Санкт-Петербургский государственный университет

ГАЩЕНКО Екатерина Александровна

Выпускная квалификационная работа

Алгоритмы детектирования нестандартных аминокислот в пептидных антибиотиках

Уровень образования: бакалавриат
Направление: 01.03.01 «Математика»
Основная образовательная программа: СВ.5000.2017 «Математика»

Научный руководитель:
к.ф-м.н., зав. кафедрой биоинформатики и математической биологии СПбАУ РАН им. Ж.И. Алфёрова,
Вяткина Кира Вадимовна

Рецензент:
к.ф-м.н., научный сотрудник лаборатории физико-химических методов исследования Института химической физики им. Н.Н. Семенова Российской академии наук,
Иванов Марк Витальевич

Санкт-Петербург
2021 год

1 Введение

Масс-спектрометрический анализ важен для открытия и изучения различных лекарств. Часто при анализе мы не можем пологаться на стандартные базы данных, так как они не содержат неизвестных модификация и тем более неизвестных соеденений. Перебор возможных формул соединений тоже не подходит, так как это слишком дорогостоящий по времени метод. На сегодняшний один из лучших подходов для решения подобных задач – это поестрение спектрального графа, в котором каждая вершина которого соответствует пику из данного масс-спектра, а ребра проводятся между вершинами, остоящими друг от друга на некоторую массу из "алфавита". При этом ребро направлено от вершины с меньшей массой к вершине с большей массой. Например, пептидное секвинирование может быть выполнено с использованием такого "алфавита"размера 20.
К наиболее известным методам анализа белков и пептидов относятся методы изложенные в статьях [2, 3, 5, 6, 7, 8, 9, 10].
В недавней статье [1] был предложен алгоритм анализа малых молекул, основанный на этих идеях. При этом есть основания полагать, что данный алгоритм может даже применен для анализа модифицированных пептидов и пептидных антибитиков с нестандартными аминокислотами. Цель данной работы была проверка этой гипотезы и адаптация предложенного подхода к указанному случаю.

2 Обозначения

s^l	индексы пиков для спектра l
m_i^l	масса m/z пика $i \in s^l$
m^l	множество m/z спектра l
p_i^l	интенсивность p пика $i \in s^l$
p^l	множество p спектра l
D^l	множество (s^l, m^l, p^l) для спектра l
D	множество (s^l, m^l, p^l) для всех спектров l
Δ	алфавит разностей размера d
Z	максимальный возможный заряд
E^l	граф для спектра l
E_z^l	множество ребер графа E^l для заряда z
E	объединение графов E^l по всем спектрам

3 Алгоритм

3.1 Структура графа

Пусть у нас есть алфавит разностей масс Δ. Для каждого спектра l построим граф. Пусть вершины графа соотвествуют массам спектра. Между вершинам i и j есть ребро $\iff$ существует заряд z, т.ч. $E^l_{z,i,j,k} = 1$.

$$E^l_{z,i,j,k} = \begin{cases} 1, |m_j^l - m_i^l - \dfrac{\Delta_k}{z}| \leq \epsilon \\ 0, |m_j^l - m_i^l - \dfrac{\Delta_k}{z}| > \epsilon \end{cases}$$

Заметим, что $Pr(D|\Delta) = Pr(D|\Delta, E) = Pr(D|E) = \prod_l Pr(D^l|E) = \prod_l Pr(s^l, m^l, p^l|E)$.
Наша цель найти такой алфавит Δ^*, что

$$\Delta^* = \underset{\Delta}{\operatorname{argmax}} Pr(\Delta|D) = \underset{\Delta}{\operatorname{argmax}} \prod_l Pr(s^l, m^l, p^l|E) \cdot Pr(\Delta)$$

3.2 Некомбинаторный метод

Найдем все возможные разности $m_i^l - m_j^l$ для всех пар (i, j), $i \neq j$ и всех спектров l. Отсортируем полученный список по велечине $(p_i^l \cdot p_j^l)$ и возьмем первые d разностей. Это будет наш самый первый вариант алфавита.

3.3 Построение графа

Заметим, что если фиксировать l, z, то E_z^l можно построить с помощью перебора всех разностей и всего алфавита за $O(n^2 \cdot d)$. Где n – размер множества разностей.
Если мы отсортируем множество разностей, фиксируем m_j^l и Δ_k, то мы можем найти m_i^l, т.ч. $|m_j^l - m_i^l - \frac{\Delta_k}{z}| \leq \epsilon$, за время $O(log(n))$ с помощью бинарного поиска. Т.е. граф E_z^l можно построить за $O(n \cdot d \cdot log(n))$.

Рассмотрим еще один подход. Обозначим исходное множество разностей как M. Добавим в массив разностей все элементы вида $-x$, где $x \in M$. Получим новое множество M'. Его размер $2n$. Отсортируем множество разностей M', фиксируем Δ_k.
Можно найти все пары (i, j), т.ч. $|m_j^l - m_i^l - \frac{\Delta_k}{z}| \leq \epsilon$, за время $O(n)$ для фиксированного Δ_k. А тогда граф строится за время $O(n \cdot d)$.
Для этого поставим два указателя в конец и начало массива. Обозначим их $first$ и $last$. Заметим, что

$$last + first = last - first^*$$

, где $first^*$ элемент M. Таким образом, мы хотим найти все такие $first$, $last$, что $\frac{\Delta_k}{z} - \epsilon \leq last + first \leq \frac{\Delta_k}{z} + \epsilon$, при условии, что указатель $last$ всегда находится в "положительной"части M', а указатель $first$ в "отрицательной".
Теперь двигаем два указателя навстречу друг к другу: если $\frac{\Delta_k}{z} - \epsilon > last + first$, то $first = first + 1$, если $\frac{\Delta_k}{z} + \epsilon < last + first$, то $last = last - 1$. Иначе запоминаем пару (i, j) как подходящую. При этом если $first$ или $last$ достигают середины массива (т.е. достигают границ "положительной"и "отрицательной"частей), то дальше мы их не изменяем. Если алгоритму необходимо в какой-то момент передвинуть один из указателей за эту границу, то алгоритм останавливается. Корректность алгоритма почти очевидна: это известный подход поиска двух элементов в массиве с заданной суммой, поэтому приводить доказательсво мы не станем.

Algorithm 1 FindMasses(Δ, ϵ, M)

$M' = (-M).concate(M)$
$Ans = [\,]$
$f = 0$
$l = M'.size - 1$
while $True$ **do**
  if $M'[f] + M'[l] < \Delta - \epsilon$ **then**
    if $f < M.size - 1$ **then**
      $f += 1$
    else
      $break$
    end if
  else if $M'[f] + M'[l] > \Delta + \epsilon$ **then**
    if $l > M.size + 1$ **then**
      $l -= 1$
    else
      $break$
    end if
  else
    $Ans.append((M[f], M[l - M.size]))$
  end if
end while
return Ans

3.4 Семплирование

Давайте максимизируем

$$L(\Delta) = \prod_l Pr(s^l, m^l, p^l | E) \cdot Pr(\Delta)$$

Будем использовать для этого семплирование. Фиксируем Δ. На нулевом шаге это алфавит, полученный при помощи некомбинаторного метода.
Случайно выберем $i \in [0, d-1]$. Теперь есть два способа изменить Δ_i. Способ можно выбрать либо случайно, либо применить оба, а затем выбрать наилучший. Мы используем второй вариант.
Способ первый: случайно выбираем одну из разностей из множества разностей.
Способ второй: скалируем Δ_i. Мы хотим получить то же значение m/z, но для другого z. Например, если $\Delta_i = 3$, то мы можем сделать ее равной 1. Тогда $\Delta_{i,old}/3 = \Delta_i/1$. Или пусть теперь $\Delta_i = 2$, тогда $\Delta_{i,old}/3 = \Delta_i/2$. И так далее. Для этого случайно выбираем число из множества $\{3, 3/2, 2, 2/3, 1/2, 1/3\}$ и домножаем на него.
Обозначим через Δ' новый алфавит. Если $L(\Delta') > L(\Delta)$, то оставляем Δ', иначе возвращаем Δ. Продолжаем семплирование.

Заметим, что в расчетах $L(\Delta)$ есть $Pr(s^l, m^l, p^l|E)$. Эта вероятность вычисляется с помощью графа E (как ее вычислять, описано ниже). Чтобы быстрее строить граф E' для Δ' достаточно найти все пары (i, j) для которых выполнено $|m_j^l - m_i^l - \frac{\Delta_i}{z}| \leq \epsilon$ для некоторого z, удалить их и затем добавить новые ребра для (i, j), т.ч. $|m_j^l - m_i^l - \frac{\Delta'_i}{z}| \leq \epsilon$.

3.5 Вычисление $Pr(s^l, m^l, p^l|E)$

Разобьем граф на подграфы, ребрам каждого из которых соответствует один и тот же заряд:

$$Pr(D^l|E^l) = \prod_z Pr(D^l|E_z^l)$$

Пусть $g(E_z^l)$ – множество компонент связности в графе E_z^l. Компоненты связности можно найти с помощью алгоритма поиска в глубину для графа E_z^l. Тогда

$$Pr(D^l|E_z^l) = \sum_{G \in g(E_z^l)} Pr(D^l|G)$$

Рассмотрим связный граф G. Тогда $Pr(D^l|G)$ вычисляется как произведение интенсивностей всех ребер в G. Интенсивность ребра – произведение интенсивностей вершин на его концах. Обозначим множество ребер графа через $E(G)$. Тогда

$$Pr(D^l|G) = \prod_{(i,j) \in E(G)} p_i \cdot p_j$$

Итоговая формула

$$Pr(s^l, m^l, p^l|E) = \prod_z \sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} p_i \cdot p_j$$

Algorithm 2 L(E, P)

1: $L = 1$
2: $C = FindComponents(E)$
3: **for all** $l \in [0, L-1]$ **do**
4: $\quad L* = Pr(E[l], P[l], C[l])$
5: **end for**
6: **return** L

M – множество m/z всех спектров.
P – множество интенсивностей всех спектров.
E – списки ребер графов для всех спектров.
C – для каждого спектра, для каждого заряда и для каждой вершины содержит номер компоненты связности, к которому относится в вершина.
Метод $FindComponents$ находит все компоненты связности графа для каждого спектра l и для каждого заряда z с помощью DFS.

Algorithm 3 Pr(E, P, C)

1: $Pr = 1$
2: **for all** $z \in [0, Z-1]$ **do**
3: $\quad Pr* = PrSum(E[z], P, C[z])$
4: **end for**
5: **return** Pr

Algorithm 4 PrSum(E, P, C)

1: $PrSum = 0$
2: $N = P.size$
3: $Pr = [1] * NumOfConnectivityComponents$
4: **for all** $(i, j) \in E$ **do**
5: $\quad Pr[C[i]]* = P[i] * P[j]$
6: **end for**
7: **for all** $p \in Pr$ **do**
8: $\quad PrSum+ = p$
9: **end for**
10: **return** $Prsum$

3.6 Вычисление $Pr(\Delta)$

В алфавите нам нельзя допускать появления:

- слишком маленьких разностей
- слишком больших разностей
- двух близких разностей
- двух разностей, из которых получаются одинаковые m/z при некоторых z

Введем в рассмотрение следующие велечины

$$Pr_1(\Delta) = \begin{cases} 1, \forall k\ 1 - \epsilon < \Delta_k < MaxDelta \\ 0, \text{else} \end{cases}$$

$$Pr_2(\Delta_1, \Delta_2) = \begin{cases} 0, \exists z_1, z_2\ 1 - \dfrac{\Delta_1 \cdot z_1}{\Delta_2 \cdot z_2} \in [1 - \epsilon, 1 + \epsilon] \\ 1, \text{else} \end{cases}$$

$$Pr_3(\Delta_1, \Delta_2) = \begin{cases} 0, |\Delta_1 - \Delta_2| < \dfrac{1}{2} \\ 1, \text{else} \end{cases}$$

Тогда

$$Pr(\Delta) = Pr_1(\Delta) \cdot \prod_{i \neq j} Pr_2(\Delta_i, \Delta_j) \cdot \prod_{i \neq j} Pr_3(\Delta_i, \Delta_j)$$

3.7 Улучшения

Во время работы алгоритма можно "подсказывать"семплированию. Например, если мы знаем, что ожидаем увидеть какие-то значения из определенного множества, то можно на случайных шагах, но не слишком часто, давать алгоритму не случайную массу из множества разностей, а массу из нашего множества. Если при этом $L(\Delta)$ не ухудшается, то можно принять новый алфавит.
После работы алгоритма необходимо проверить, нельзя ли улучшить текущий алфавит. Мы могли, например, добавить массу Δ_k, но при этом существует такой заряд z, что $\Delta'_k = \frac{\Delta_k}{z}$ улучшает $L(\Delta)$. Поэтому для каждого $k \in [0, d-1]$ мы проверяем, не увеличивает ли значение $L(\Delta)$ замена Δ_k на $\Delta_k/1$, $\Delta_k/2$, .., Δ_k/Z.
Для улучшения производительности можно распараллелить семплирование: выбирать несколько $i \in [0, d-1]$, изменять их и выбирать наилучший результат.

3.8 Особенности

Для наших эксперимнтов мы вырали следующие значения:

- $Z = 3$
- $MaxDelta = 250$
- $d = 20$
- $\epsilon = 0.05$

Для предобработки интенсивностей: удалим из рассмотрения все слишком большие интенсивности (либо k максимальных, либо те, которые в k раз больше медианы всех интенсивностей). Отнормируем оставшиеся интенсивности, поделив на сумму всех интенсивностей в спектре. Теперь мы можем рассматривать интенсивности как вероятности.
Может возникнуть проблема из-за того, что мы слишком часто берем произведение маленьких чисел. Из-за нехватки точности, все вырождается в 0. Чтобы избежать этого, можно вычислять не $L(\Delta)$, а $log(L(\Delta))$. Но теперь мы можем, например, получить слишком большие по модулю отрицательные числа и нам снова не хватит точности.
Можно домножить каждую интенсивность на достаточно большую константу C. После этого интенсивности перестанут быть вероятностями, но все еще будут отнормированы.
Тогда формула имеет вид:

$$Pr(s^l, m^l, p^l|E) = \prod_z \sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} (p_i \cdot p_j \cdot C^2) = \prod_z \sum_{G \in g(E_z^l)} (\prod_{(i,j) \in E(G)} p_i \cdot p_j) \cdot C^{2|E(G)|}$$

Заметим, что теперь мы дополнительно "поощряем"компоненты связности с большим числом ребер. Если мы заранее знаем, что все компоненты связности имеют одинаковый размер, то лучше использовать следующую формулу:

$$log(Pr(s^l, m^l, p^l|E)) = \sum_z log((\sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} p_i \cdot p_j) \cdot C)$$

В этом случае значение

$$\sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} p_i \cdot p_j$$

еще не будет достаточно маленьким, чтобы обратиться в ноль, поэтому на этом шаге еще имеет смысл домножать на что-то.
При этом C будет всегда в одной и той же степени, вне зависимости от графа:

$$logL(\Delta) = \sum_l \sum_z (log(\sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} p_i \cdot p_j) + C) = \sum_l \sum_z log(\sum_{G \in g(E_z^l)} \prod_{(i,j) \in E(G)} p_i \cdot p_j) + C \cdot L \cdot Z$$

Здесь нет множителя $Pr(\Delta)$, так как если он равен 0, то $L(\Delta)$ сразу равно 0 и его неинтересно рассматривать. А другое возможное значение $Pr(\Delta)$ это 1. Поэтому его можно сократить.

4 Результаты

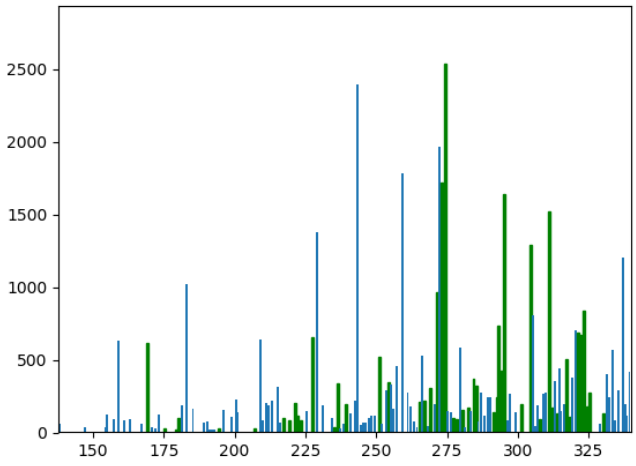

Рис. 1: Спектр №1. Пики, которые принадлежат одной компоненте связности, отмечены зеленым

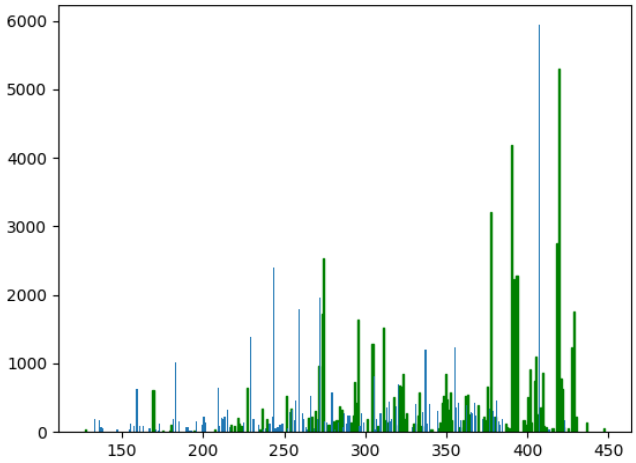

Рис. 2: Спектр №2. Пики, которые принадлежат одной компоненте связности, отмечены зеленым

На рисунках 1 и 2 можно увидеть, что компоненты связности получаются достаточно большими.
При этом они в дейтсвительности не такие огромные, как может сначала показаться: из-за того, что проводим вычисления с допуском некоторой погрешности, у нас в одну компоненту связности добавляются точки со схожими массами. Поэтому если, например, объеденить вершины со схожими массами, в одной такой компоненте связности будет около 10 вершин.

Масса	возможное соединение
142.168	AA
217.006	NC
56.863	G
0.99024	
78.3939	
80.3625	
28.0038	
2.05728	
190.021	CS
48.2414	
67.4305	
41.9502	
114.043	N
3.11802	
184.094	PS
27.2958	
6.12299	
17.1668	NH_3
52.0472	
192.496	

Результат работы алгоритма после 20000 итераций и возможное соединение, которое имеет схожую массу. Алгоритм был протестирован на тестовом наборе данных* для карбоангидразы, полученные результаты приведены в следующей таблице:

Масса	возможное соединение
36.0212	
187.975	GM
77.9372	
56.9215	G
42.0253	
1.99812	
101.068	T
24.2838	
28.0309	
99.0572	V
61.3288	
185.927	EG
98.0086	
19.7259	
114.044	N
60.16	
54.0291	
129.043	E
47.9103	
191.975	

Тестовый набор данных представляет собой множество тандемных масс-спектров, снятых с триптических пептидов карбоангидразы, содержащий 21880 МС/МС-спектров. Детали эксперимента приведены в работе [11] (файл 140411_QE_Cah-1.mzXML).

4.1 Ссылки

Репозиторий с кодом: Github

Список литературы

[1] Patrick A. Kreitzberg, Marshall Bern; Qingbo Shu; Fuquan Yang; Oliver Serang Alphabet Projection of Spectra

[2] Ma, B.; Zhang, K.; Hendrie, C.; Liang, C.; Li, M.; Doherty-Kirby, A.; Lajoie, G. PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. 2003, 17, 2337-42.

[3] Frank, A.; Pevzner, P. PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. Anal. Chem. 2005, 77, 964-973.

[4] Chi, H.; Chen, H.; He, K.; Wu, L.; Yang, B.; Sun, R.-X.; Liu, J.; Zeng, W.-F.; Song, C.-Q.; He, S.-M.; Dong, M.-Q. pNovo+: De Novo Peptide Sequencing Using Complementary HCD and ETD Tandem Mass Spectra. J. Proteome Res. 2013, 12, 615-625.

[5] Taylor, J. A.; Johnson, R. S. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. 1997, 11, 1067-1075.

[6] Dancik, V.; Addona, T. A.; Clauser, K. R.; Vath, J. E.; Pevzner, P. A. De Novo Peptide Sequencing via Tandem Mass Spectrometry. J. Comput. Biol. 1999, 6, 327-342.

[7] Bandeira, N.; Tang, H.; Bafna, V.; Pevzner, P. Shotgun Protein Sequencing by Tandem Mass Spectra Assembly. Anal. Chem. 2004, 76, 7221-7233.

[8] Bandeira, N.; Clauser, K. R.; Pevzner, P. A. Shotgun Protein Sequencing: Assembly of Peptide Tandem Mass Spectra from Mixtures of Modified Proteins. Mol. Cell. Proteomics 2007, 6, 1123-1134.

[9] Bandeira, N.; Pham, V.; Pevzner, P.; Arnott, D.; Lill, J. R. Automated de novo protein sequencing of monoclonal antibodies. Nat. Biotechnol. 2008, 26, 1336-1338.

[10] Kira Vyatkina; Si Wu; Lennard J. M. Dekker; Martijn M. VanDuijn; Xiaowen Liu; Nikola Tolic; Mikhail Dvorkin; Sonya Alexandrova; Theo M. Luider; Ljiljana Pasa-Tolic; Pavel A. Pevzner De Novo Sequencing of Peptides from Top-Down Tandem Mass Spectra.

[11] Kira Vyatkina, Lennard J.M. Dekker, Si Wu, Martijn M. VanDuijn, Xiaowen Liu, Nikola Tolić, Theo M. Luider, Ljiljana Paša-Tolić De Novo Sequencing of Peptides from High-Resolution Bottom-Up Tandem Mass Spectra Using Top-Down Intended Methods