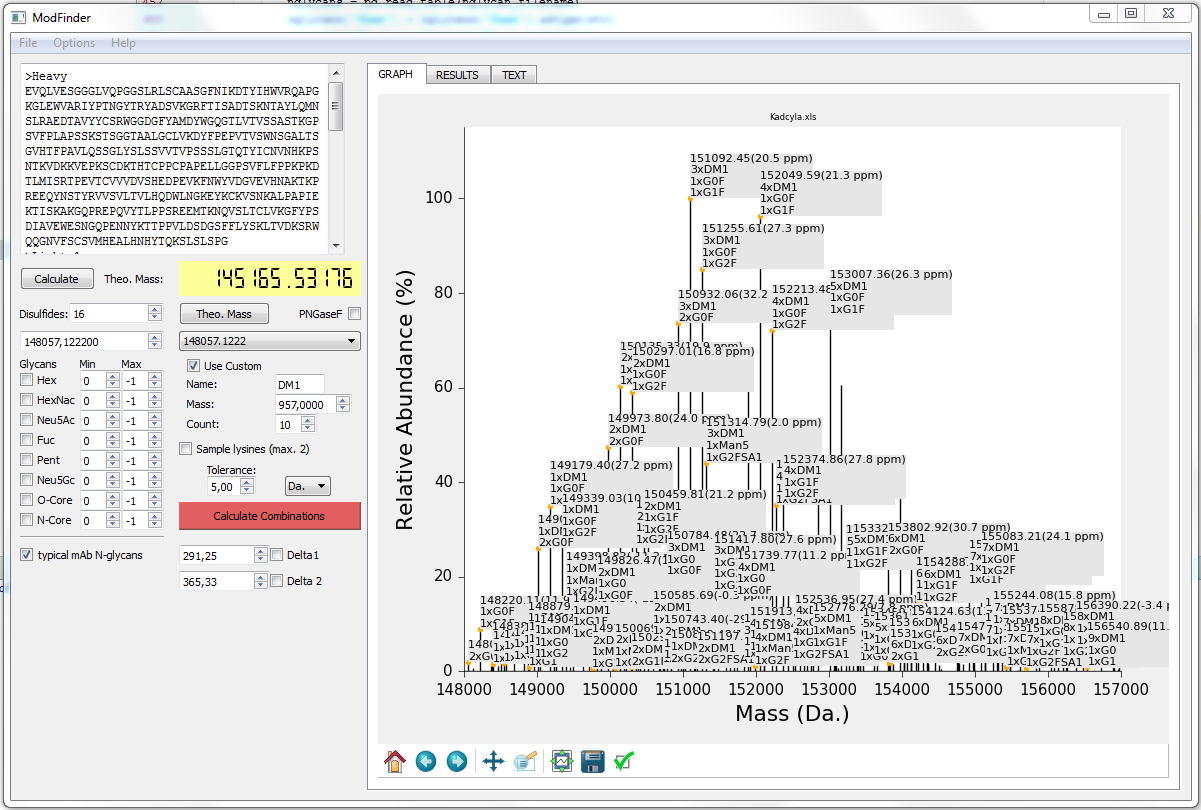
**MoFi – Current Status and Plan for Improvements**  
Wolfgang Skala, 07.06.2017



There are three layers for the program:

1. **Core Algorithm**  
   Problem statement  
   *Given*: An unexplained mass *M*, a list of possible modifications with masses *m*1, … *mn*, which may be used from 0 to *x*max times, and a tolerance *e*.  
   *Return*: A list of tuples (*x*1, …, *xn*), where each tuple satisfies .  
     
   This is an instance of the SUBSET SUM problem, which is known to be NP-complete. Hence, an efficient algorithm is unlikely to exist.
2. **Data layer**This layer describes how to represent and store proteins and modifications.
3. **GUI**This is the visual representation of the program.

Notes on the various levels:

**1. Core algorithm**

* The core algorithm currently comprises an exhaustive search, which is done in C++ via the Python/C API for the sake of speed.  
  Is it possible to implement a comparably fast search in pure python?
* How could we delimit the search space?
* Calculate an upper limit for the search space (2x, for peak with lowest and highest weight)

**2. Data layer**

* Create a unified framework for modifications
* Implement storing and loading sets of modifications   
  (along with current settings for the numbers?)
* Simplify and document the source code

**3. GUI**

* Completely redesign the GUI (see hand-drawn proposal)
* Show “mass of protein (without modifications)”, (calculated from FASTA sequence only)  
   “mass of known modifications” (disulfides, PNGase digest, mandatory custom mods), and  
  “unexplained mass” (peakwise, or in peak selection box)
* Selecting a result in the table should highlight the corresponding peak
* Number of shown delta masses should be customizable
* Implement ppm tolerance (currently, this combobox does nothing)

