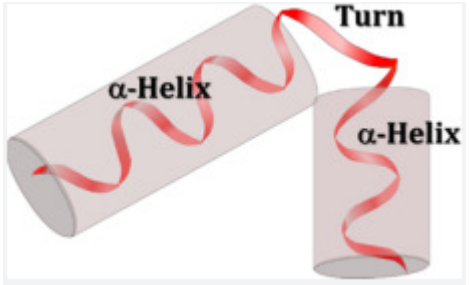
Мотивы в последовательностях

С.А.Спирин

3 апреля 2026

Что такое «мотив»?

Структурные мотивы



Спираль-поворот-спираль
(helix-turn-helix, HTH)

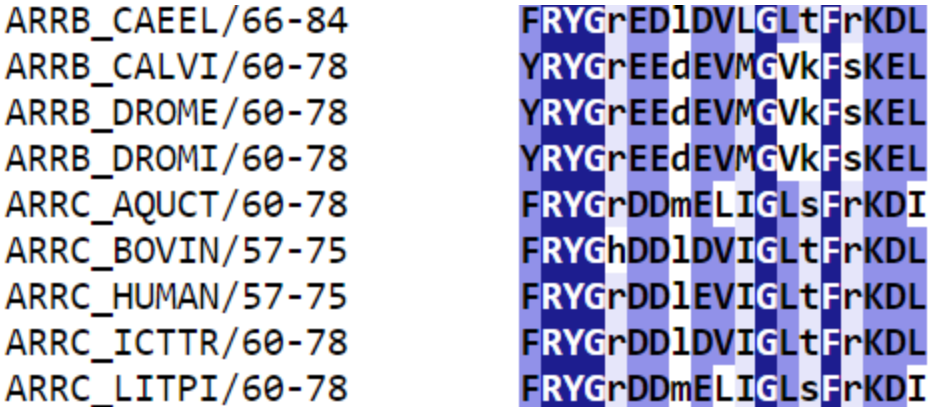
Бета-шпилька
(beta hairpin)

Что такое «мотив»?

Мотивы в последовательностях белков

VLQRRRGSIPQ

cAMP- and cGMP-dependent
protein kinase phosphorylation site



Мотив, характерный для белков-аррестинов

Что такое «сигнал»?

Мы под сигналом будем понимать мотив, который узнаётся какими-либо клеточными системами

Примеры:

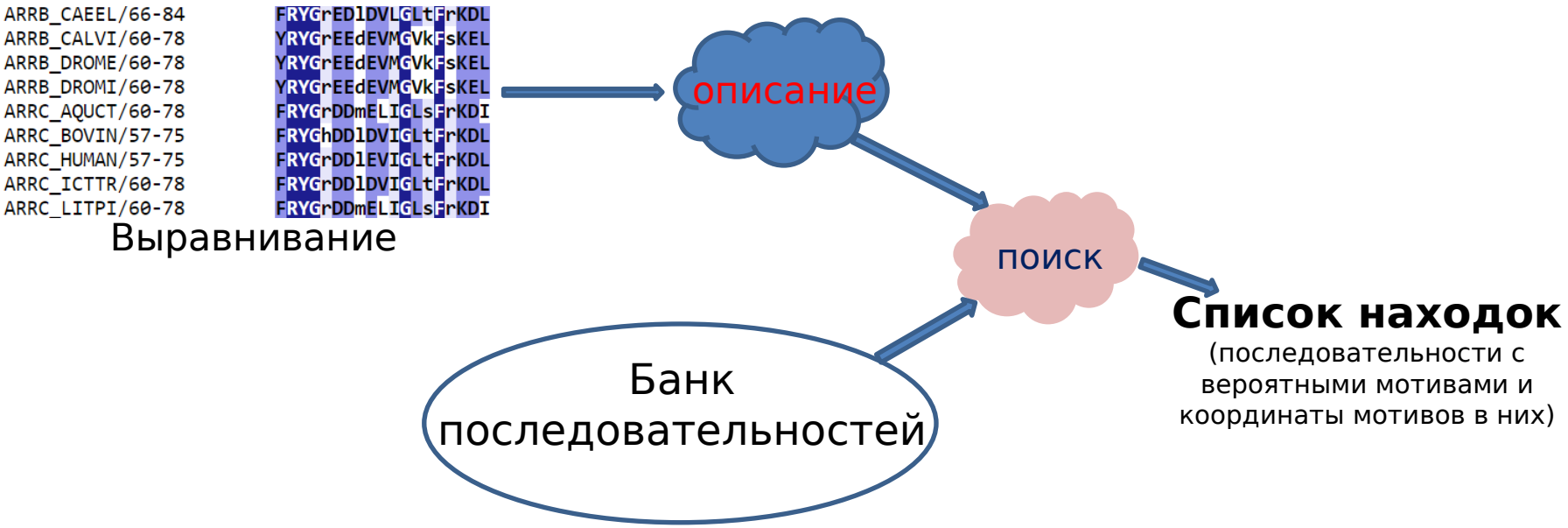
1. Сигнал ядерной локализации (в белках)
2. Последовательность Шайна – Дальгарно (в РНК)
3. Сайт связывания транскрипционного фактора (на ДНК)

Мотивы, характерные для функциональных классов белков (как в примере с аррестинами), обычно не называют сигналами.
Хотя граница между двумя понятиями не слишком чёткая

Как нам формализовать мотив?

Дано выравнивание участков последовательностей, представляющих какой-либо мотив.

Хочется на основе этого выравнивания так описать мотив, чтобы его было легко искать в других последовательностях.



Варианты описания мотива

1. Консенсус
2. Паттерн
3. Частотная матрица
4. Позиционная весовая матрица (PWM)

Консенсус

Самая частая буква в каждой колонке
(возможны варианты)

```
FRYGREDLDVLGLTFRKDL
YRYGREEDEVMGVKFSKEL
YRYGREEDEVMGVKFSKEL
YRYGREEDEVMGVKFSKEL
FRYGRDDMELIGLSFRKDI
FRYGHDDLDVIGLTFRKDL
FRYGRDDLEVIGLTFRKDL
FRYGRDDLDVIGLTFRKDL
FRYGRDDMELIGLSFRKDI
.RYG.......G..F.K..
fRYGrdd.eviGl.FrKdl
fRYGrddleviGltFrKdl
```

строгий

с порогом 50%

без порога

Консенсус

Преимущество: просто и наглядно

Недостаток: масса информации пропадает

Как искать: прикладываем ко всем позициям последовательности и смотрим, соответствует ли

Как вариант можно различать строгие и нестрогие позиции, в строгих не допускать разночтений, а в нестрогих допускать заранее оговорённое количество (одно, два, …)

Консенсус часто используется для «человеческого» описания мотива, но редко для поиска мотива в новых последовательностях

Паттерн

Син. «регулярное выражение»

FRYGREDLDVLGLTFRKDL
YRYGREEDEVMGVKFSKEL
YRYGREEDEVMGVKFSKEL
YRYGREEDEVMGVKFSKEL
FRYGRDDMELIGLSFRKDI
FRYGHDDLDVIGLTFRKDL
FRYGRDDLEVIGLTFRKDL
FRYGRDDLDVIGLTFRKDL
FRYGRDDMELIGLSFRKDI

[FY]-R-Y-G-[RH]-[DE](2)-x-[DE]-[LIV]-[LIM]-G-[LV]-x-F-x-K-[DE]-[LI]

Полное описание синтаксиса см.
https://prosite.expasy.org/scanprosite/scanprosite_doc.html
в разделе “Pattern syntax”

Паттерн

Преимущества: относительно просто, можно сделать «руками»

Недостаток: слишком жёстко — редкие в позиции буквы учитываются так же, как обычные, а буква, не встретившаяся в какой-нибудь позиции, оказывается в ней запрещённой

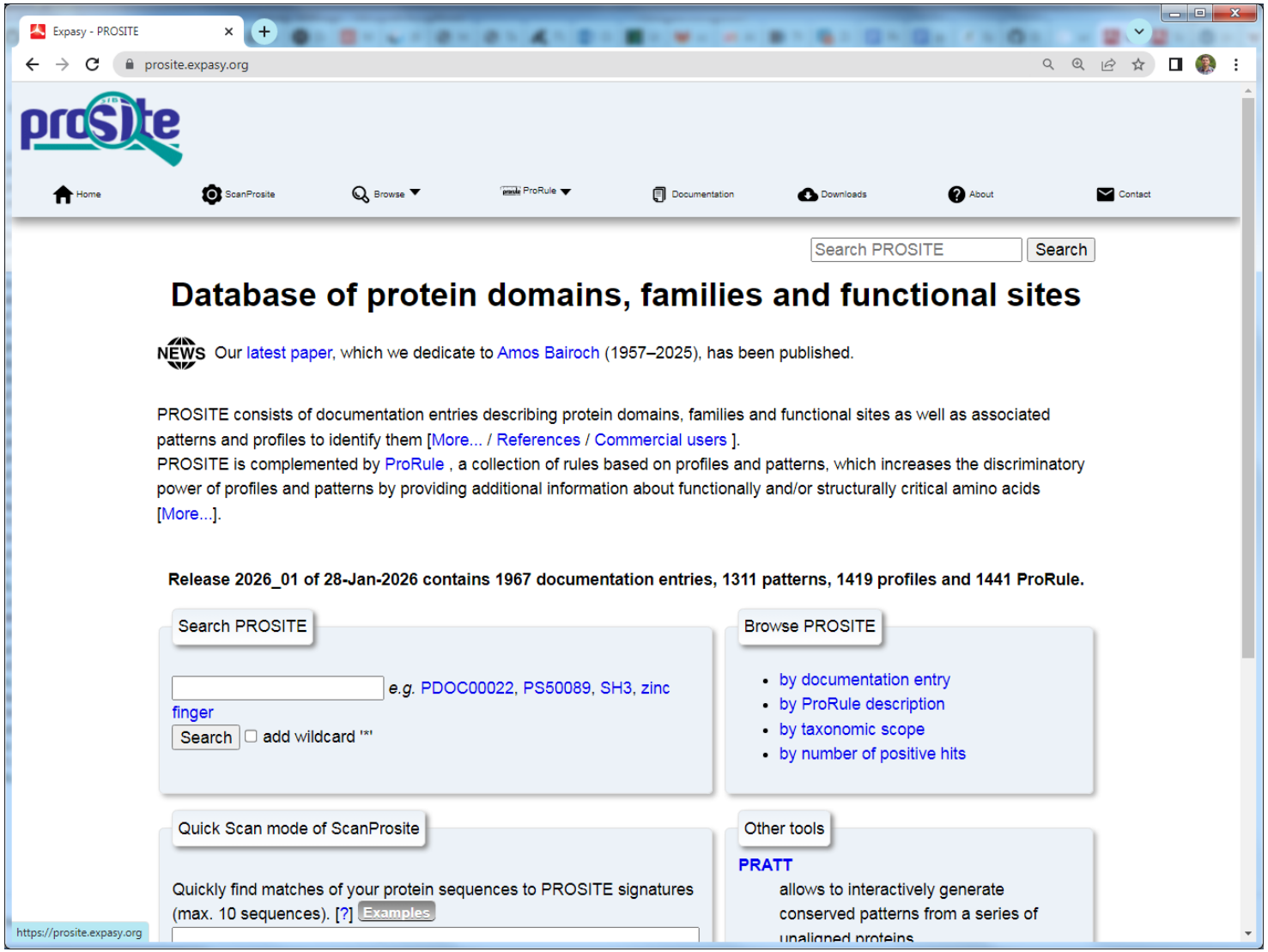
Как искать: есть специальные программы (fuzzpro и fuzznuc в EMBOSS, ScanProsite в сети)

Паттерны для мотивов в ДНК/РНК

Можно использовать тот же синтаксис, что для белков, а можно т.н. “ambiguity codes”

N — любая буква
R — A или G (puRine)
Y — C или T (pYrimidine)
S — C или G (Strong)
W — A или T (Weak)
M — A или C (aMino)
K — G или T (Keto)
B = не A
D = не C
H = не G
V = не T (и не U)

Банк Prosite



https://prosite.expasy.org/

Prosite: паттерн для аррестинов

ARRESTINS, PS00295; Arrestins signature (PATTERN)

- Consensus pattern:
  [FY]-R-Y-G-x-[DE](2)-x-[DE]-[LIVM](2)-G-[LIVM]-x-F-x-[RK]-[DEQ]-[LIVM]
- Sequences in UniProtKB/Swiss-Prot known to belong to this class: 49
  - detected by PS00295: 40 (true positives)
  - undetected by PS00295: 9 (6 false negatives and 3 'partials')
- Other sequence(s) in UniProtKB/Swiss-Prot detected by PS00295:
  NONE.

(*ср. с тем, что на слайде 9*)

Частотная матрица

letter-probability matrix

консенсус: GVHFGHQTRRWNPKM

A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	
0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	G
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	V
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	H
0.000	0.000	0.000	0.000	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	F
0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	G
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	H
0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.000	0.000	Q
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.125	0.000	0.750	0.000	0.000	0.000	T
0.125	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.000	0.125	0.500	0.000	0.000	0.000	0.000	0.000	R
0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.125	R
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	W
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	P
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.125	0.125	0.000	0.000	0.000	0.000	0.000	K
0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	M

В каждой ячейке стоит отношение $f_i(\mathrm{X}) = N_i(\mathrm{X})/N$,
где $N_i(\mathrm{X})$ — сколько раз встретилась буква X в позиции i,
а N — число последовательностей во входном выравнивании

Частотная матрица

letter-probability matrix

консенсус: GVHFGHQTRRWNPKM

A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	
0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	G
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	V
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	H
0.000	0.000	0.000	0.000	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	F
0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	G
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	H
0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.000	0.000	Q
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.125	0.000	0.750	0.000	0.000	0.000	T
0.125	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.000	0.125	0.500	0.000	0.000	0.000	0.000	0.000	R
0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.125	R
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	W
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	P
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.125	0.125	0.000	0.000	0.000	0.000	0.000	K
0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	M

В EMBOSS программа **prophecy** умеет создавать частотные матрицы, а программа **profit** — искать по ним мотив в последовательностях
(к сожалению, то и другое только для белков)

При поиске профиль «прикладывается» ко всем возможным позициям и числа, которые в матрице соответствуют последовательным буквам, складываются. Там, где сумма оказалась больше некоторого порога, диагностируется мотив.

Частотная матрица

Преимущество: гибкость — теперь типичные для позиции буквы дают больший вклад в результат, чем редкие.

Недостаток: плохо теоретически обосновано. Частота — это оценка вероятности, а вероятности в такой ситуации надо бы перемножать, а не складывать.

(*А почему бы не перемножать? Подумайте!*)

Позиционная весовая матрица

Positional weighting matrix, PWM

PWM делается из частотной матрицы

Сначала каждая частота делится на фоновую (то что получается, принято называть «отношением правдоподобия»)

Потом берём логарифм этого отношения:

$$w_i(\mathrm{X}) = \log\left(f_i(\mathrm{X})/p(\mathrm{X})\right)$$

Здесь $f_i(\mathrm{X})$ – частота буквы X в позиции i, а $p(\mathrm{X})$ – частота буквы X в “фоне” (банке или геноме)

Теперь можно искать так же, как раньше, но складываются уже логарифмы, что соответствует произведению отношений правдоподобия!

Позиционная весовая матрица

Positional weighting matrix, PWM

$$w_i(\mathrm{X}) = \log\left(f_i(\mathrm{X})/p(\mathrm{X})\right)$$

Очевидный недостаток — непонятно, как быть с нулевыми значениями частот $f_i(\mathrm{X})$ (логарифм нуля не определён).

Два выхода:

1. Заменять логарифм нуля очень отрицательным числом (–1000, например)
2. Применять **псевдоотсчёты** (pseudocounts)

Первый вариант плох своей «жёсткостью»: буква получается практически запрещённой в данной позиции (а вдруг в каком-то примере мотива она всё же случится?)

Псевдоотсчёты

Считаем заранее, что **все** буквы есть, но в небольшом количестве.

То есть в формуле для частоты вместо $f_i(\mathrm{X}) = N_i(\mathrm{X})/N$ пишем:

$$f_i(\mathrm{X}) = \big(N_i(\mathrm{X}) + \varepsilon(\mathrm{X})\big) / \big(N + \Sigma_{\mathrm{X}} \varepsilon(\mathrm{X})\big)$$

Теперь нулей не будет!

Простейший вариант (правило Лапласа) — все $\varepsilon(\mathrm{X}) = 1$

Для белков предпочтительно делать $\varepsilon(\mathrm{X})$ пропорциональными частотам X в банке

PWM с псевдоотсчётами — один из основных способов описания сигналов на ДНК, прежде всего сайтов связывания транскрипционных факторов

Поиск мотивов программой MEME

Входные данные — набор последовательностей
Выдача — мотивы, общие для этих последовательностей
Каждый мотив описан как частотной матрицей, так и PWM

Для каждого мотива посчитаны различные характеристики, в том числе *информационное содержание* (ИС, об этом в следующий раз) и E-value. Последнее надо понимать как матожидание числа находок мотивов с таким же или большим ИС в таком же числе последовательностей таких же длин.

ИС **одной** позиции мотива — это мера отличия частот букв в данной колонке от банковских частот, а ИС всего мотива равно сумме ИС по позициям.

Параметры MEME:

- Диапазон длин мотивов
- Максимальное число различных мотивов
- Для ДНК: искать ли мотивы на комплементарной цепи?
- Сколько примеров каждого мотива на входную последовательность (oops — ровно 1, zoops — 0 или 1, anr — сколько угодно)

Программа MAST

MAST позволяет искать мотивы, найденные программой MEME, в других последовательностях, используя PWM этих мотивов.

MEME и MAST установлены на kodomo