On accelerating MSBUDDY

Abstract

Annotating mass spectra (MS) with chemical structures is an unsolved problem. A simpler problem is that of annotation of MS with chemical formulae (i.e. C5H10O). For small molecules approaches such MSBUDDY have seen practical success. In this report we briefly introduce what MSBUDDY does, what are the main performance bottlenecks of the approach, and current, work-in-progress solutions to these problems. The primary focus of the report is to (1) detail the algorithmic flow of MSBUDDY, (2) identify performance bottlenecks and (3) propose GPU-based solutions for said bottlenecks. The report assumes general knowledge of the problem at hand and skips the introduction in favor of brevity.

Methods

Settings

We assume the standard settings the task of annotating chemical formulae given results of a tandem mass spectrometry (MS/MS) run. Concretely, in programming terms, this means we are given a list of tandem spectra in the following format. For each spectrum in this list we are given:

- A precursor m/z (PMZ) value, a scalar in range 1-1000 Da
- Spectrum: A set of peaks, where a peak is a tuple (m/z, intensity), where:
- Each m/z (MZ) is a scalar in 1-PMZ Da range
- Each intensity is a scalar in (0, 1] range
- The size of the set is practically limited to at most 1000.
- PMZ tolerance (aka MS1 tolerance), a measurement error of PMZ values.
- MZ tolerance (aka MS2 tolerance), a measurement error of spectrum MZ values.
- A dataset of 3,514,066 unique valid molecular formulas from biochemical repositories. These are each represented as element count vectors of length 12 (for 12 most biologically active elements).

The task of MSBUDDY is mix the chemical formulae from the given dataset, so we can explain the most peaks in the spectrum and thus find the most likely formula that explanats for the given PMZ, with the given measurement tolerance.

The MSBUDDY Algorithm

The algorithm contains two distinct parts, the first starts with spectra mz values and comes up with candidate formulae (CFs) for PMZ, and the second filters and orders the CFs by the amount and significance of peaks explained. There's

a third part, which additionally statistically predicts false-detection-rate (FDR) for the selected top CF, using AI.

Step 1: From spectra to candidate formulae First, the m/z and intensity values from the spectra are subject to common filtering and preprocessing steps – small intensity peaks (noise) are removed, and only 20 peaks per every 50Da range in 1-1000 are retained, to keep the runtime low for extremely peak-abundant spectra. This is a practical runtime consideration, and not much else.

Next, from physics of fragmentation, we know that each m/z value in the spectrum corresponds to one charged part of the whole molecule. The missing, uncharged part is called neutral loss (NL) and is obtained by subtracting MZ from PMZ.

Thus, neutral loss M (NLmz) and fragment M (Fmz) are obtained. Additionally, by definition NL is not charged.

From NLmz and Fmz, the "pure" mass is obtained by subtracting or adding the mass of the adduct.

From pure mass, the lighter of the NL, Fmz are selected, and searched for, in the formula dataset. Which yields a list of possible chemical formulae that caused the Fmz and NL.

At this point, each Fmz and NL have a set of candidate formulae each. Not all candidate matchups are possible, because for a pair of candidates to be valid, they must form a valid PMZ, both within mass tolerance as well as chemical plausibility, by considering valence and the "usual" ratio of elements.

```
# inputs: spectrum (2, P), dataset (D, 12), pmz float, tolerance float
# outputs: candidate_formula (C, 12)

candidate_formula = []
for mz, intensity in spectrum:
  nlmz = pmz - mz
  fforms = database(mz, tolerance)
  nlforms = database(nlmz, tolerance)
  for fform, nlform in product(fforms, nlforms):
    form = fform + nlform
    if is_valid_formula(form) and
       is_mass_close(form, pmz, tolerance):
      candidate_formula.append(fform)

return candidate_formula
```

Algorithm 1, generating cadidate formulae for PMZ from fragments, the first step in MSBUDDY.

Concretely this means for each peak, we pairwise combine fragment and NL CFs,

and then compare it with the PMZ mass, and filtering by various heuristics. The resulting Precursor CFs are kept in a list for further processing. Only the first 350 Precursor CFs are kept for runtime reasons.

These steps are illustrated in Figure 1. "Block stitching" refers to pairwise combination of Fragment and NL CFs.

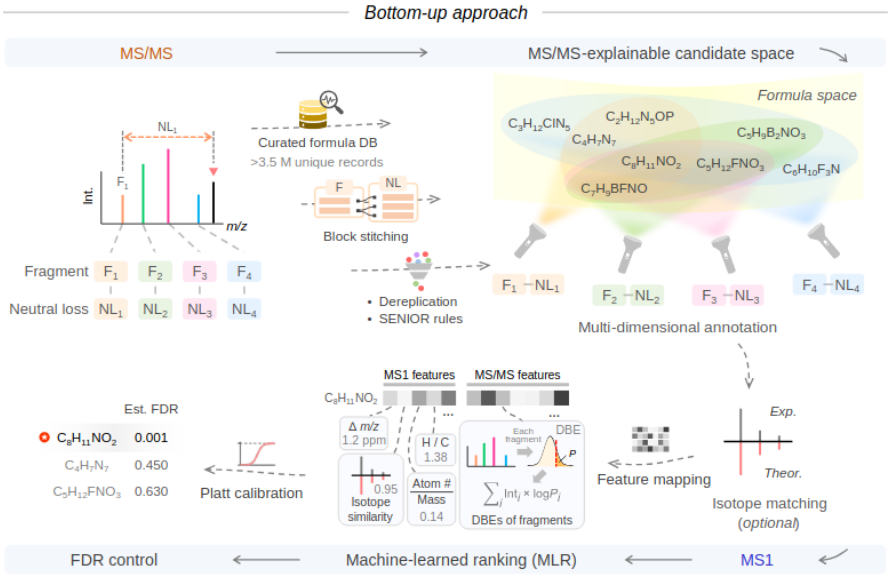

Fig. 1 | Methodological comparison between top-down and bottom-up approaches for molecular formula annotation. The top-down approach generates candidate formulae from MS1 information, followed by MS/MS explanation and candidate ranking. Heuristic scoring is applied to combine MS1 and MS/MS scores to calculate the final score. Bottom-up molecular formula determination prioritizes candidate formulae that can explain MS/MS in a chemically feasible manner. Multi-dimensional annotation drastically narrows down the candidate searching scope. Machine-learned ranking (MLR) is implemented for automated and accurate candidate ranking. False discovery rate (FDR) is controlled via Platt calibration.

Figure 1. Bottom-up annotation used in MSBUDDY. Step 1, as described here, is visualized with steps up to "multi-dimensional annotation".

Step 2: From precursor candidate formulae to spectra At this point, we have at most 350 possible candidate formulae (CFs). We have to order the CFs by likelihood, and we do so, in the following fashion.

First, each a CF combinatorial decomposition matrix (CDM) is created. Concretely, given a CF of "C5H10O", this CF is represented as [5, 10, 0, . . . , 1, 0], where first element is count of carbon (C), second is hydrogen (H) and so on. The decomposition matrix will be all possible *subformula* of this CF. Meaning, we will start by: [1, 0, . . .], and increment up to [5, 0, . . .], then, add one more hydrogen [1, 1, . . .] and re-increment C [2, 1, . . .], and so on for each element.

The resulting CDM has rows of $\prod_{i=0}^{12} N_i$, where N_0 is a count of Carbons, N_1 of

hydrogens and so on.

The CDM has many unreasonable entries, like [4, 0, ...], which would mean formula of C4, which is unchemical. A heuristic filtering is applied to weed out unreasonable formulae from CDM.

Then, for each formula in CDM, mass is calculated and compared to each fragment mass and neutral loss mass. If there's a mass match within the tolerance, the CDM formula is added as possible explanation for the given spectrum peak.

The precursor candidate formula (CF), which explains the most peaks of the given spectrum, using its CDM, is ranked the highest.