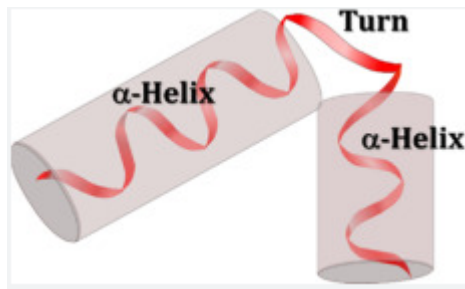


МОТИВЫ В ПОСЛЕДОВАТЕЛЬНОСТЯХ

С.А.Спирин
3 апреля 2026

Что такое «мотив»?

Структурные мотивы



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



Бета-шпилька
(beta hairpin)

Что такое «МОТИВ»?

Мотивы в последовательностях белков

VLQRRRGSIPQ



cAMP- and cGMP-dependent
protein kinase phosphorylation site

ARRB_CAEEL/66-84
ARRB_CALVI/60-78
ARRB_DROME/60-78
ARRB_DROMI/60-78
ARRC_AQUCT/60-78
ARRC_BOVIN/57-75
ARRC_HUMAN/57-75
ARRC_ICTTR/60-78
ARRC_LITPI/60-78

F	R	Y	G	r	E	D	I	D	V	L	G	L	t	F	r	K	D	L
Y	R	Y	G	r	E	E	d	E	V	M	G	V	k	F	s	K	E	L
Y	R	Y	G	r	E	E	d	E	V	M	G	V	k	F	s	K	E	L
Y	R	Y	G	r	E	E	d	E	V	M	G	V	k	F	s	K	E	L
F	R	Y	G	r	D	D	m	E	L	I	G	L	s	F	r	K	D	I
F	R	Y	G	h	D	D	I	D	V	I	G	L	t	F	r	K	D	L
F	R	Y	G	r	D	D	I	E	V	I	G	L	t	F	r	K	D	L
F	R	Y	G	r	D	D	I	D	V	I	G	L	t	F	r	K	D	L
F	R	Y	G	r	D	D	m	E	L	I	G	L	s	F	r	K	D	I

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?

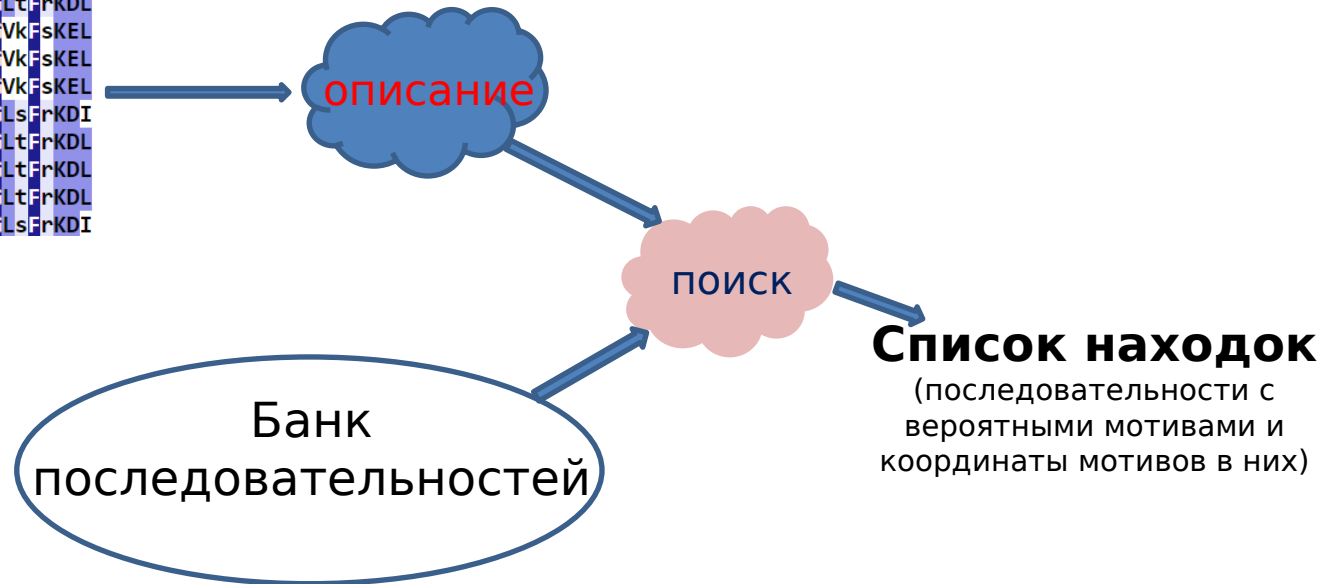
Дано выравнивание участков последовательностей, представляющих какой-либо мотив.

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

ARRB_CAEEL/66-84
ARRB_CALVI/60-78
ARRB_DROME/60-78
ARRB_DROMI/60-78
ARRC_AQUCT/60-78
ARRC_BOVIN/57-75
ARRC_HUMAN/57-75
ARRC_ICTTR/60-78
ARRC_LITPI/60-78

```
FRYGrEDIDVLGLtFrKDL  
YRYGrEEeEVMGVkFsKEL  
YRYGrEEeEVMGVkFsKEL  
YRYGrEEeEVMGVkFsKEL  
FRYGrDDmELIGLsFrKDI  
FRYGrDDIDVIGLtFrKDL  
FRYGrDDIEVIGLtFrKDL  
FRYGrDDIDVIGLtFrKDL  
FRYGrDDmELIGLsFrKDI
```

Выравнивание



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
FRYGHDDL DVIGLTFRKDL
FRYGRDDLEVIGLTFRKDL
FRYGRDDL DVIGLTFRKDL
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
FRYGHDDLVDVIGLTFRKDL
FRYGRDDLEVIGLTFRKDL
FRYGRDDLVDVIGLTFRKDL
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 — А или G (puRine)
- Y — С или Т (pYrimidine)
- S — С или G (Strong)
- W — А или Т (Weak)
- M — А или С (aMino)
- K — G или Т (Keto)
- B = не А
- D = не С
- H = не G
- V = не Т (и не U)

Банк Prosite

The screenshot shows the Prosite website interface. At the top, there is a navigation bar with links for Home, ScanProsite, Browse, ProRule, Documentation, Downloads, About, and Contact. Below the navigation bar is a search bar labeled "Search PROSITE" with a "Search" button. The main heading reads "Database of protein domains, families and functional sites". A news section indicates a latest paper by Amos Bairoch (1957–2025) has been published. The site description states that Prosite consists of documentation entries for protein domains, families, and functional sites, complemented by ProRule. A release update for 2026_01 is noted, containing 1967 documentation entries, 1311 patterns, 1419 profiles, and 1441 ProRule. The interface includes several interactive sections: "Search PROSITE" with a search input field and a "Search" button; "Browse PROSITE" with a list of browsing options; "Quick Scan mode of ScanProsite" with a description and an "Examples" link; and "Other tools" featuring the "PRATT" tool for generating conserved patterns.

Expasy - PROSITE x +
prosite.expasy.org

proSite

Home ScanProsite Browse ProRule Documentation Downloads About Contact

Search PROSITE Search

Database of protein domains, families and functional sites

NEWS Our latest paper, which we dedicate to [Amos Bairoch \(1957–2025\)](#), has been published.

PROSITE consists of documentation entries describing protein domains, families and functional sites as well as associated patterns and profiles to identify them [[More...](#) / [References](#) / [Commercial users](#)].

PROSITE is complemented by [ProRule](#), a collection of rules based on profiles and patterns, which increases the discriminatory power of profiles and patterns by providing additional information about functionally and/or structurally critical amino acids [[More...](#)].

Release 2026_01 of 28-Jan-2026 contains 1967 documentation entries, 1311 patterns, 1419 profiles and 1441 ProRule.

Search PROSITE

e.g. PDOC00022, PS50089, SH3, zinc
finger
Search add wildcard "*"

Browse PROSITE

- [by documentation entry](#)
- [by ProRule description](#)
- [by taxonomic scope](#)
- [by number of positive hits](#)

Quick Scan mode of ScanProsite

Quickly find matches of your protein sequences to PROSITE signatures (max. 10 sequences). [[?](#)] [[Examples](#)]

Other tools

PRATT
allows to interactively generate conserved patterns from a series of unaligned proteins

<https://prosite.expasy.org>

<https://prosite.expasy.org/>

Prosites: 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
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
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
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
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
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
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
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
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
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
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
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
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
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
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

В каждой ячейке стоит отношение $f_i(X) = N_i(X)/N$,
где $N_i(X)$ — 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
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
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
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
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
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
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
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
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
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
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
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
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
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
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

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:

$$w_i(X) = \log (f_i(X)/p(X))$$

Здесь $f_i(X)$ – частота буквы X в позиции i , а $p(X)$ – 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

$$w_i(X) = \log (f_i(X)/p(X))$$

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

Два выхода:

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

Первый вариант плох своей «жесткостью»: 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 $f_i(X) = N_i(X)/N$ пишем:

$$f_i(X) = (N_i(X) + \varepsilon(X)) / (N + \sum_X \varepsilon(X))$$

Теперь нулей не будет!

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

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

RWM 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