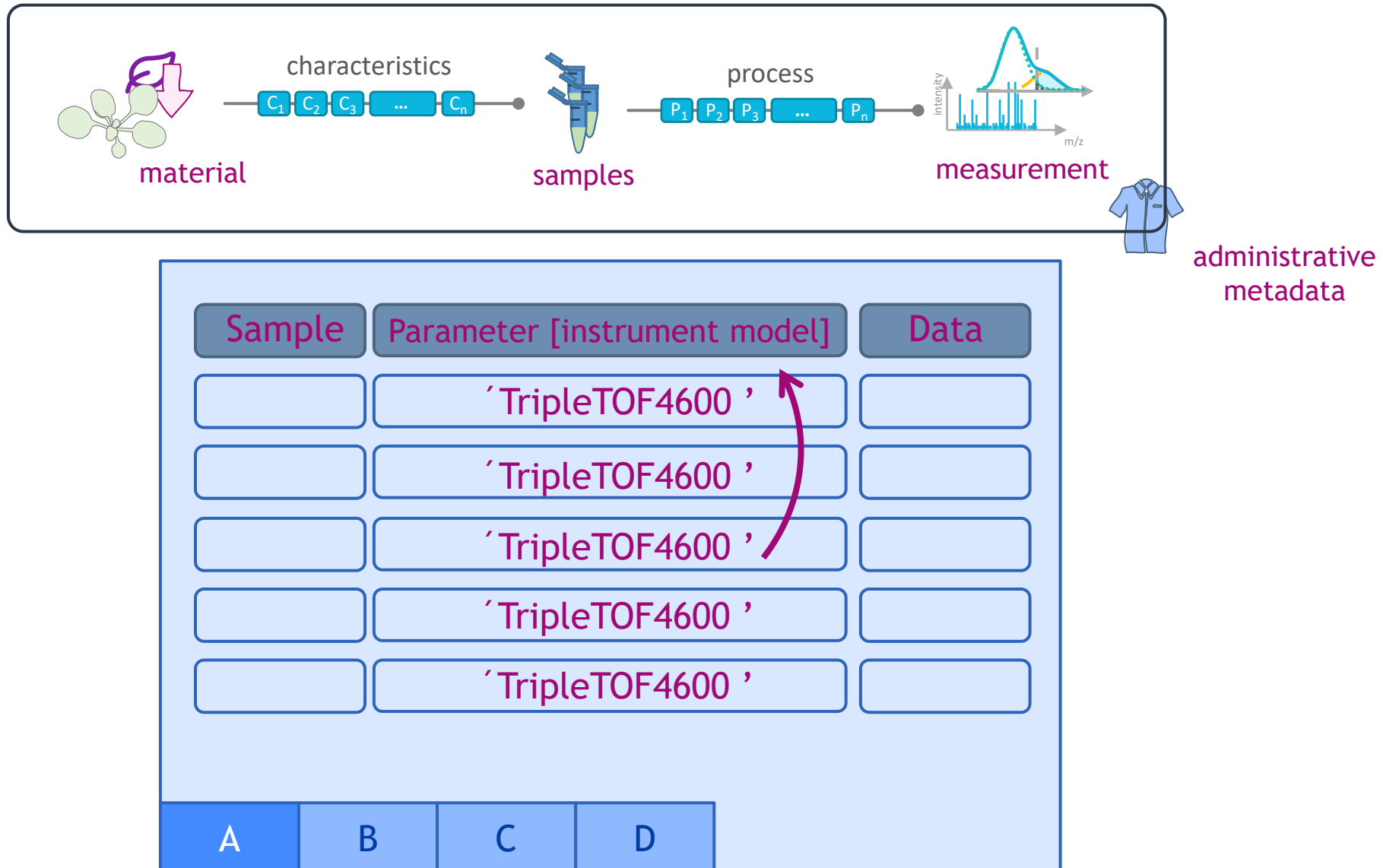
proceed to protein selection



Proteomics Workflow



Annotation of the Sample Generating Process



Swate - Metadata annotation tool

Sink Parameter ['parent term'] Source

new parameter

Excel + Swate Plug-In

The screenshot shows an Excel spreadsheet with the Swate Plug-In interface. The spreadsheet has columns labeled 'Source Name', 'Parameter (instrument model)', and 'Sample Name'. A new parameter is being added to the 'Parameter (instrument model)' column. The Swate interface on the right shows the 'Annotation building block selection' section, where a parameter is being added to the annotation table. The 'Parameter' dropdown is set to 'instrument model'. Below this, there is a section for 'Add/Update unit reference to existing building block' with an 'Add unit' dropdown and an 'Update unit for cells' button. The bottom of the Swate interface shows the 'More about Parameter' section, which explains that parameters are used to annotate experimental workflows and group parameters to create a protocol. The Swate Release Version is 0.4.7.



Swate without Excel

The screenshot shows the Swate web application interface. The browser address bar displays <https://swate-alpha.nfdi4plants.org>. The application has a green header bar with the 'SWATE' logo and a hamburger menu icon. Below the header, there is a table with three columns: 'Source Name' (blue header), 'Parameter [organism] >>' (dark blue header), and 'Sample Name' (red header). The table contains two rows: 'S1' with 'Arabidopsis thaliana' and an empty 'Sample Name' cell, and 'S2' with empty cells. Below the table, there is a small interface with a box containing '1' and a '+' button. On the right side, there is a sidebar titled 'Building Blocks' with a sub-header 'Add annotation building blocks (columns) to the annotation table.' Below this, there are two search bars: one for 'Parameter' and one for 'Term' and 'Unit'. Each search bar has a dropdown arrow and a 'Start typing to search' placeholder. At the bottom of the sidebar, there is a red button labeled 'Add building block'. At the bottom of the application, there is a footer area with a tab labeled 'EvilSheep25' and a '+' button, and a release version string 'Swate Release Version 0.8.0-alpha07'.

Source Name	Parameter [organism] >>	Sample Name
S1	Arabidopsis thaliana	
S2		

1 +

SWATE

Building Blocks

Add annotation building blocks (columns) to the annotation table.

Parameter ▼ Start typing to search

Term Unit Start typing to search

[Use advanced search header](#) [Use advanced search body](#)

Add building block

Swate Release Version 0.8.0-alpha07

EvilSheep25 +

AutoSave toolTalk.xlsx

FileHomeInsertDrawPage LayoutFormulasDataReviewViewDeveloperHelpTable Design

Paste

Format Painter

Calibri

11

Alignment

Custom

Number

Conditional Formatting

Format as Table

Cell Styles

Styles

Insert

Delete

Format

Cells

AutoSum

Fill

Clear

Editing

Sort & Filter

Find & Select

Analysis

Share

Comments

	A	B	E	I	J	K
	Source Name	Characteristics [sample label]	Factor [temperature unit]	Data File Name		
2	Heat_15A_OD_R1	15N	32.00 degree Celsius	Heat_15A_OD_R1.wiff		
3	Heat_15A_OD_R2	15N	32.00 degree Celsius	Heat_15A_OD_R2.wiff		
4	Heat_180A_OD_R1	15N	32.00 degree Celsius	Heat_180A_OD_R1.wiff		
5	Heat_180A_OD_R2	15N	32.00 degree Celsius	Heat_180A_OD_R2.wiff		
6	Heat_2880A_OD_R1	15N	32.00 degree Celsius	Heat_2880A_OD_R1.wiff		
7	Heat_2880A_OD_R2	15N	32.00 degree Celsius	Heat_2880A_OD_R2.wiff		
8	Heat_5760A_OD_R1	15N	32.00 degree Celsius	Heat_5760A_OD_R1.wiff		
9	Heat_5760A_OD_R2	15N	32.00 degree Celsius	Heat_5760A_OD_R2.wiff		
10	Heat_5760A_15D_R1	15N	32.00 degree Celsius	Heat_5760A_15D_R1.wiff		
11	Heat_5760A_15D_R2	15N	32.00 degree Celsius	Heat_5760A_15D_R2.wiff		
12	Heat_5760A_180D_R1	15N	32.00 degree Celsius	Heat_5760A_180D_R1.wiff		
13	Heat_5760A_180D_R2	15N	32.00 degree Celsius	Heat_5760A_180D_R2.wiff		
14	Heat_5760A_2880D_R1	15N	32.00 degree Celsius	Heat_5760A_2880D_R1.wiff		
15	Heat_5760A_2880D_R2	15N	32.00 degree Celsius	Heat_5760A_2880D_R2.wiff		
16	Heat_5760A_5760D_R1	15N	32.00 degree Celsius	Heat_5760A_5760D_R1.wiff		
17	Heat_5760A_5760D_R2	15N	32.00 degree Celsius	Heat_5760A_5760D_R2.wiff		
18	Cold_15A_OD_R1	15N	4.00 degree Celsius	Cold_15A_OD_R1.wiff		
19	Cold_15A_OD_R2	15N	4.00 degree Celsius	Cold_15A_OD_R2.wiff		
20	Cold_180A_OD_R1	15N		Cold_180A_OD_R1.wiff		
21	Cold_180A_OD_R2	15N		Cold_180A_OD_R2.wiff		
22	Cold_2880A_OD_R1	15N		Cold_2880A_OD_R1.wiff		
23	Cold_2880A_OD_R2	15N	4.00 degree Celsius	Cold_2880A_OD_R2.wiff		
24	Cold_5760A_OD_R1	15N	4.00 degree Celsius	Cold_5760A_OD_R1.wiff		
25	Cold_5760A_OD_R2	15N	4.00 degree Celsius	Cold_5760A_OD_R2.wiff		
26	Cold_5760A_15D_R1	15N	4.00 degree Celsius	Cold_5760A_15D_R1.wiff		
27	Cold_5760A_15D_R2	15N	4.00 degree Celsius	Cold_5760A_15D_R2.wiff		
28	Cold_5760A_180D_R1	15N	4.00 degree Celsius	Cold_5760A_180D_R1.wiff		
29	Cold_5760A_180D_R2	15N	4.00 degree Celsius	Cold_5760A_180D_R2.wiff		
30	Cold_5760A_2880D_R1	15N	4.00 degree Celsius	Cold_5760A_2880D_R1.wiff		
31	Cold_5760A_2880D_R2	15N	4.00 degree Celsius	Cold_5760A_2880D_R2.wiff		
32	Cold_5760A_5760D_R1	15N	4.00 degree Celsius	Cold_5760A_5760D_R1.wiff		
33	Cold_5760A_5760D_R2	15N	4.00 degree Celsius	Cold_5760A_5760D_R2.wiff		
34	Highlight_15A_OD_R1	15N	22.00 degree Celsius	Highlight_15A_OD_R1.wiff		
35	Highlight_15A_OD_R2	15N	22.00 degree Celsius	Highlight_15A_OD_R2.wiff		
36	Highlight_180A_OD_R1	15N	22.00 degree Celsius	Highlight_180A_OD_R1.wiff		
37	Highlight_180A_OD_R2	15N	22.00 degree Celsius	Highlight_180A_OD_R2.wiff		

new parameter

factor

characteristic

datafile / sample

Swate

SWATE

notation building block selection

Add annotation building blocks (columns) to the annotation table.

Parameter

instrument

instrument

MS:1000463

instrument model

MS:1000031

instrument vendor

MS:1001269

medical instrument

ENVO:00010621

Mascot:Instrument

MS:1001656

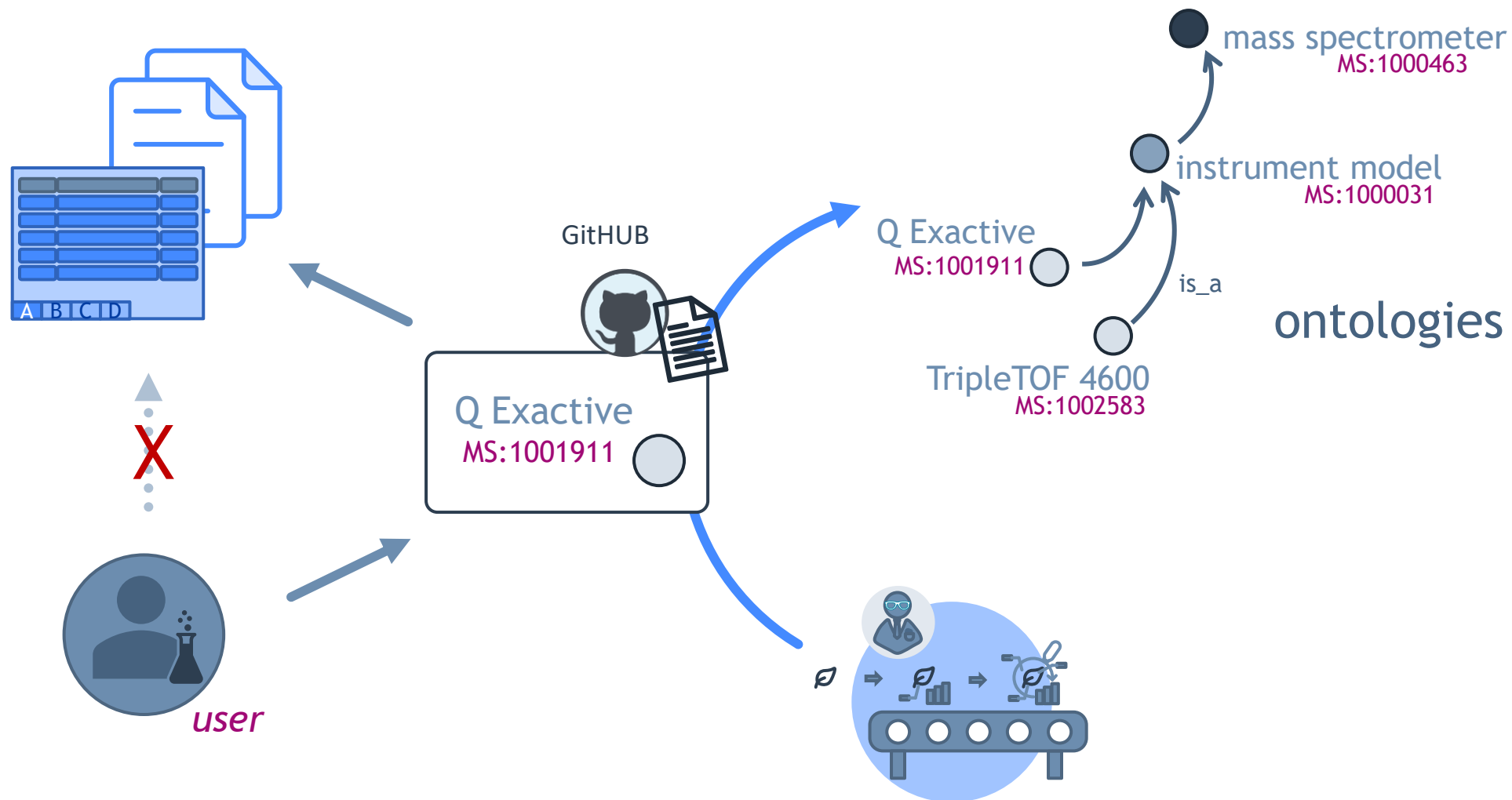
Add/Update unit

Use advanced search

More about Parameter:

Swate Release Version 0.4.7

Crowdsourcing controlled vocabulary development



Computational Systems Biology



Thank you for your
attention