

A peak fit algorithm to analyze mass spectra



*M. C. M. Costa, V. H. Martins, I. Prazeres, A. C. F. Santos, A. Medina
Instituto de Física, Universidade Federal do Rio de Janeiro, 21941-972 Rio de Janeiro, RJ

Introduction

The foremost purpose of this work is to introduce a Python language written code to interpret an array of ion counts and mass to charge ratios for experimental data analysis provided by a time-of-flight mass spectrometer (TOF). In order to test our new algorithm, we compared the branching ratios calculated with the new code and the results previously found by our group for the fragmentation of the CH₂FCF₃ (also known as SUVA 134a) by synchrotron radiation in the EUV and X-ray energy range [1]. Lastly, we also present the new prospect for the program, to extend our code to examine the 3D spectra produced by the coincidence measurements between one electron and two ions (PEPIICO).

Program Steps

- The first step is to remove the baseline. Although the function for removing the baseline for SUVA's spectra is linear, any type of function could be defined to be fitted to the experimental data. The function chosen is, then, fitted to data's intervals that don't have peaks and the baseline is subtracted to the original y-data (counts).

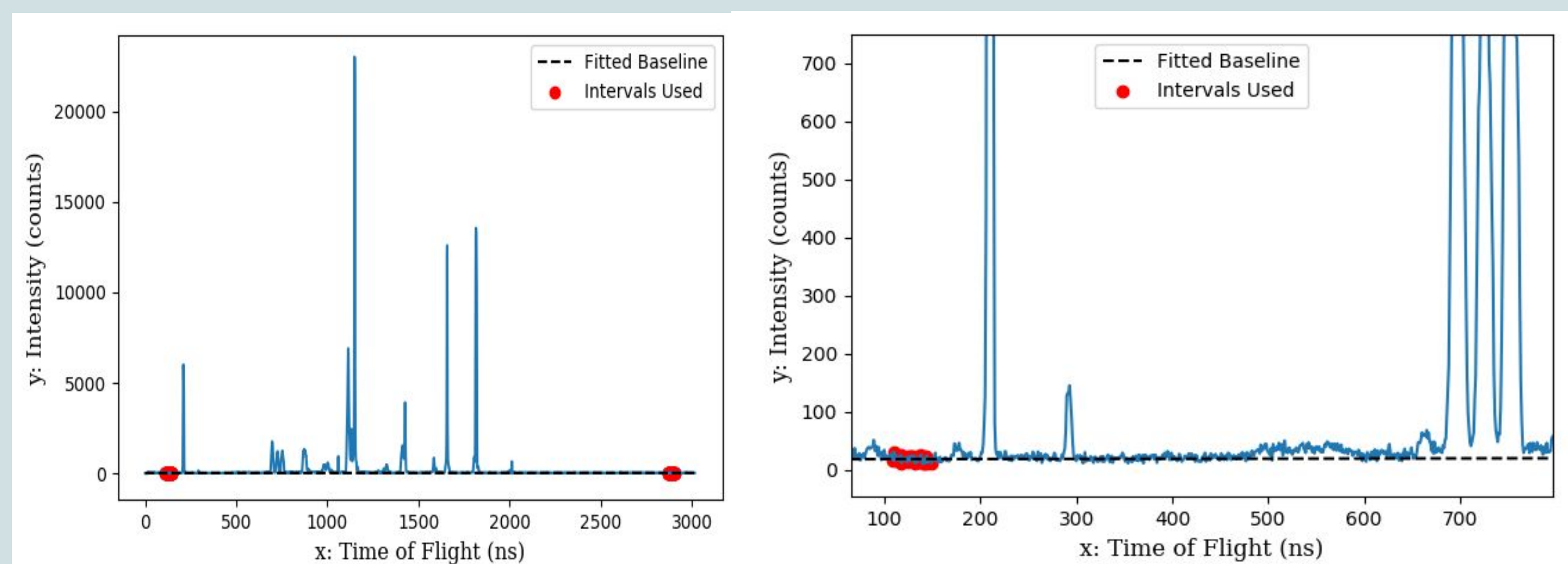


Fig. 1: Spectra and the process of removing the baseline.

- The code proceeds to find peaks in the spectra and enables the user to remove any peaks wrongly detected or generated from some unwanted contamination.

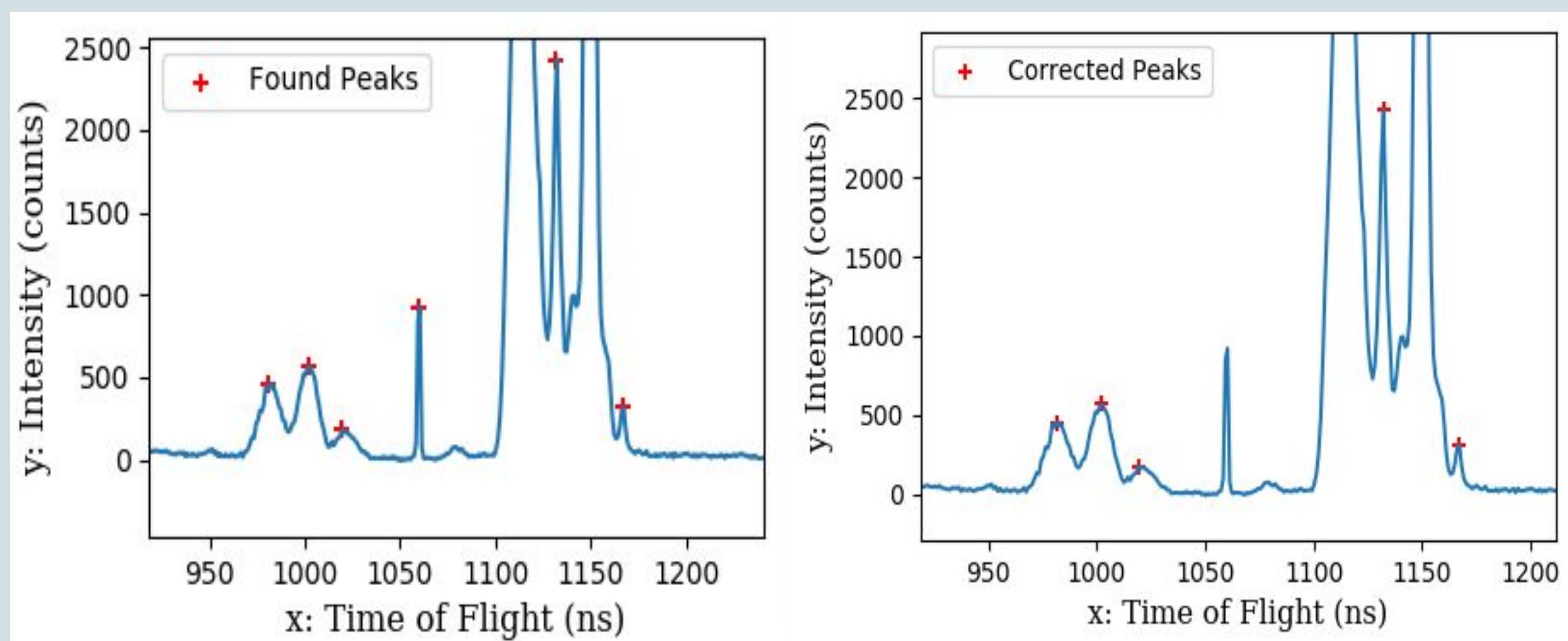


Fig. 2: Nitrogen contamination (N_2^+) peak removed.

- From the found peaks the program can automatically guess the initial parameters and define the number of gaussian contributions that should be added. A non-linear least squares method, subsequently, is used to fit the function defined to the data.

```
131 def gaussiana2(w, *parametros):
132     y = np.zeros_like(w)
133     for g in range(0, (3 + len(picos)), 3):
134         amp = parametros[c]
135         cen = parametros[c + 1]
136         sigma = parametros[c + 2]
137         y = y + amp * (1 / (sigma * (np.sqrt(2 * np.pi)))) * (np.exp((-1.0 / 2.0) * ((xp - cen) / sigma) ** 2)))
138     return y
139
140 popt2, pcov2 = scipy.optimize.curve_fit(gaussiana2, xp, yp, p0=chute)
141 perr = np.sqrt(np.diag(pcov2))
142
```

Fig. 3: Program lines for the definition of the gaussian function and fit method used.

- The partial fit with contributions from all peaks is shown and can be analyzed before the correction of any errors. Following a one Gaussian function can be refitted to an isolated peak once the limits of its width is known.

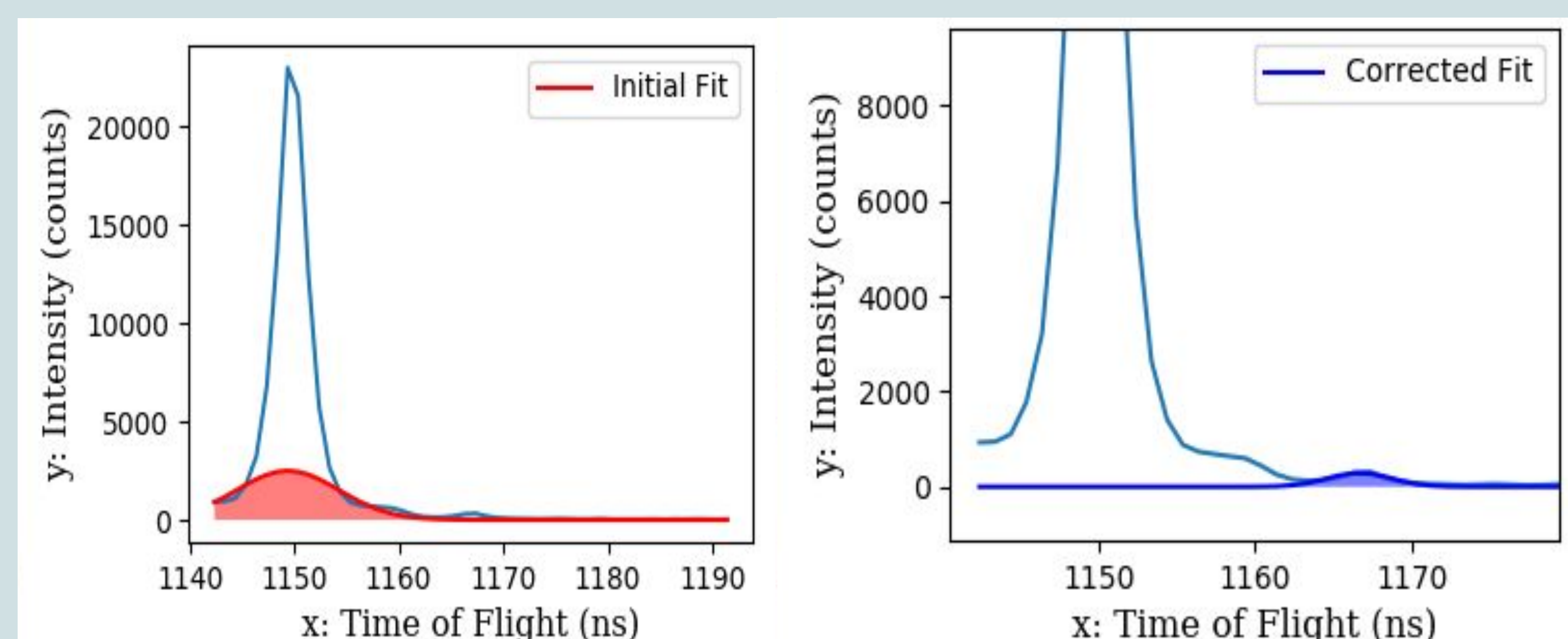


Fig. 4: One peak refit correction.

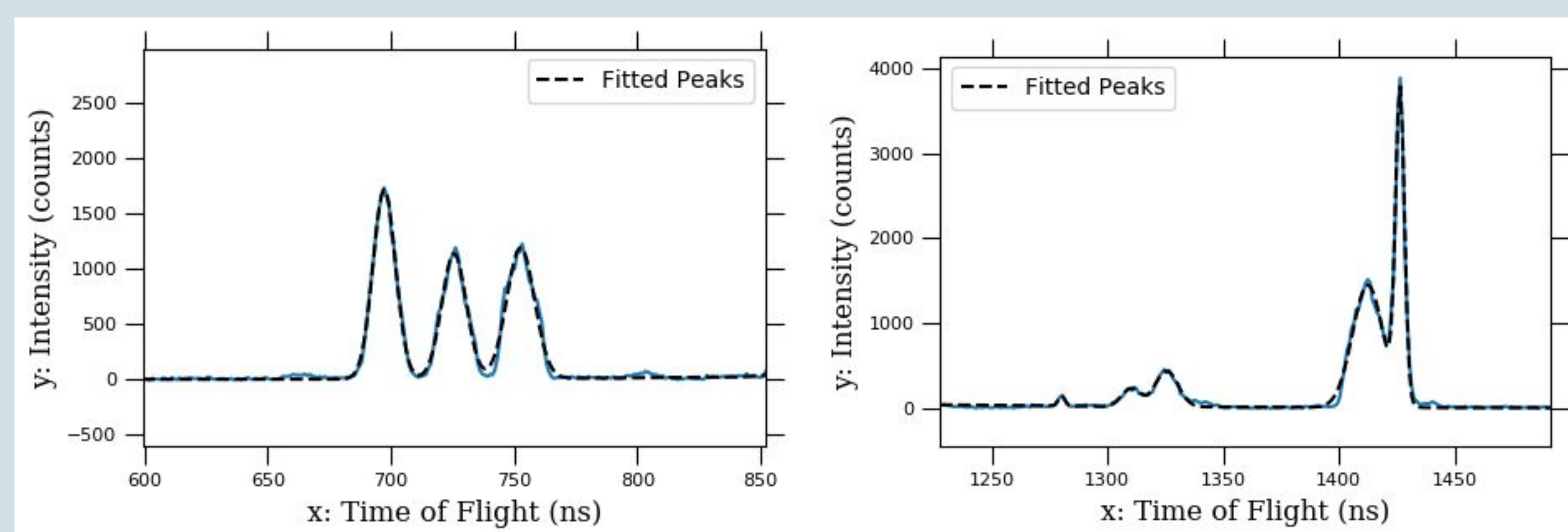


Fig. 5: Final fit.

PEPICO Results

We put forward a comparison between our code's result and the reference work as well as some branching ratios of the CH₂FCF₃ fragments.

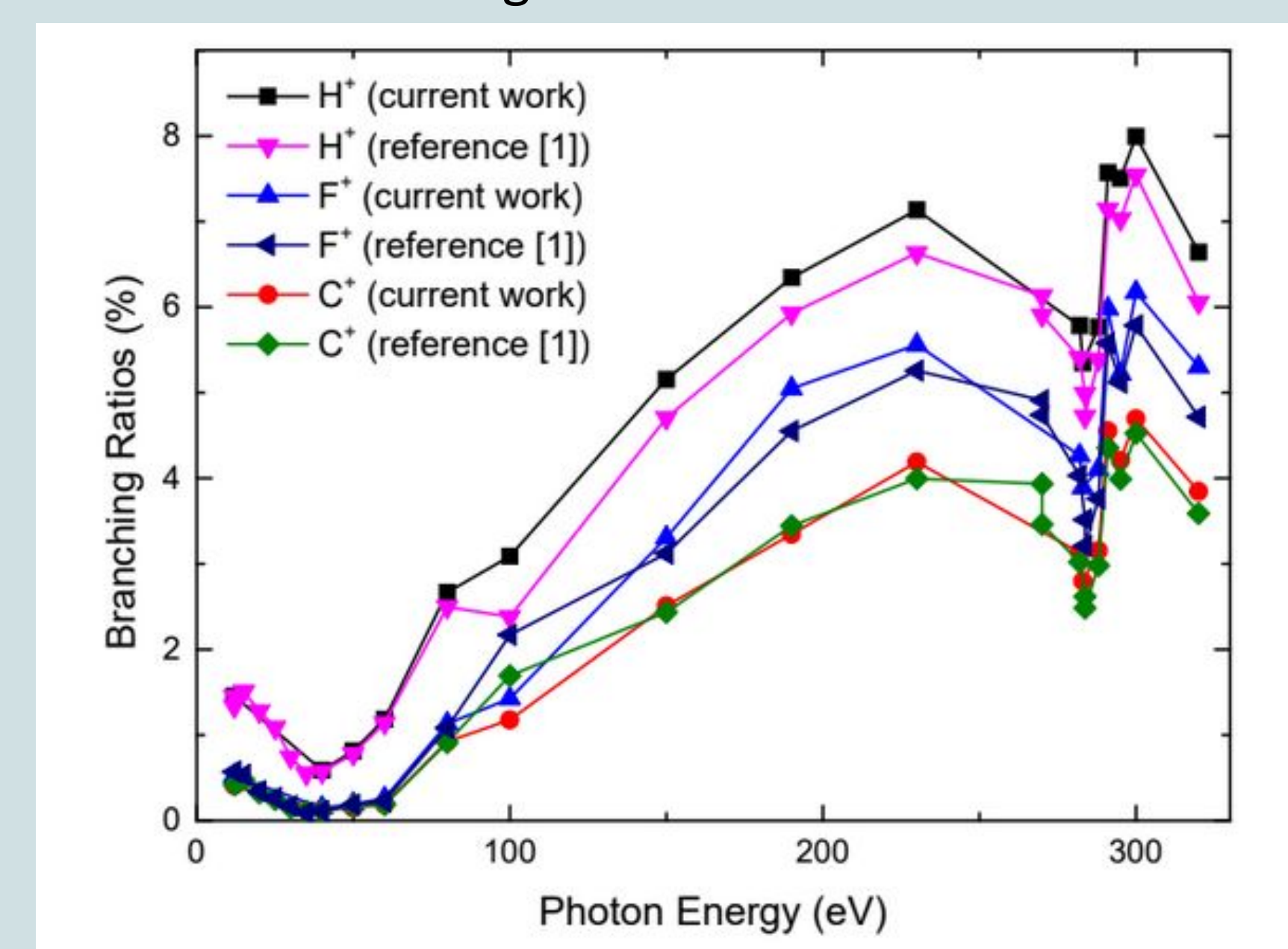


Fig. 6: Lightest fragments comparison.

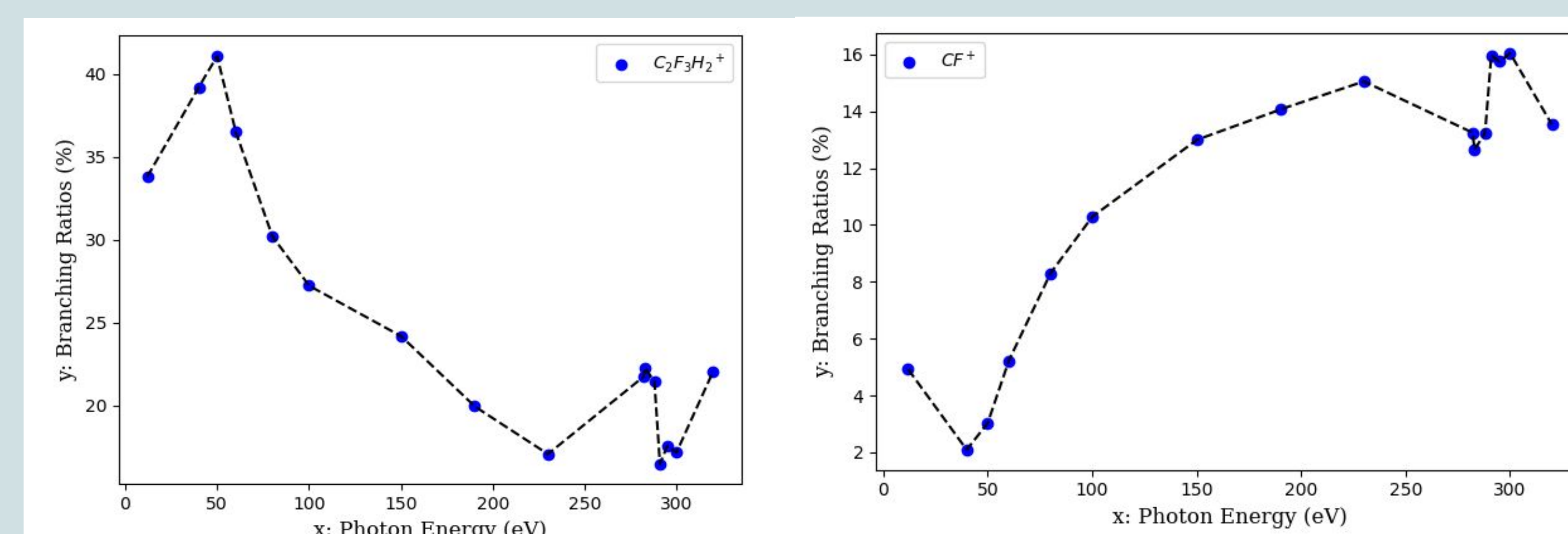


Fig. 7: Branching ratios of the CH₂FCF₃ fragments

Conclusion and Prospects

It is clearly noticeable the same pattern for SUVA's lightest fragments upon valence and C 1s edge (~290 eV) photon energies. Despite the fact that a time-of-flight spectra can already be analyzed in various programs, the further action is to implement a one electron - two ions coincidence investigation within the code.

This double coincidence results in a 3D spectra, as shown in Fig. 8, the coincidence islands can take different slopes and widths, we set as our main goal to produce a program capable of examining these parameters and discovering any desirable molecule's dissociation path.

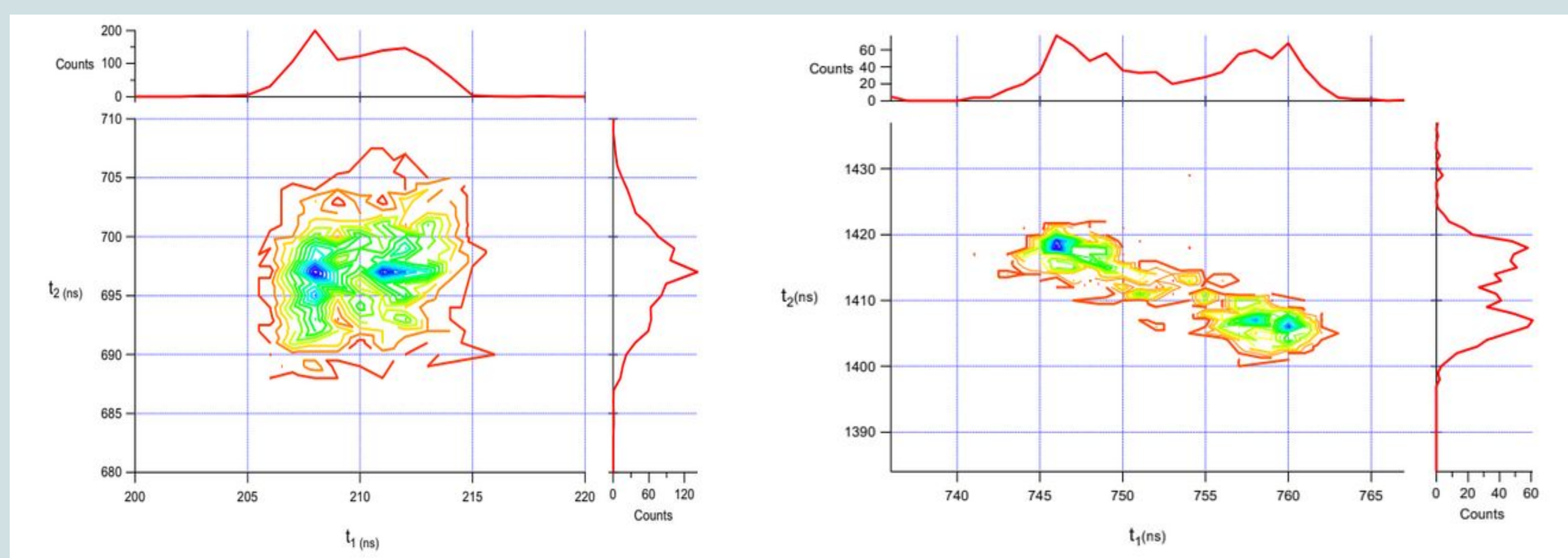


Fig. 8 : PEPICO spectra examples.

References:

- [1] V. Morcelle, A. Medina, L. C. Ribeiro, I. Prazeres, R. R. T. Marinho, M. S. Arruda, L. A. V. Mendes, M. J. Santos, B. N. C. Tenório, A. B. Rocha, A. C. F. Santos; Fragmentation of Valence and Carbon Core Excited and Ionized CH₂FCF₃ Molecule. J. Phys. Chem. A 122, 9755 (2018)