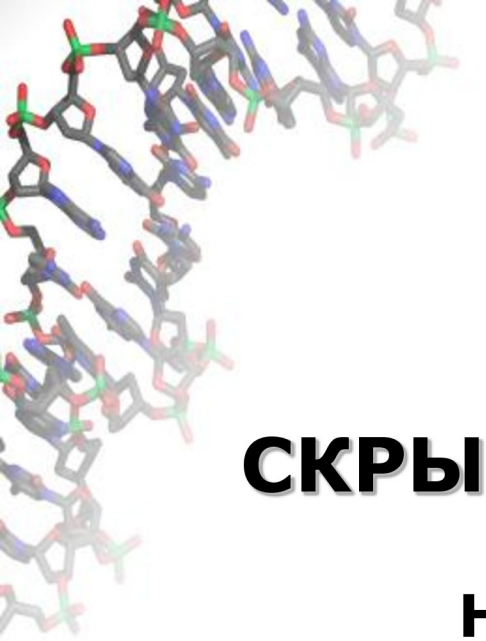
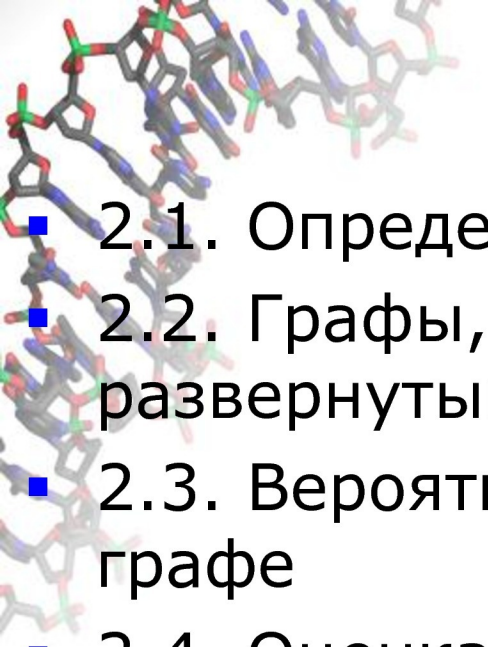


СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ

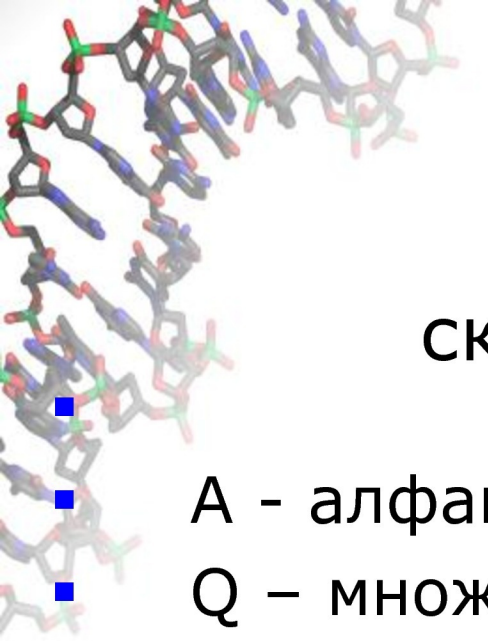
Hidden Markov Models (HMM)

Durbin R., Eddy D., Krogh A., Mitchison G. Pairwise alignment. Biological sequence analyses. Probabilistic models of proteins and nucleic acids. //Cambridge University Press, Cambridge, UK. 1998. P. 12-45.



ПЛАН (ч.1)

- 2.1. Определение
- 2.2. Графы, связанные с СММ: основной и развернутый
- 2.3. Вероятности и траектории на развернутом графе
- 2.4. Оценка параметров СММ
- 2.5. Пример: сегментация последовательностей
- 2.6. Выводы



2.1. Определение скрытой марковской модели

- A - алфавит
- Q – множество состояний; Q = {1,…, N }
- φ: ({0}ỤQ) x Q -> [0, 1]
  – вероятность перехода из i в j (i > 0);
  - вероятность старта в состоянии j (i = 0).

$$\sum_{j=1..N} \varphi(i, j) = 1$$

- σ: QxA -> [0, 1] – вероятность порождения символа

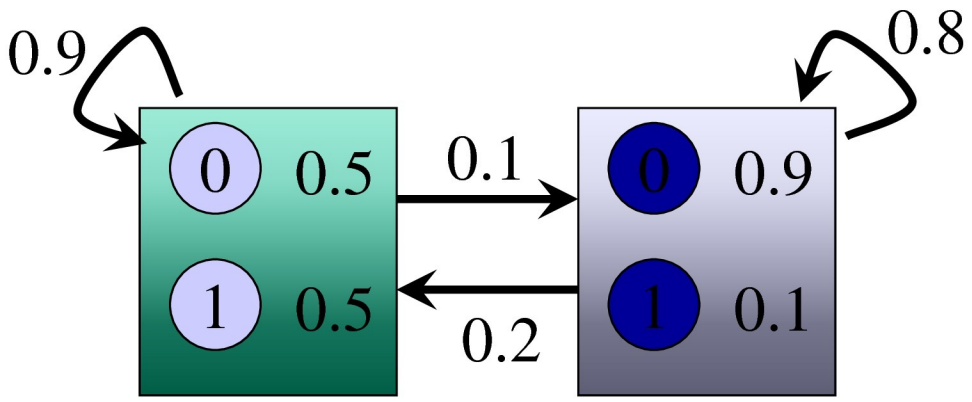
$$\sum_{a \in A} \sigma(i, a) = 1$$

ПРИМЕР (временами фальшивая монета, ВФМ)

- Прямоугольники означают ***состояния***
- Кружки означают результат бросания (***эмиссии***)
- Стрелки – возможные ***переходы*** между состояниями
- Числа около кружков – вероятности эмиссии σ_i
- числа около стрелок – вероятности переходов между состояниями φ_{ik}

•Сумма весов исходящих стрелок равна 1

•Сумма весов эмиссии в каждом состоянии равна 1



Вероятности

- Вероятность порождения слова $v = a_1 \ldots a_m$
- при условии прохождения по траектории состояний $t = \{q_0=0, q_1, \ldots, q_m\}$:

$$P(v|t) = \prod_{i=1..n} \varphi(q_{i-1}, q_i) \cdot \sigma(q_i, a_i)$$

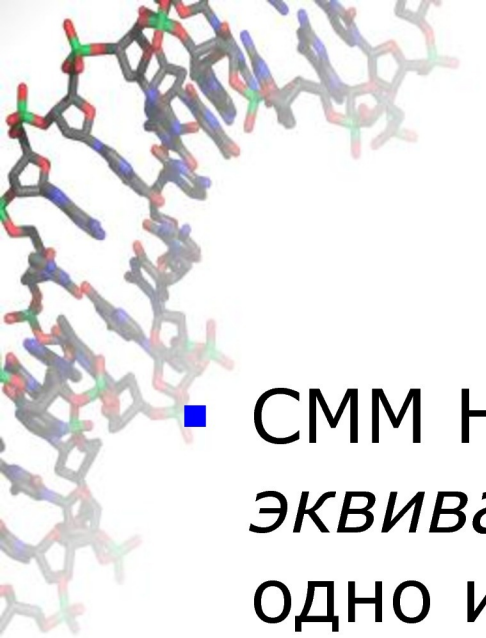
Вероятность порождения слова $v = a_1 \ldots a_m$

$$P(v) = \sum_t P(v|t)$$

- * Утв. СММ задает распределение вероятностей на словах данной длины m:

$$\sum_{v \text{ – слово длины } m} P(v) = 1$$

Упражнение 4.1. Доказать !!!



Эквивалентные СММ

- СММ H1 и H2 называются *эквивалентными*, если они задают одно и то же распределение вероятностей.

- Пример – «раздвоение» состояний

2.2. Графы, связанные с СММ

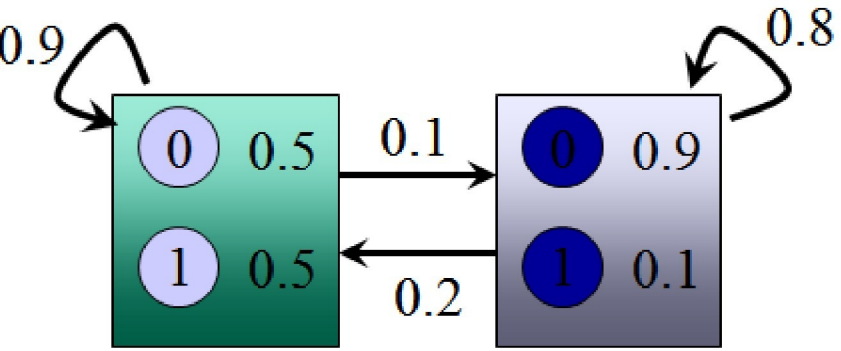
- Пусть дана СММ $H = \langle A, Q, \varphi, \sigma \rangle$.

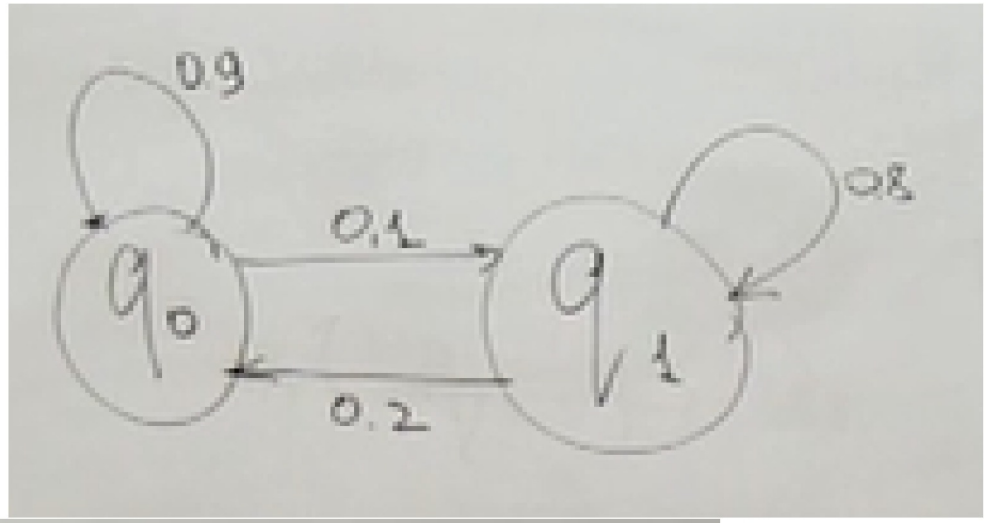
- **Основной граф**.
- *Множество вершин* $Q = \{0, 1, \ldots, N\}$.
- *Множество ребер* $E = \{(i, j, a)\}$ - для каждой пары вершин есть $|A|$ ребер, ведущих из i в j; каждое из ребер помечено своей буквой.
- *Вероятности:* Каждому ребру (i, j, a) приписана вероятность

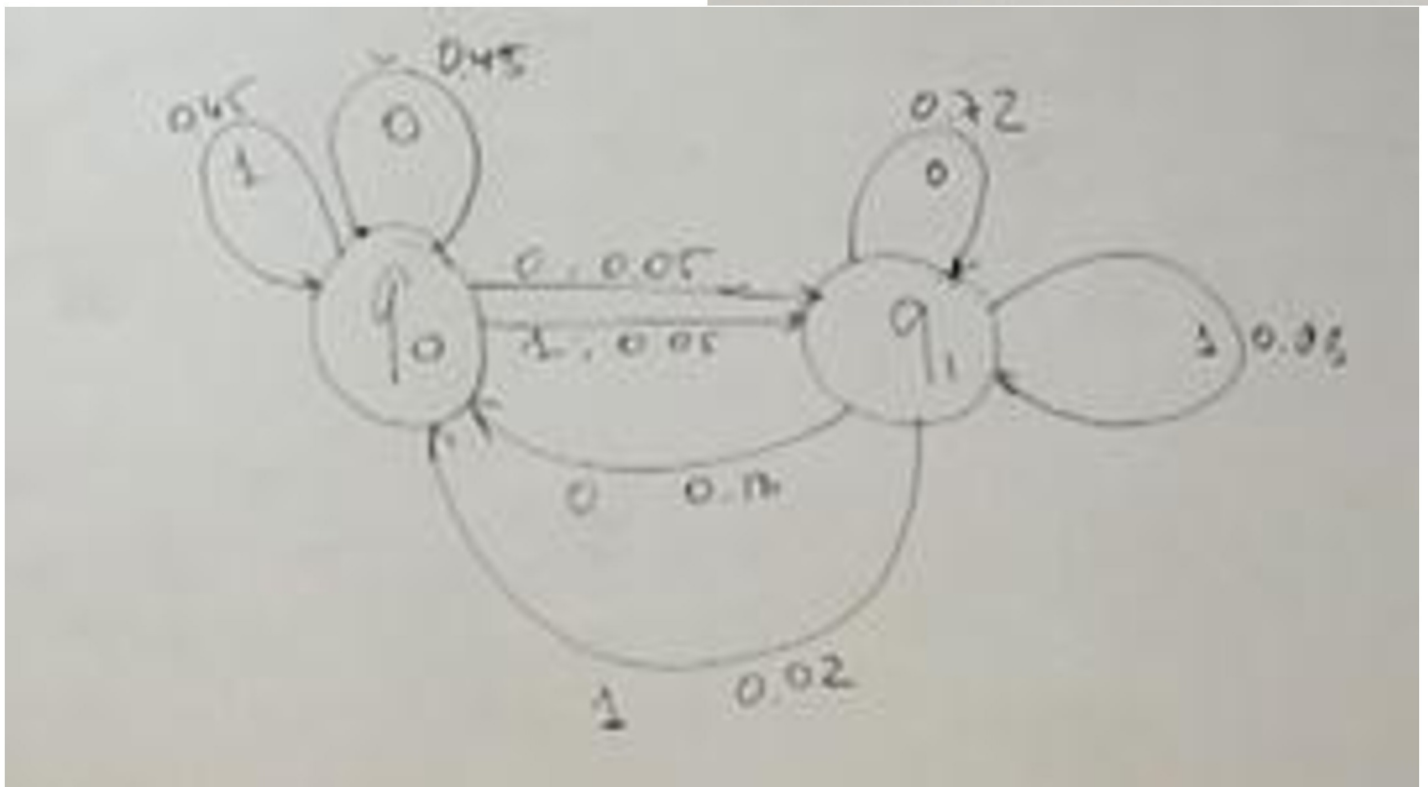
$$p(i, j, a) = \varphi(i, j) \cdot \sigma(j, a)$$

  * Ребра, имеющие вероятность 0, можно опускать
- ***Основной граф, как правило, содержит циклы.***

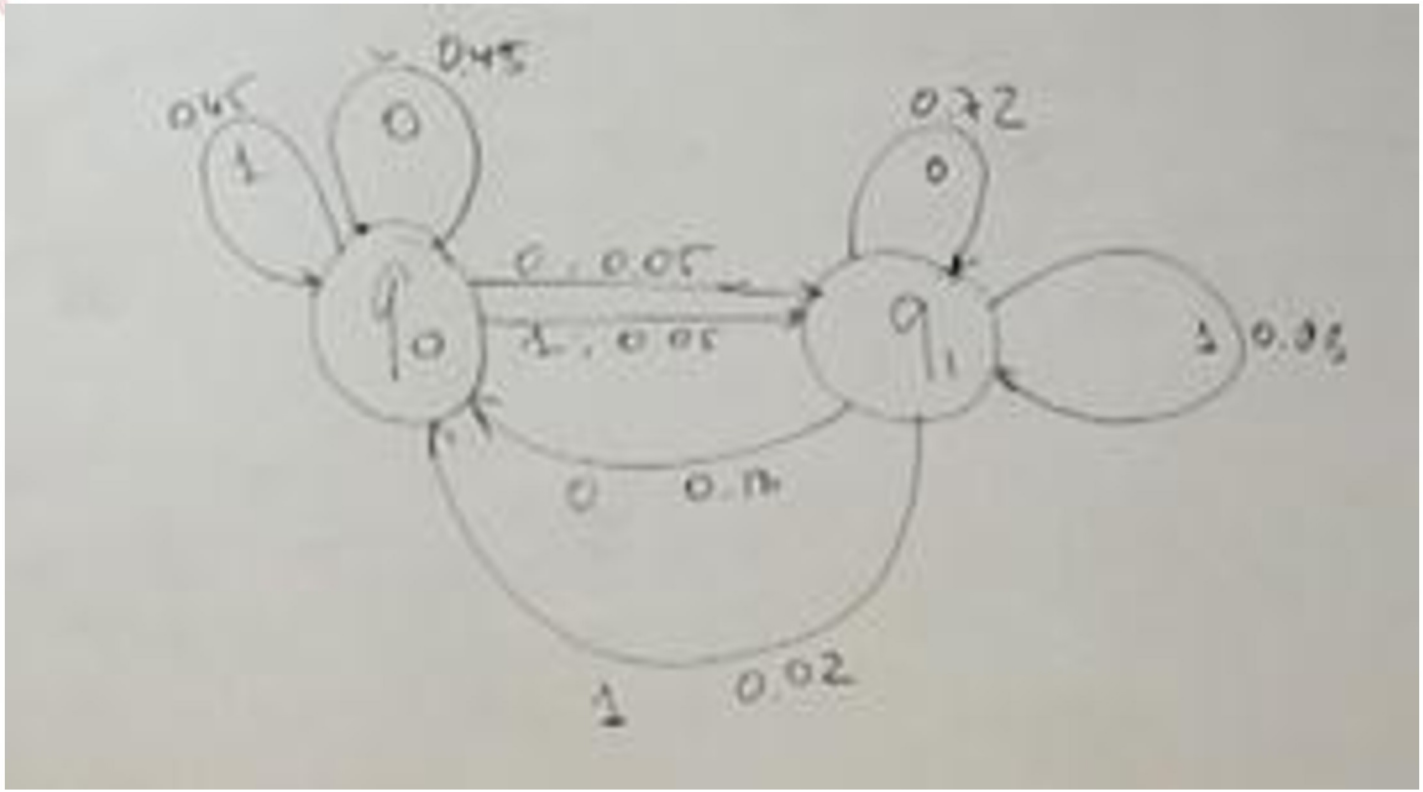
Основной граф, как правило, содержит циклы.

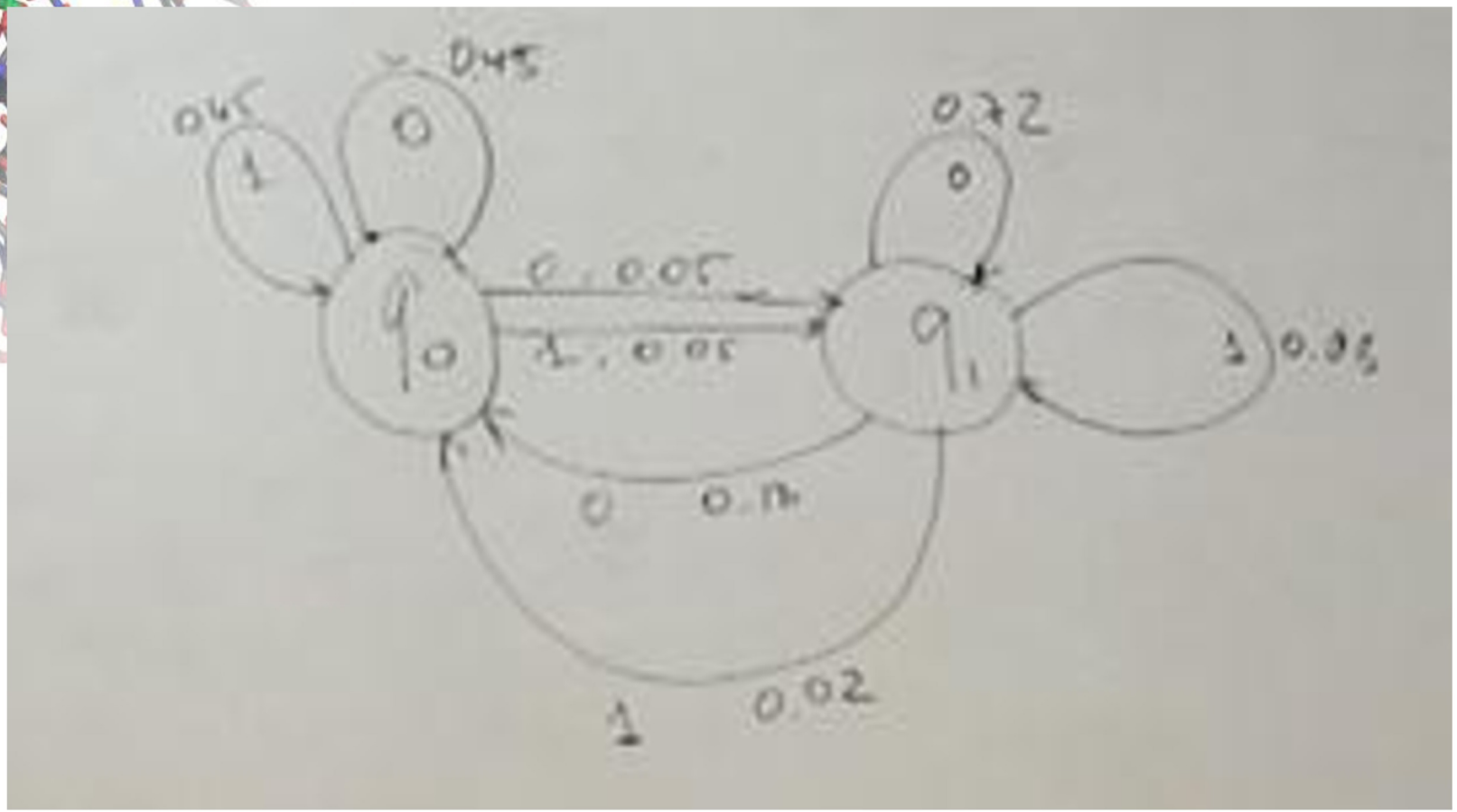






Интерпретации:
1) порождение последовательностей
2) приписывание вероятностей последовательностям





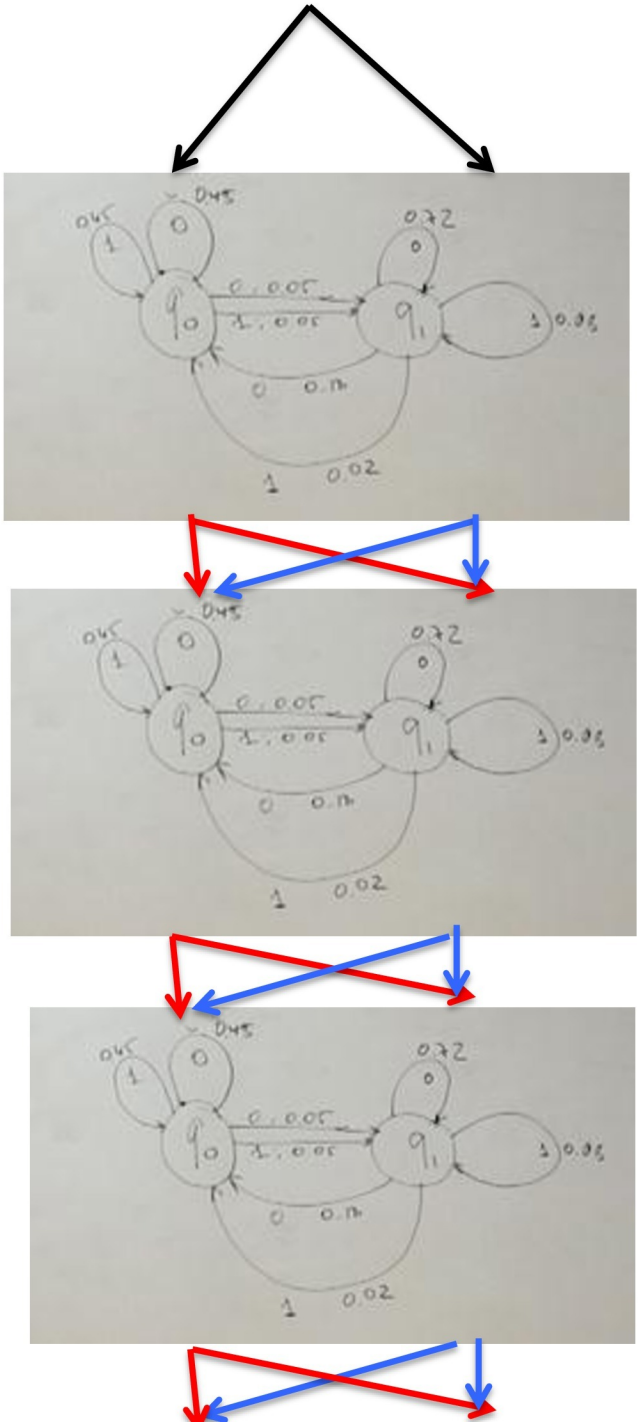
- Вероятность порождения слова v = a**1**…a**m**
- при условии прохождения по траектории состояний t = {q**0**=0, q**1** , …, q**m**}:

$$P(t) = \mathbf{\Pi i=1..\ n}\ \varphi(q\mathbf{i-1},\ q\mathbf{i})\cdot\sigma\ (q\mathbf{i},\ a\mathbf{i})$$

Вероятность порождения слова v = a**1**…a**m**

$$P(v) = \Sigma\ \{P(t) \mid word(t) = v\}$$

1



Развернутый граф – циклов НЕТ.

2.3. Задачи на развернутом графе *Вероятности.*

- 1. Найти вероятность данного слова $v = a_1 \ldots a_m$.

  Решение. Рассмотрим граф Tr(H, m, v) – редукцию развернутого графа, в котором для каждой тройки (i, j, k) оставлено только ребро (i, j, a_k, k). Решаем задачу Больцмана для Tr(H, m, v)

  Время ~ $N^2 \cdot m$

- 2. Дан набор V слов длины m. Найти суммарную вероятность слов из набора V.

  Время ~ $N^2 \cdot m \cdot |V|$

3.* Набор слов V длины m задан конечным автоматом с ***r*** состояниями. Найти суммарную вероятность слов из набора V.

Время - ?

Вероятности

- 3.* Набор слов V длины m задан конечным автоматом с ***r*** состояниями. Найти суммарную вероятность слов из набора V.

  Время - ?

  Пример. Набор V задается профилем (позиционно-зависимой системой весов).

Напомним: Для любого конечного множества слов можно построить автомат, допускающий это множество и содержащий ~ L состояний, где L – суммарная длина слов (автомат Ахо-Корасик).

Задачи на развернутом графе.
Разметка

- Дано слово v = $a_1 \ldots a_m$.
- 4. Найти траекторию t = {q_0, q_1, …, q_m} для которой вероятность P(v|t) максимальна.
  - Решение. Задача Беллмана на графе Tr(H, m, v). **[Viterby]**

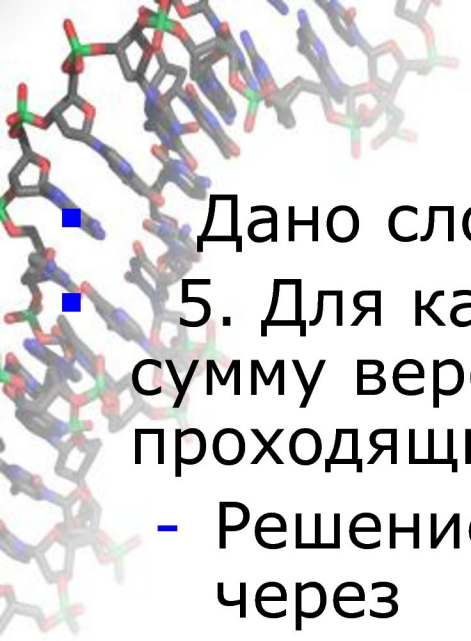
    Обозначение: **TrMaxGlob(v).**

    TrMaxGlob(v) = argmax{t| P(v, t)}

    Обозначение: T(v, k) – вершина k-го слоя, через которую проходит траектория TrMaxGlob(v)

    TrMaxGlob(v) = = {0, T(v, 1), …, T(v, m)}

    **Варианты: все наилучшие траектории, все траектории, которые *ненамного* отличаются от лучшей.



Задачи на развернутом графе
Разметка

- Дано слово v = $a_1 \ldots a_m$.
- 5. Для каждой вершины (i, k) найти P(i,k | v) – сумму вероятностей P(t) по всем v-траекториям, проходящим через вершину (i,k).
  - Решение. Задача о вероятности прохождении через вершину для графа вероятностей

    Tr(H, m, v) – вычисление сумм Больцмана для всех вершин [*forward-backward*]

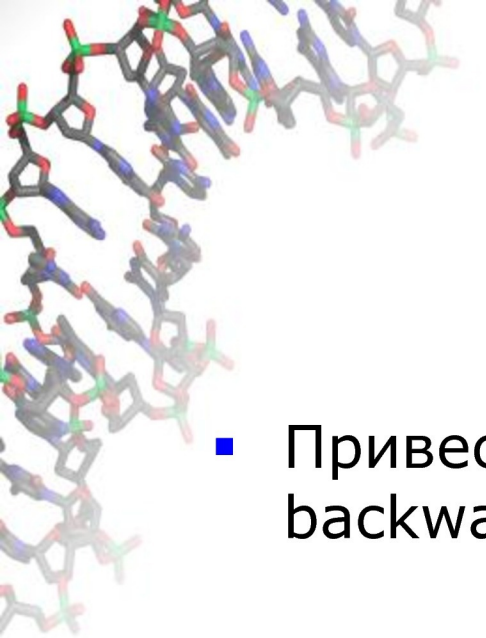
Термин: *апостериорная* вероятность вершины (i, k).

Обозначение: B(v, k) = argmax{i|P(i,k | v)}.

(вершина, имеющая максимальную ап.в. в своем слое).

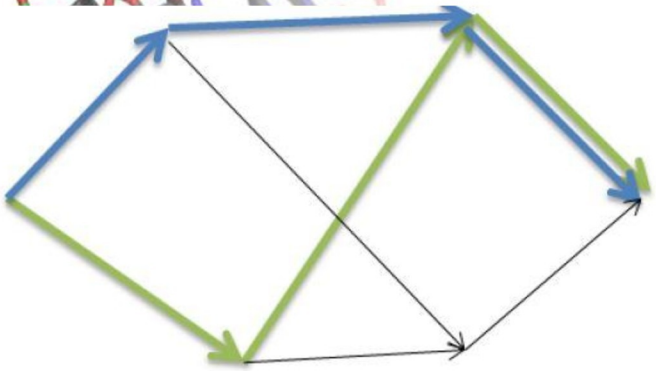
Обозначение: **TrMaxLoc(v)** =

= {0, B(v, 1), …, B(v, m)}



Немного поиграем

- Привести пример, когда траектория forward-backward имеет нулевую вероятность
- Привести пример, когда траектории Viterby и forward-backward не имеют общих вершин.



Что максимизирует траектория forward-backward ?

Вероятность $Prob(q)$ траектории $q = \{q_0, q_1, \ldots, q_T\}$ определяется естественным образом — как произведение вероятностей по траектории.

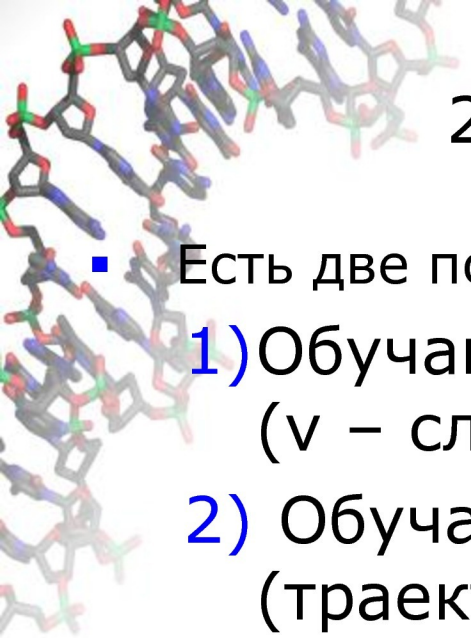
Пусть $q = \{q_0, q_1, \ldots, q_T\}$ и $r = \{r_0, r_1, ..., r_T\}$ — две траектории. Для простоты считаем, что все траектории имеют одно и то же начальное и конечное состояния.

Пусть $id(q, r)$ — количество позиций x таких, что $q_x = r_x$, $x = 1, \ldots, T-1$. Например, для зелёной и синей траекторий на рисунке количество таких позиций равно 1.

С каждой фиксированной траекторией R можно связать случайную величину $id(q, R)$.

Обозначим $E(R) = \sum_q Prob(q) \cdot id(q, R)$.

Говоря неформально, $E(R)$ — это среднее число общих вершин у случайной траектории и заданной траектории R.



2.4. Оценка параметров СММ

- Есть две постановки задачи.

1) Обучающая выборка - множество пар <v, t> (v – слово, t – траектория).

2) Обучающая выборка - множество слов (траектории неизвестны).

В обоих случаях предполагается известными сами модели,

т.е. конечные автоматы описаны, но неизвестны вероятности переходов и вероятности эмиссии букв.

* В крайнем случае, можно считать, что все переходы допустимы. Это увеличивает множество параметров, а значит и требует увеличения размеров обучающей выборки.

Оценка параметров СММ при известных траекториях

- Используется техника оценки параметров ***методом наибольшего правдоподобия.***
- Пусть дан набор независимых наблюдений
  $\{x^1, \ldots, x^n\}$; *все* x^i – независимые наблюдения.
  Пусть вероятность наблюдения ***x*** *задаются формулой* $Prob(x) = P(x|\theta)$, *зависящей от параметра* θ.

  Наша цель:

  Найти значение параметра θ, которое «наиболее соответствует» наблюдениям $\{x^1, \ldots, x^n\}$;

– .

Метод наибольшего правдоподобия

- Пусть дан набор наблюдений $\{x^1, \ldots, x^n\}$;

  все x^i – независимые наблюдения.

  Пусть вероятность наблюдения ***x*** *задаются формулой* $Prob(x) = P(x|\theta)$, *зависящей от параметра* θ.

  Наша цель:

  Найти значение параметра θ, которое «наиболее соответствует» наблюдениям $\{x^1, \ldots, x^n\}$.

- ИДЕЯ:
- $\boldsymbol{\theta^* = \operatorname{argmax}_\theta P(x^1 \ldots x^n \mid \theta) = \operatorname{argmax}_\theta \{\Sigma_j \log P(x^j \mid \theta)\}}$

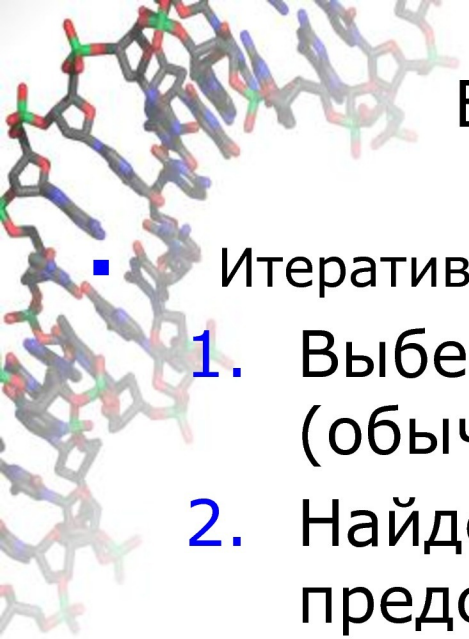
- **Неформально: ищем такой набор параметров, при котором обучающая выборка имеет максимальную вероятность.**

- Можно показать, что при большом количестве наблюдений справедливы оценки

  $a_{kl} = A_{kl} / \Sigma_{l'} A_{kl'}$; $e_k(b) = E_k(b) / \Sigma_{b'} E_k(b')$;

  - A_{kl} – наблюденное количество переходов из состояния k в состояние l;
  - $E_k(b)$ – количество символов b, порожденных в состоянии k.

  * При малых размерах выборки используют технику псевдоотсчетов (= регуляризация), добавляя к наблюденным значениям поправку, связанную с априорной гипотезой о вероятностях.



Если параметры неизвестны…

- Итеративный алгоритм Баума-Велча.
  1. Выберем некоторые наборы параметров СММ (обычно они генерируются случайно).
  2. Найдем для них оптимальные пути во всех представленных примерах
  3. По найденным оптимальным путям определим новые параметры
  4. Перейдем к шагу 2.
- Показано, что алгоритм сходится (отношение правдоподобия растет на каждой итерации)
- Есть опасность нахождения локального, а не глобального экстремума.

Перевычисление параметров по Бауму-Велчу-1

Перелистнем…

- Пусть дана СММ H = <A, Q, φ, σ > с N состояниями и набор слов v_1,…, v_R . Положим:
-
- F(s, t) – это *оценка* ***количества переходов*** из состояния s в состояние t (исходя из выборки слов V_1, …, V_R.при заданной СММ H = <A, Q, φ, σ >);

- E(s, b) – *оценка* ***количества эмиссий*** символа b, когда СММ находилась в состоянии s (исходя из выборки слов V_1, …, V_R.при заданной СММ H = <A, Q, φ, σ >).

Перевычисление параметров по Бауму-Велчу-2

Перелистнем…

- F(s, t) – это *оценка* ***количества переходов*** из состояния s в состояние t (исходя из выборки слов v1, …, v**R**.при заданной СММ H = <A, Q**,** φ, σ >);
- E(s, b) – *оценка* ***количества эмиссий*** символа b, когда СММ находилась в состоянии s (исходя из выборки слов v1, …, vR.при заданной СММ H = <A, Q**,** φ, σ >).

- **Сведение к оценкам по одному слову:**

$$F(s,t) = \sum_{j=1,\dots,R} F1(s, t, v_j);$$

$$E(s,b) = \sum_{j=1,\dots,R} E1(s, b, v_j).$$

Перевычисление параметров по Бауму-Велчу-3

Перелистнем...

- **Оценки по одному слову:**

$$F1(s,t;v) = (1/Pr(v)) \bullet \bullet \left(\sum_{k=0,\ldots,m-1} Pr(v;\pi_k=s, \pi_{k+1}=t)\right)$$

$$E1(s,t;v) = (1/Pr(v)) \bullet \left(\sum_{k:\ x_k=b} Pr(v;\pi_k=s)\right)$$

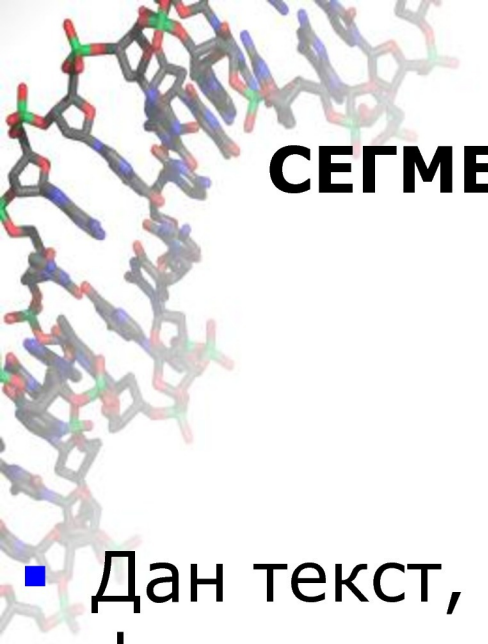
Перевычисление параметров по Бауму-Велчу

Перелистнем и это!

- **Вероятность**

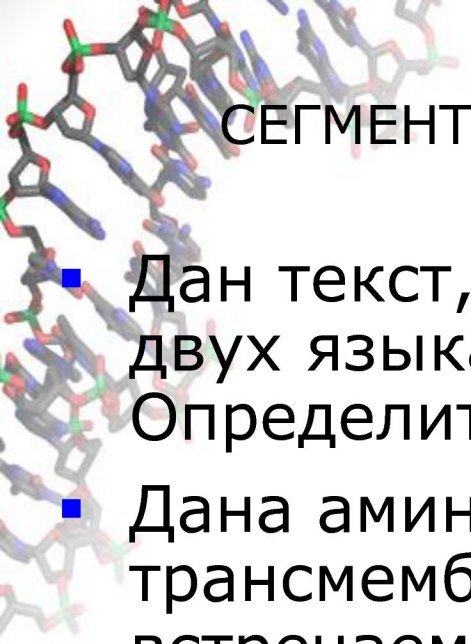
$$\Pi_{j=1,..,\,R} \Pr(vj)$$

не убывает !



2.5. Пример: СЕГМЕНТАЦИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ
ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ

- Дан текст, в котором перемежаются фрагменты с различными свойствами. Определить границы фрагментов.



СЕГМЕНТАЦИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ. Примеры.

- Дан текст, в котором перемежаются фрагменты на двух языках с одинаковым алфавитом. Определить границы фрагментов.
- Дана аминокислотная последовательность трансмембранного белка. Известно, что частоты встречаемости аминокислот в трансмембранных и в растворимых частях белка различаются (аналог разных монет). Определить по последовательности где находятся трансмембранные участки.
- Дана геномная последовательность. Статистические свойства кодирующих областей отличаются от свойств некодирующих областей. Найти кодирующие области.
- • • •
- • • •

ПОДХОД к РЕШЕНИЮ ЗАДАЧИ

- Дана СММ H = {A, Q, φ, σ}, причем

$$Q = Q_1 + Q_2 + \ldots + Q_k ,$$

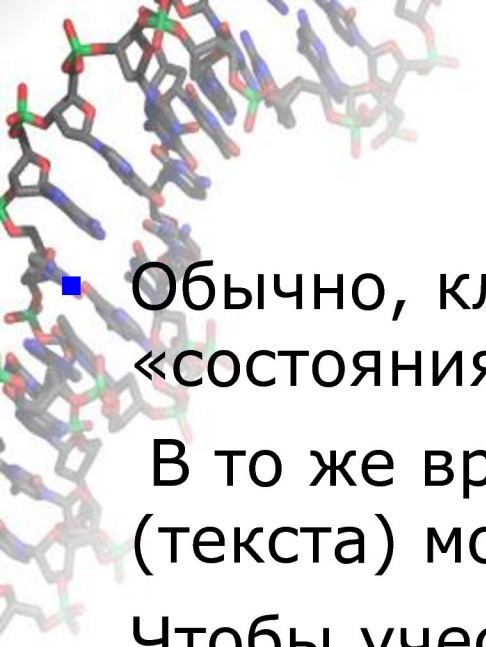
причем

(*) вероятности перехода между состояниями, лежащими в различных классах Q_j, существенно меньше, чем вероятности переходов внутри одного класса.

Пусть дана символьная последовательность v, предположительно описываемая данной СММ.

Восстановим по v траекторию состояний (одним из двух способов – как Trmaxglob(v) или Trmaxloc(v)).

В этих траекториях состояния из одного класса будут идти блоками (см. (*)). Это и будет искомая сегментация (разбиение на блоки).



УТОЧНЕНИЕ ГРАНИЦ

- Обычно, классы Q_1, Q_2,…,Q_k описывают основные «состояния» процесса (текста).

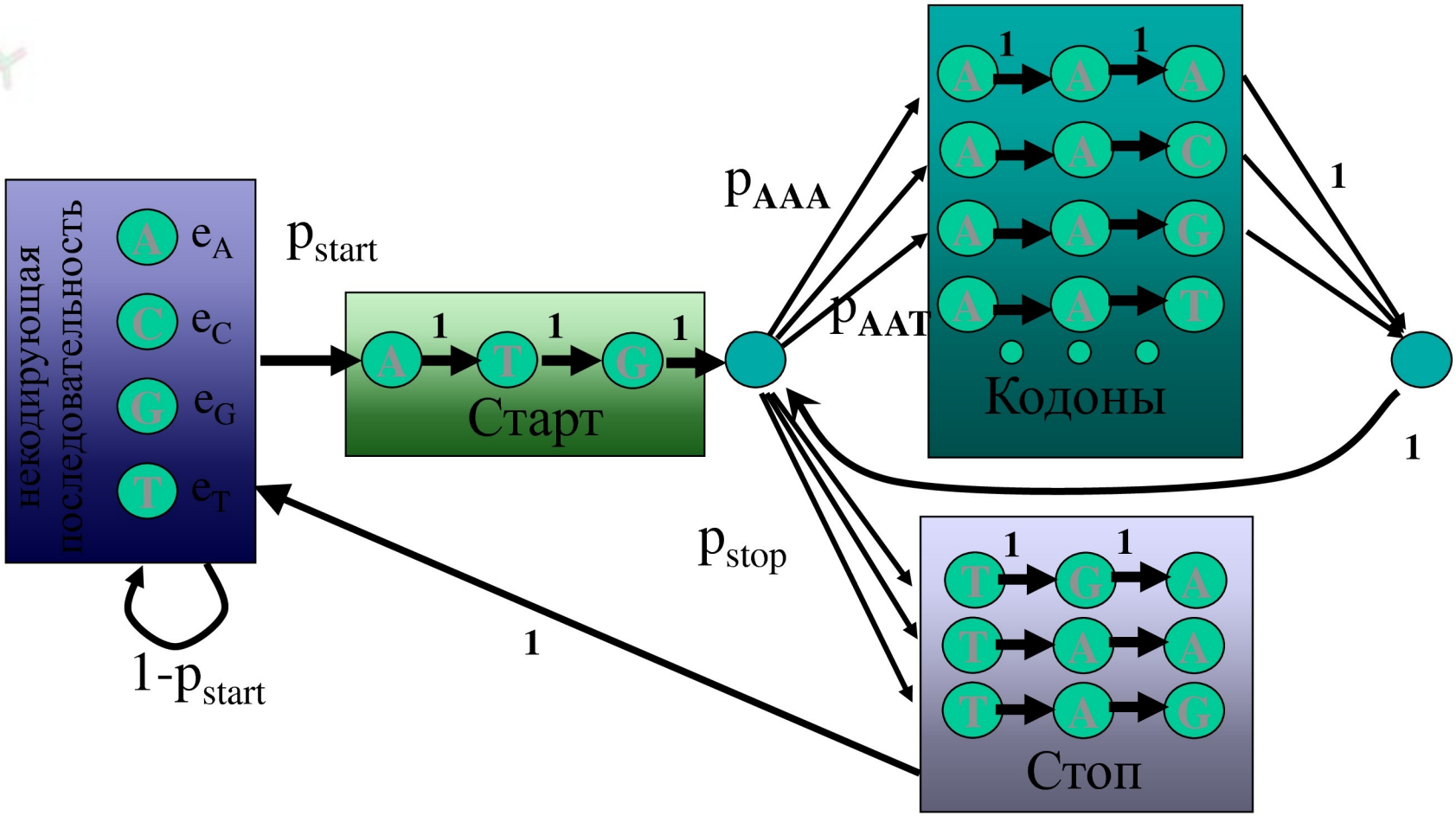
  В то же время, вблизи границ свойства процесса (текста) могут меняться.

  Чтобы учесть это, приходится вводить новые состояния модели, что увеличивает ее сложность.

  Альтернатива: грубо предсказать положение границ с помощью основной модели, а потом уточнять их локально (например, перебором вариантов максимизируя некоторый критерий).

Пример сегментации.
Предсказание кодирующих областей в прокариотах

- Реальная схема HMM для поиска кодирующих областей сложнее. Например, она учитывает неравномерность следования кодонов друг за другом.



Оценка качества сегментации
Посимвольное сравнение

- Дано: тестовая выборка троек вида
  - слово,
  - истинная траектория состояний,
  - восстановленная траектория состояний.

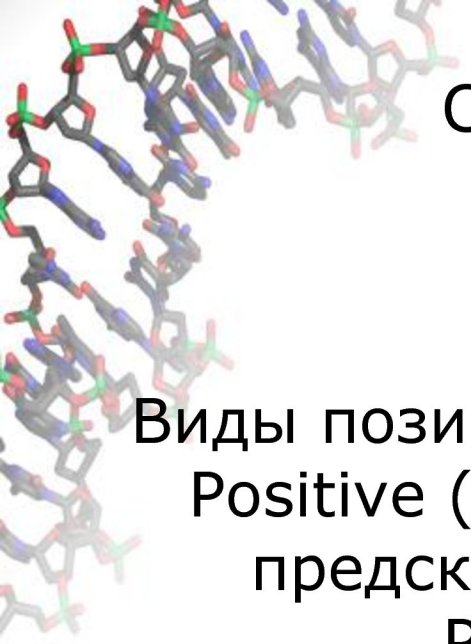
Обозначения:

L – *суммарная длина слов*

Ncorr – количество позиций, в которых истинное и восстановленное состояния совпадают

Однородные классы:

$$\mathbf{Q = Ncorr / L}$$



Оценка качества сегментации
Посимвольное сравнение
***Неоднородные классы*:**

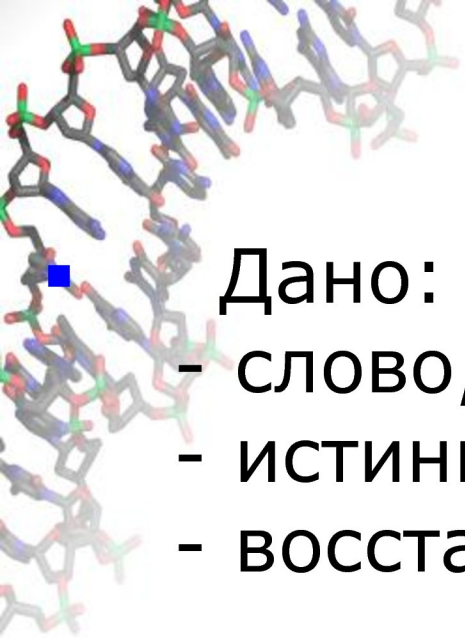
Виды позиций:

Positive (P)– редкий, Negative (N)– частый

предсказано	реально	обозн.кол-ва
P	P	TP
P	N	FP
N	P	FN
N	N	TN

Точность: $Acc(v) = Num(TP)/Num(TP+FN)$

Достоверность: $Conf(v)= Num(TP)/Num(TP+FP)$

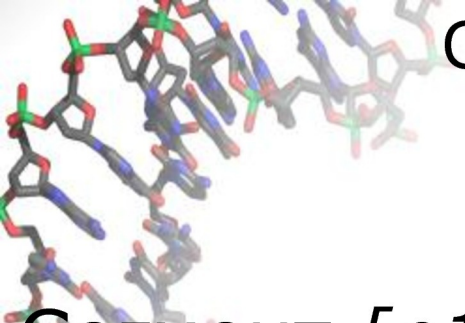


Оценка качества сегментации
Посегментное сравнение

- Дано: тестовая выборка троек вида
  - слово,
  - истинная траектория состояний,
  - восстановленная траектория состояний.

- ***Каждая траектория разбита на сегменты***

- ***Каждому сегменту соответствует класс состояний***



Оценка качества сегментации
Посегментное сравнение
Сопоставление сегментов

Сегмент *[p1, p2]* истинного разбиения ***соответствует*** сегменту *[q1, q2]* восстановленного разбиения, если длина общей части этих сегментов составляет не менее *r%* каждого из сегментов (*r >50%)* и им сопоставлен один и тот же класс состояний.

Утв. 1. Если сегмент *[p1, p2]* соответствует сегменту *[q1, q2],* то и сегмент *[q1, q2]* соответствует сегменту *[p1, p2].*

Утв. 2. Каждый сегмент одного разбиения соответствует не более, чем одному сегменту другого разбиения.

Оценка качества сегментации
Посегментное сравнение

- Обозначения:

NSegmTrue – *общее количество сегментов в истинном разбиении*

NSegmAlg – *общее количество сегментов в алгоритмическом разбиении*

Ncorr – количество позиций, в которых истинное и восстановленное состояния совпадают

- ***Однородные классы*:**

Точность: Acc(v) = *NCorr/NSegmTrue*

Достоверность: Conf(v) = *NCorr/NSegmAlg*

Оценка качества сегментации
Посегментное сравнение

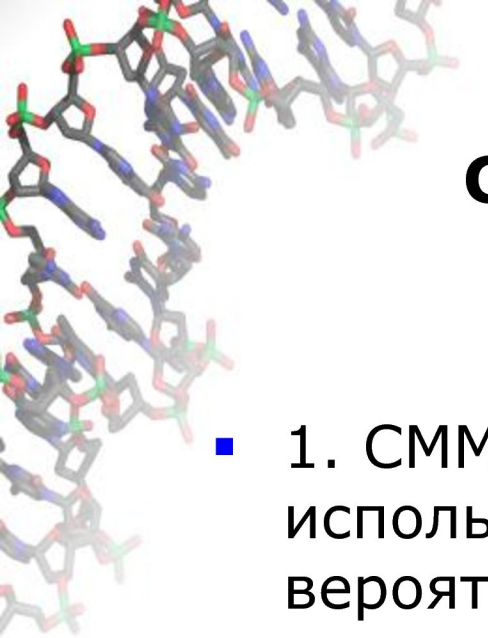
- ***Неоднородные классы.***

Positive – редкий, Negative – частый (P, N).

предсказано	реально	обозн.кол-ва
P соответствует	P	TP

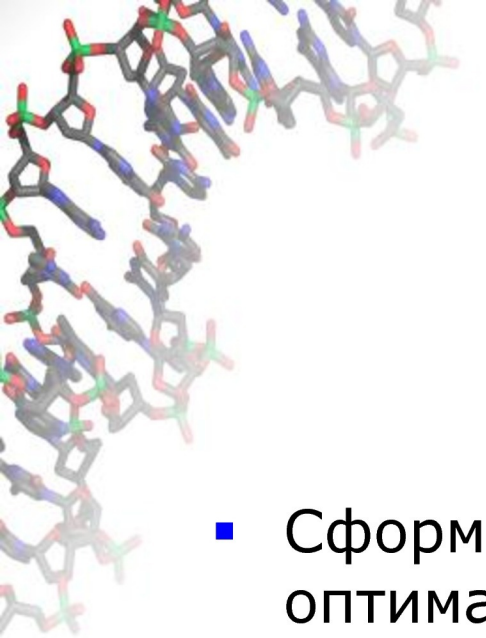
Точность: Acc(v) = *TP/NSegmTrue*

Достоверность: Conf(v) = TP*/NSegmAlg*



Скрытые марковские модели
Выводы

- 1. СММ – удобный и наиболее общий из используемых сейчас способов задавать вероятности.
- Алгоритмически – это задачи ДП на ациклическом графе (графе траекторий)
- Польза от СММ – точка зрения, которая подсказывает, как подбирать параметры



Задача++ .

- Сформулировать задачу построения оптимального глобального выравнивания на языке скрытых Марковских моделей (СММ).