

Лабораторная работа № 5

Поиск регуляторного мотива

1 Мотив ДНК

Мотив ДНК определяется как подстрока строки ДНК, имеющая определенное биологическое значение. Мотив ДНК представляет собой небольшой кусок нуклеотидов А, Т, С или G длиной всего от 5 до 25 пар оснований. Известно, что регуляторный мотив слабо меняется в процессе эволюции,

Мотивы, присутствующие в экзонах (кодирующая часть генома), определяют структуру белка или метку белков, которые будут отправлены в определенные части клетки для осуществления таких процессов, как, например, фосфорилирование. Мотивы, присутствующие в интронах (которые составляют некодирующую часть генома), обычно представляют собой регуляторные последовательности, которые определяют степень экспрессии генов и сайты связывания белков.

Конкретные мотивы последовательностей могут функционировать как регуляторные последовательности, контролирующие биосинтез, или как сигнальные последовательности, которые направляют молекулу в конкретный сайт внутри клетки или регулируют ее созревание. Поскольку регуляторная функция этих последовательностей важна, считается, что они сохраняются в течение длительных периодов эволюции. В некоторых случаях эволюционное родство можно оценить по степени сохранности этих участков.

Мотив длиной k представляет собой k -мер. k -мер — это подстрока длины k — последовательность из k подряд идущих символов в строке (или нуклеотидов в последовательности ДНК, РНК или аминокислот в задачах биоинформатики).

Благодаря появлению доступных методов для массового прочтения последовательностей ДНК, стремительно растет объем прямых данных по ДНК-белковому узнаванию как *in vivo*, так и *in vitro*. Компьютерный анализ характерных ДНК-паттернов, мотивов, распознаваемых факторами транскрипции, потенциально позволяет изучать структуру регуляторных районов с однонуклеотидным разрешением. Однако, классические инструменты для анализа мотивов не справляются с возрастающими объемами данных и не учитывают специфику современных экспериментальных подходов. Поэтому требуются специальные программы для анализа регуляторных мотивов.

2 Мотивы последовательности и консенсусные последовательности

Оказывается, что даже выполняющие одну и ту же функцию мотивы могут отличаться друг от друга. Наша задача найти консенсусную последовательность для этих мотивов.

Консенсусная последовательность (consensus sequence) — искусственная последовательность ДНК, содержащая в каждой позиции нуклеотид, наиболее часто встречающийся у нескольких гомологичных последовательностей. Обычно такие последовательности характерны для генов, кодирующих один и тот же белок у различных организмов.

Все реальные мотивы не должны отличаться от консенсусной последовательности более чем несколькими заменами.

Пример. Например, рассмотрим участок ДНК

```
1 tagtggctcttttgagtgTAGATCTGAAGgggaaagtatttccaccagttcggggtcacccagcagggcagggtgacttaat
2 cgcgactcggcgctcacagttatcgacgttagacaaaacggagtTGGATCCGAAactggagtttaatcggagtcctt
3 gttacttgtgagcctgggTAGACCCGAAatataattgttggctgcatagcggagctgacatacagtaggggaaatgcgt
4 aacatcaggctttgattaaacaatttaagcacgTAATCCGAAattgacctgatgacaatacgaacatgccggctccggg
5 accaccggataggctgcttatTAGGTCCAAAaggtagtatcgtaataatggctcagccatgtcaatgtcggcattccac
6 TAGATTCGAAatcgatcggtttctccctctgtgggttaacgaggggtccgacctgtctcgcatgtgccgaacttgatccc
7 gaaatggttcggtgcgatatcaggccgttctcttaacttggcggtgCAGATCCGAAcgtctctggaggggtcgtgcgcta
8 atgtatactagacattctaacgctcgcttattggcggagaccatttctccactacaagaggctactgtgTAGATCCGTA
9 ttcttacacccttcttTAGATCCAAAcctgttggcgccatcttcttttcgagtccttgtaacctcatttgctctgatgac
10 ctacctatgtaaaacaacatctactaacgtagtcgggtcttctctgatctgccctaacctacaggTCGATCCGAAattcg
```

Мы видим, что в этом участке 10 раз повторяется один и тот же 10-мер с точностью до 1-2 нуклеотидов. Эти 10-меры выделены заглавными буквами и жирным шрифтом, ошибочные нуклеотиды выделены красным. Расстояние Хэмминга, то есть количество разных символов в каждой позиции из двух строк одинаковой длины, в этом примере меньше или равно 2.

Учитывая набор мотивов, консенсусная последовательность представляет собой последовательность, полученную путем взятия наиболее частых нуклеотидов в каждом положении.

Например, консенсусная последовательность

TAGATCTGAA

TGGATCCGAA

TAGACCCGAA

TAAATCCGAA

TAGGTCCAAA

TAGATTCTGAA

CAGATCCGAA

TAGATCCGTA

TAGATCCAAA

TCGATCCGAA

представляет собой TAGATCCGAA.

Если мы напишем одну последовательность одну под другой и окрасим нуклеотиды в разный цвет, это будет легче увидеть: Т — наиболее распространенный остаток в положении 1, А — наиболее распространенный остаток в положении 2 и так далее.

Таблица 1. Матрица Motifs

№	1	2	3	4	5	6	7	8	9	10
1	T	A	G	A	T	C	t	G	A	A
2	T	g	G	A	T	C	C	G	A	A
3	T	A	G	A	c	C	C	G	A	A
4	T	A	a	A	T	C	C	G	A	A
5	T	A	G	g	T	C	C	a	A	A
6	T	A	G	A	T	t	C	G	A	A
7	c	A	G	A	T	C	C	G	A	A
8	T	A	G	A	T	C	C	G	t	A
9	T	A	G	A	T	C	C	a	A	A
10	T	c	G	A	T	C	C	G	A	A

Следует отметить, что позиция 10 является наиболее консервативной (нуклеотид А полностью сохраняется в этих положениях), в то время как позиции 2 и 8 являются наименее консервативными.

В Python мы представим матрицу мотива как список строк Motifs. Будем называть i-ю строку этой матрицы Motifs[i]; a j-й символ этой строки — Motifs[i][j].

Матрице Motifs поставим в соответствие число $\text{Score}(\text{Motifs})$ — сумму числа непопулярных (строчных) букв в матрице мотива.

Для нашего примера

$$\text{Score}(\text{Motifs}) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 2 + 1 + 0 = 10.$$

Построим теперь $4 \times k$ -матрицу, обозначаемую $\text{Count}(\text{Motifs})$ подсчета числа вхождений каждого нуклеотида в каждый столбец матрицы мотива. Элемент (i, j) матрицы $\text{Count}(\text{Motifs})$ содержит то число раз, которое нуклеотид с номером i появился в j -м столбце мотива.

Таблица 2. Матрица $\text{Count}(\text{Motifs})$

Нуклеотид	1	2	3	4	5	6	7	8	9	10
A	0	8	1	9	0	0	0	2	9	10
C	1	0	0	0	1	9	9	0	0	0
G	0	1	9	1	0	0	0	8	0	0
T	9	1	0	0	9	1	1	0	1	0

Один из способов представления такой матрицы в Python, состоит в том, чтобы создать список (list) для каждой строки матрицы, а затем организовать эти списки в больший словарь (тем самым создавая словарь, ключами которого являются нуклеотиды и значения — списки).

По матрице $\text{Count}(\text{Motifs})$ мы можем образовать **консенсусную последовательность**, обозначаемую $\text{Consensus}(\text{Motifs})$, из наиболее часто встречающихся нуклеотидов в каждом столбце матрицы Motifs. Если мы правильно выбрали мотивы, то $\text{Consensus}(\text{Motifs})$ и будет давать предполагаемый регуляторный мотив.

Например, на основе матрицы $\text{Count}(\text{Motifs})$ (см. таблицу 2) для матрицы Motifs (см. таблицу 1) консенсусной строкой является TAGATCCGAA.

На практике, для визуального представления мотивов часто используют логотип последовательностей — графическое представление консервативности каждой позиции в мотиве.

Приведем логотип последовательностей матрицы Motifs (см. таблицу 1 и рис. 1), построенный на сайте

<https://weblogo.berkeley.edu/logo.cgi>



Рис. 1: Логотип последовательностей матрицы Motifs

Задания

1. Написать программу поиска консенсусной последовательности. Проиллюстрировать на примере из своего варианта. Вывести матрицу Motifs и матрицу Count(Motifs).
2. Используя сайт <https://weblogo.berkeley.edu/logo.cgi> построить логотип последовательностей матрицы Motifs на примере из своего варианта.

Варианты

1. АТТАТАСТТА
AACCTGTGCT
CATTCACATT
TCAAAGCCAT
CTTCCAGAAA
GCTTAGCAAC
ACGAAAAAAAAA
AAAAATTGTT
TAGATCTGAA
TATTTTTCAT
CTGAGAGCAT
TTAGAAAGGT

2. TCAATAAGTC
AGTTCAACAA
ATATTTATCT
TGAACCAAGT
CCTTTTTTTC
TAAGCTTTTT
ACAAC TTTGT
TCGGTTCAAA
AGAAATGCAA
TCCCTTATGT
TATTCCTTCA
TTTGTTTATA
3. AATCATTAAT
ATGATGGTTT
GAATGAGCTT
TTAAAGATGA
AAAAAAATCA
CCCTTTCATA
GTTGTTTAAA
GCAGCTTTTT
CTGTTAAATA
TGGAGGTCAT
TTTTTACAGA
AGAGAAACCG
4. AAATATACCA
AGATACTGTT
GCCTCCTAAT

TTGTACCATT
TGAAGTACCA
AAATTGACTA
AATACATTTT
CCTATTATGT
TTCTACATTT
TCAAGATGTA
CCTAAGATTG
CAGTGGGCCA

5. AGGCAAAGGG
GAAGTGAGCA
TGTCTTCACA
TGTCAGAGCA
GAAGACTGAG
AGTGAATGAG
GAAGTGCCAG
ACACTTTTAA
ACCATCAGGT
CTTGTTAGAA
CTCATTCATT
ACCATGAGAA

6. CAGCAAGGGG
GATATCCGCC
TCTATAATCA
ATCACCTTCC
ACTAATTGCC
TTCCTTAACA

CTGGGAATTA
ACAATTTTAC
ATGAGATTTG
GGTGGGGACA
CAGAGCTAAA
CTATATCAAC

7. TATCTACAAT
TTGCTTGATA
AGGAACAATT
TGTACACATT
CCTGAGTCCA
CCCAGAAGGA
GTGAAGTACA
GTCTGAATGG
AAAGTGTAGT
GGAAGAAGTT
TCTAGTAGCA
AAAAAGGGGA

8. CTATTCTCTG
AAAGCTTTGT
AATTAAACTA
AAGAGAAGAA
ACAAAGGTAC
AGCCAGAAAG
GAAGCCCTGG
TCATAGAAAG
TAAATAGAGT

GTTCACTCAG

AAGAACAAGG

TTGTGTACCA

9. CAACTATGGC

CAGTAAACTC

AGTGGAGAAG

AGGACTTAAA

CAATGGCTGG

GAAATGTAGA

GGGGTCACGA

ATCCTAAAAC

TCTGGACATG

AGATAGAATG

TTCACGTAAT

10. GCGGCCGCTG

CGTCCGCCAG

TAGCGGGTTG

CAGGCGCACC

CTCCCCTCCA

GGGCGGCCAC

GCAGCTGTCA

GTGCCGCCGC

CACTGCGAGG

CTGGAGCGGA

GCCCGGGTGG

CCGAGGGAGG

11. GGACCCCGCG
AGAGGGCCGC
GCGCCGGCCG
CCGCCGCCCC
GGCGCCAGG
CTCGGTGCTG
GAGAGTCATG
CCTGTGAGCC
CTGGGCACCT
CCTGATGTCC
TGCGAGGTCA
CGGTGTTCCC

12. AAACCTCAGG
GTTGCCCTGC
CCCACTCCAG
AGGCTCTCAG
GCCCCACCCC
GGAGCCCTCT
GTGCGGAGCC
GCCTCCTCCT
GGCCAGTTCC
CCAGTAGTCC
TGAAGGGAGA
CCTGCTGTGT

13. GGAGCCTCTT
CTGGGACCCA
GCCATGAGTG

TGGAGCTGAG
CAACTGAACC
TGAAACTCTT
CCACTGTGAG
TCAAGGAGGC
TTTTCCGCAC
ATGAAGGACG
CTGAGCGGGA
AGGACTCCTC

14. TCTGCCTGCA
GTTGTAGCGA
GTGGACCAGC
ACCAGGGGCT
CTCTAGACTG
CCCCTCCTCC
ATCGCCTTCC
CTGCCTCTCC
AGGACAGAGC
AGCCACGTCT
GCACACCTCG
CCCTCTTTAC

15. ACTCAGTTTT
CAGAGCACGT
TTCTCCTATT
TCCTGCGGGT
TGCAGCGCCT
ACTTGAACCT

ACTCAGACCA
CCTACTTCTC
TAGCAGCACT
GGGCGTCCCT
TTCAGCAAGA
CGATGGCTGT

16. GCTCAGGCAG
CTGGCGCTCC
TCCTCTGGAA
GAACTACACC
CTGCAGAAGC
GGAAGGTCCT
GGTGACGGTC
CTGGAACTCT
TCCTGCCATT
GCTGTTTTCT
GGGATCCTCA
TCTGGCTCCG

17. ATAAAGAACT
ACCTGCGACT
GGGTAATTTA
TGAAGAAAAG
AGGTTTAATT
AACTCACAGT
TTCACAGGCT
GTACAGGTAG
CATGGCTGGG

AGACCTCAGG

AAACTTACAA

TCATGGCGGA

18. GCCCCAGGCT

CTGTCAACTG

CCCCATCTCA

AGTCTTCAAA

GCCTACTTCC

CATGAGTCAA

GAGTAACTCA

CTGTTCTGAA

GCTGTGTTAA

AGCATTGTAT

TACTCTGTTC

TCACACTGCT

19. AAGTCATTTT

GCAGCTAAGA

TTATTTGTCC

CCTGTTACAT

AACAGGGGGT

TATTCATAGA

GGTTATTCAT

AAACAGGTTA

TTCATCGAGT

TAAGTATAGA

ATGCAGGTCT

CCTGACCCTT

20. ATAAAAGGTC

TTACCTAATA

TATTTTACAA

TAAATACCCA

CATGGTACCC

TGAATCCATG

CTAAATTGAT

CTATCTCATA

TAGGCATAAT

GTGTAAAAGA

CTCTAGTGGT

CAAATGACTC

21. TAATGTATTA

ATGTTTTATC

TGGGGGTTCT

CTTTCTCCCT

TTACTCACTG

GCTAGAAGCC

TGTACAATTC

CTTCAAATTG

AGCTAATCTG

TAGGTACTCC

AAAATCACTG

CAGTTGATAG

22. AAAATATCTT

AAACAAAAT

ATAGCCCATG

CCAGTCAAAC
GCTAGGGAAG
ATTATTTTCT
CAGAAAACCA
CATTGACAAT
ATCAGACACA
TGGAAGCAGA
ATATCTTAAA
TAATAGTTAA

23. GTTTATAGGC
AATAAGCTAT
TATCCTCTAG
GACTTAGCGA
ACTGGCTGTC
AGCCAACCTT
ATATTTGGAT
ACTCCTGGAA
GGATAAAATA
TGACAGTTAA
TTGGTTTATG
TGAGAAGAGT

24. ATTACTTGGG
CTACTCAGTG
AAGCTAATAC
CCAACATATG
ACCTTTCTTC
ACAGCCTTAT

TAATATCTTT
CACATCACAA
GTGGTGCCAG
AATTGAATAG
AGAGACTTTA
AAACAAGGCT

25. TTCAAAACTA
CTCAGCAAGT
CAGCTATGAA
AACCTTCAGA
CGTTTTCCAG
CAAATAGTGA
AAGGACATTA
AATGTTTTGG
TCATCTAAAG
AAACAAAGAG
CTGTTGAATG
GAAAAATAAA