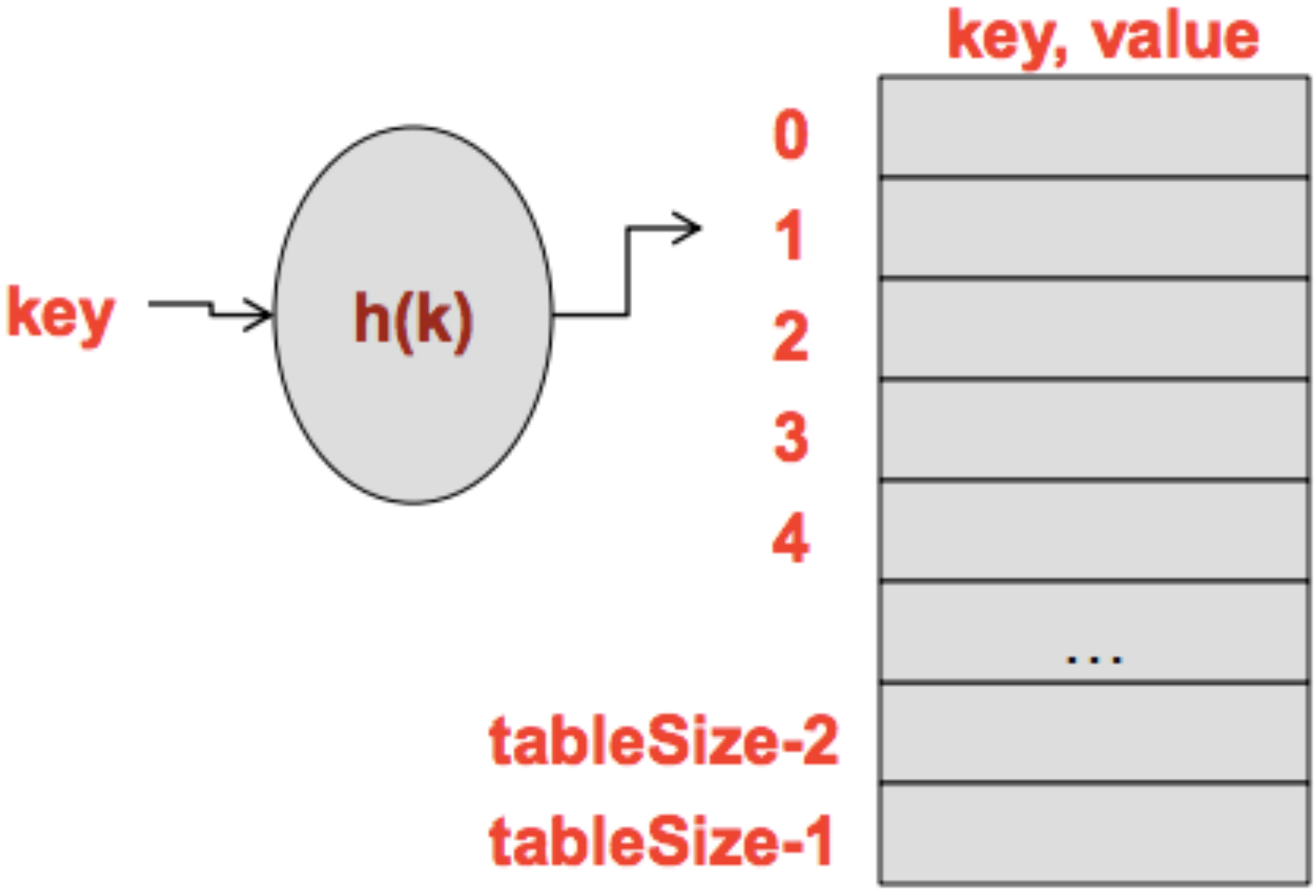
BLAST:
Basic Local Alignment Search Tool

Altschul et al. J. Mol Bio. 1990.

Hashing

A hash function maps a key to a value



Hash table

- Hash table is a data structure: a way to store key-value pairs, and a way to retrieve them

- Based on the idea of a hash function. This maps a key or an object (e.g., a string, or a more complex record) to an integer, the “address”

- The value of the key is then stored at that address in memory

Hashing: an example

- Key: (AAACGTAT, 1234321)
  - i.e., a 8 bp-string and its location in genome
- We want to store many such strings and their locations
  - and later retrieve all locations of a particular string really quickly
- Hash function h(AAACGTAT) = 435

Hashing: an example

- Let's assume that there are 4^8 = 64K memory locations available.

- The first time we see (AAACGTAT, *), we store it at address h(AAACGTAT) = 435.

- The next time we see (AAACGTAT, *), we compute h(AAACGTAT), go to 435, find it already occupied. A collision!

How to handle collisions

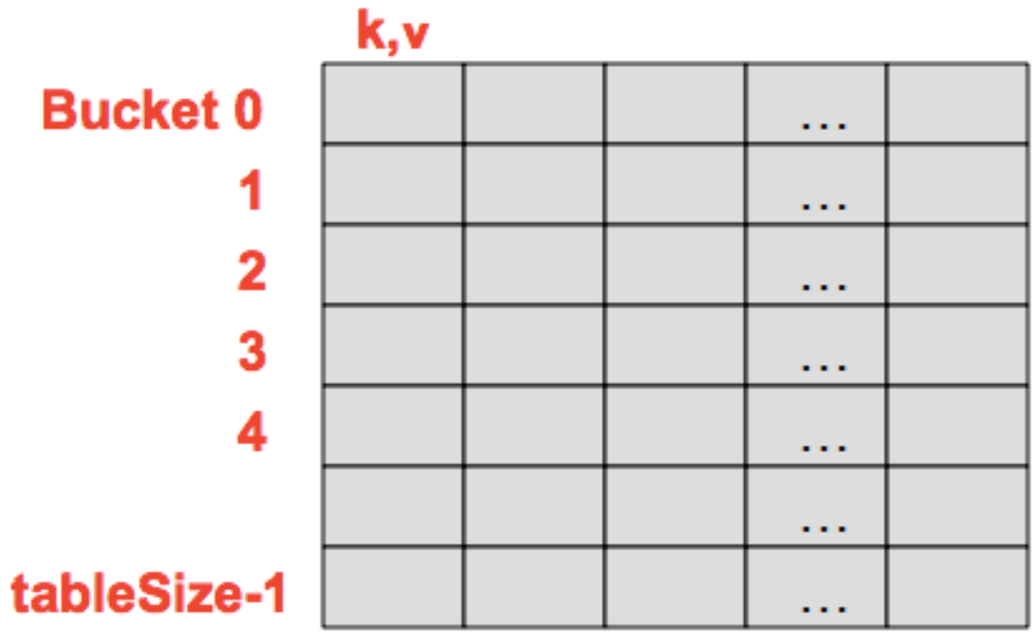
- Buckets: Address 435 can store multiple keys/ objects (e.g., as a linked list)

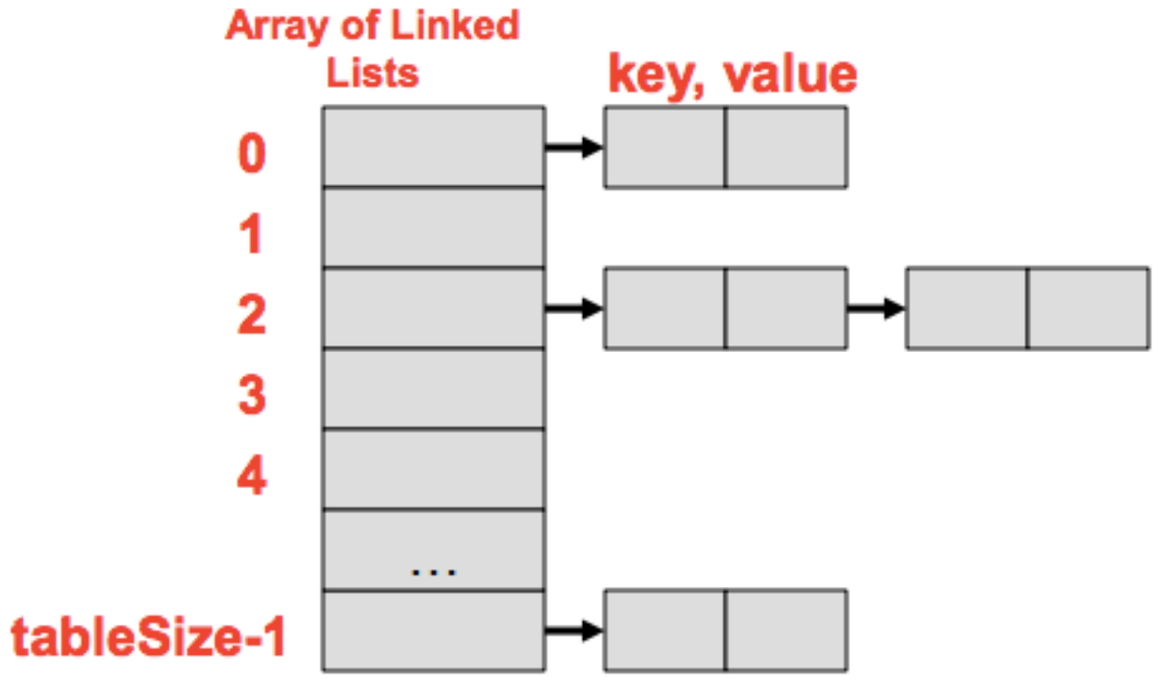
- Linear probing: If an address is occupied, store the key/object in next available location

- Multiple hashing: have an army of hash functions. If the first one (“h”) led to a collision, try another hash function (“h2”)

Bucketing and Chaining

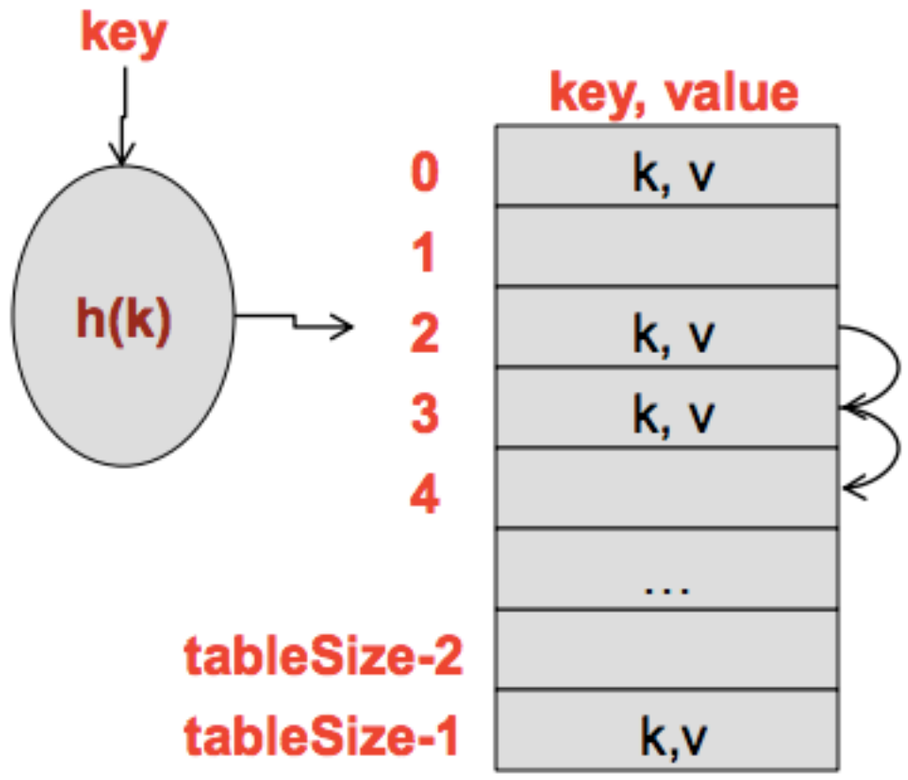
- Rather than searching for a free entry, make each entry in the table an ARRAY (bucket) or LINKED LIST (chain) of items/entries
- Buckets
  - How big should you make each array?
  - Too much wasted space
- Chaining
  - Each entry is a linked List





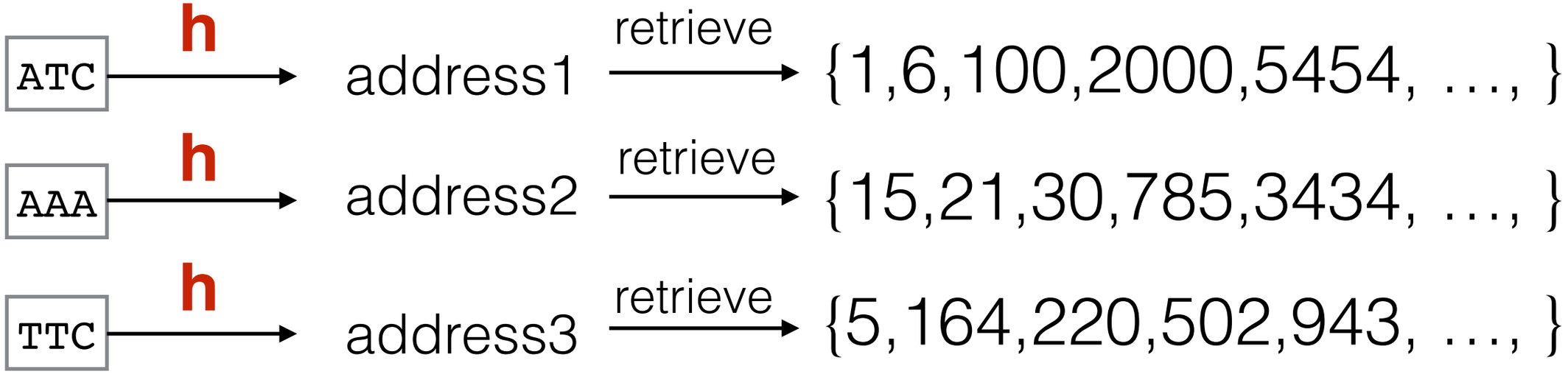
Open addressing and linear probing

- Open addressing means an item with key, k, may not be located at h(k)
- Assume, location 2 is occupied with another item
- If a new item hashes to location 2, we need to find another location to store it
- Linear Probing
  - Just move on to location h(k)+1, h(k)+2, h(k)+3,…



Preprocessing and hash

Preprocessing:
store exact matches of all short patterns on the text by a hash table



BLAST: finding maximal segment pairs

- Given two sequences of same length, the similarity score of their alignment (without gaps) is the sum of similarity values for each pair of aligned residues

- Maximal segment pair (MSP): Highest scoring pair of identical length segments from the two sequences being compared (“query” and “subject”)

- The similarity score of an MSP is called the MSP score

- BLAST heuristically **aims** to find them

Maximal segment pairs and High scoring pairs

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: SPAC869.02c [Schizosaccharomyces pombe]

 Score = 33.1 bits (74),  Expect = 0.24
 Identities = 27/95 (28%), Positives = 50/95 (52%), Gaps = 10/95 (10%)

Query   30   ERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAH   89
             ++M  ++P      P+F+ +H  +     + +A AL N   ++DD+  +LSA  D
Sbjct   59   QKMLGNYPEV---LPYFNKAHQISL--SQPRILAFALLNYAKNIDDL-TSLSAFMDQIVV  112

Query   90   K---LRVDPVNFKLLSHCLLVTLAAHLPAEF-TPA  120
             K    L++    ++ ++ HCLL T+   LP++  TPA
Sbjct   113  KHVGLQIKAEHYPIVGHCLLSTMQELLPSDVATPA  147
```

- Goal: report database sequences that have MSP score above some threshold S.
  - Thus, sequences with at least one locally maximal segment pair that scores above S.

High scoring pairs (or local maximal segment pairs)

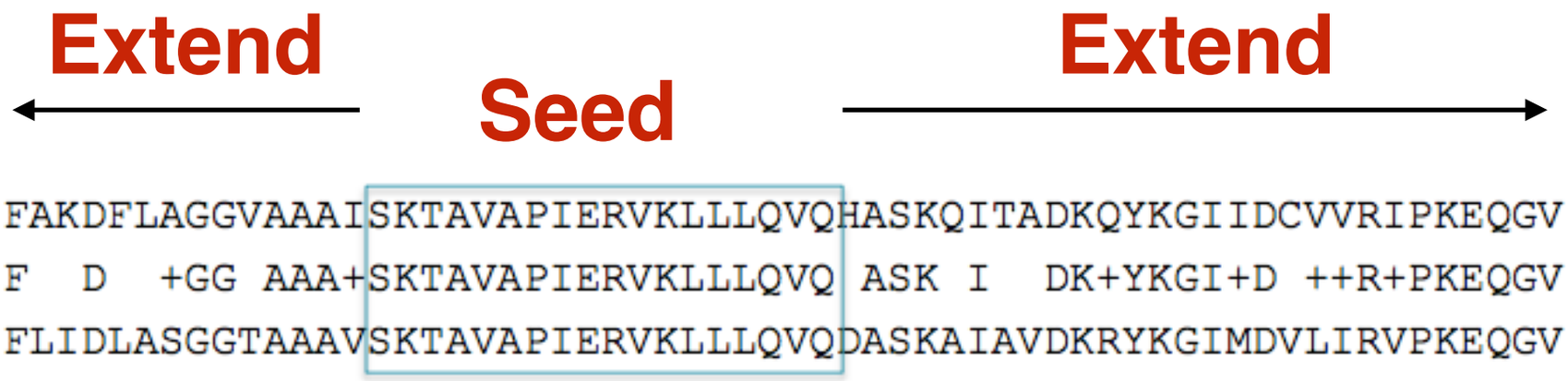
- A molecular biologist may be interested in all conserved regions shared by two proteins, not just their highest scoring pair

- A segment pair (segments of identical lengths) is locally maximal if its score cannot be improved by extending or shortening in either direction

- BLAST attempts to find all locally maximal segment pairs above some score cutoff.

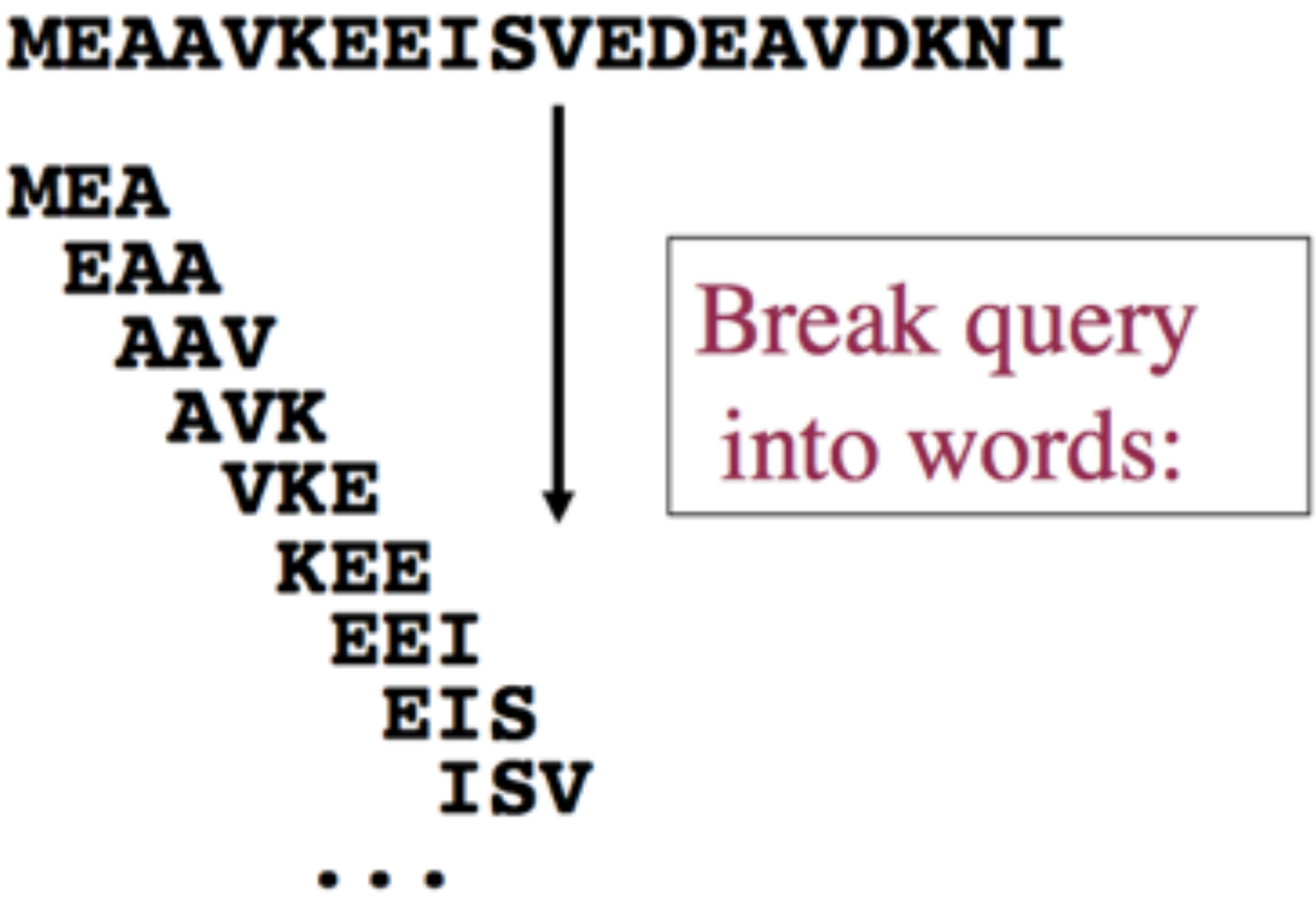
A quick way to find MSPs

- **Homologous sequences tend to have very similar or even identical substrings, also called seeds.**
- **From a seed, it is possible to construct a local HSP/MSP by extending to flanking regions.**



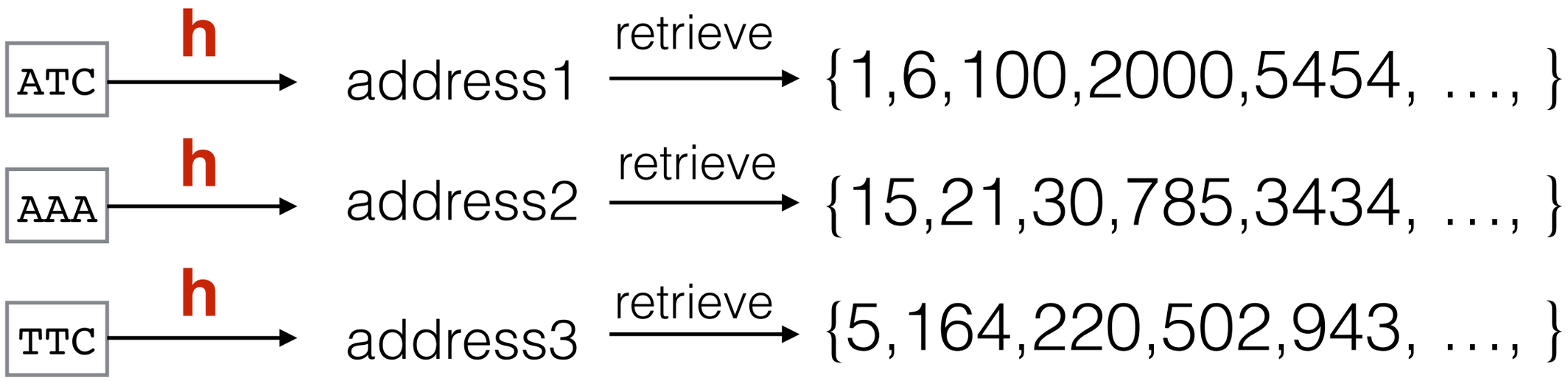
Efficient algorithm?

1. Break query sequence into words

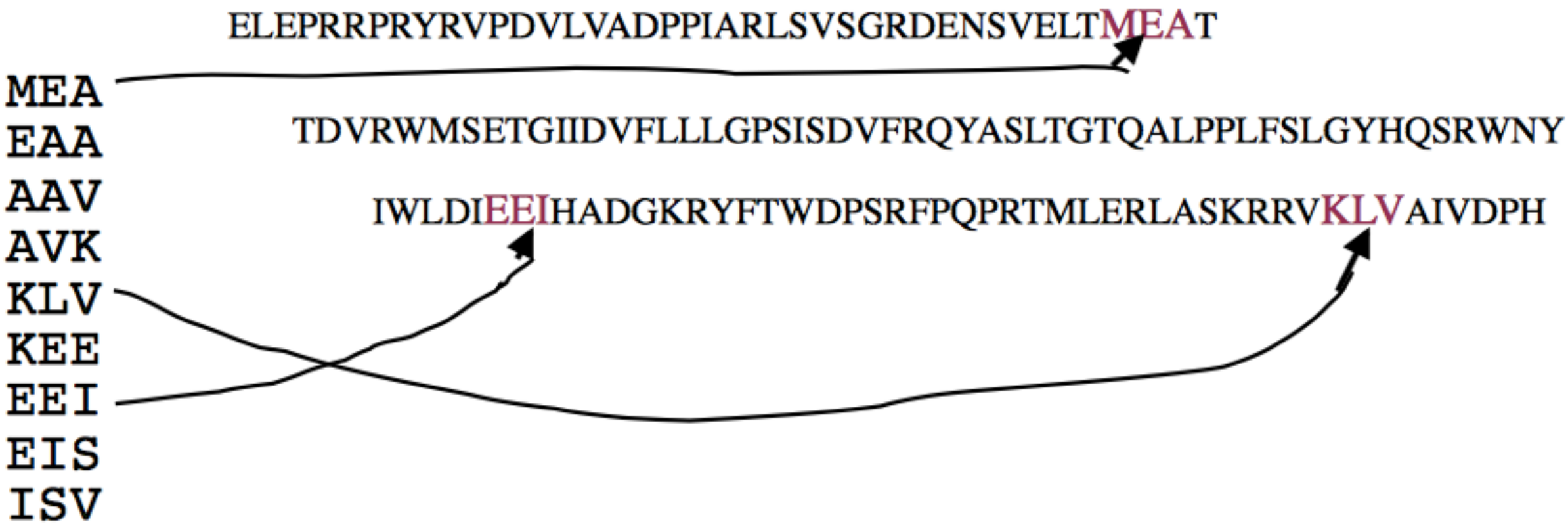


2. Find database hits

- Find exact matches to query words
- Can be done in efficiently
  - Hashing
  - Alternatively AC finite state machine



2. Find database hits



3. Extend hits

1. Find “seeds” (initial matches) of a fixed length (e.g. 11)
2. Try extending an alignment from each seed

How to handle possible mismatches in words?

MVRERKCILCHIVYGSKKEMDEHMRSMLHHRELENLKGRDIS

Query word, W=3 for proteins ↓
(W=11 for nucleotides)

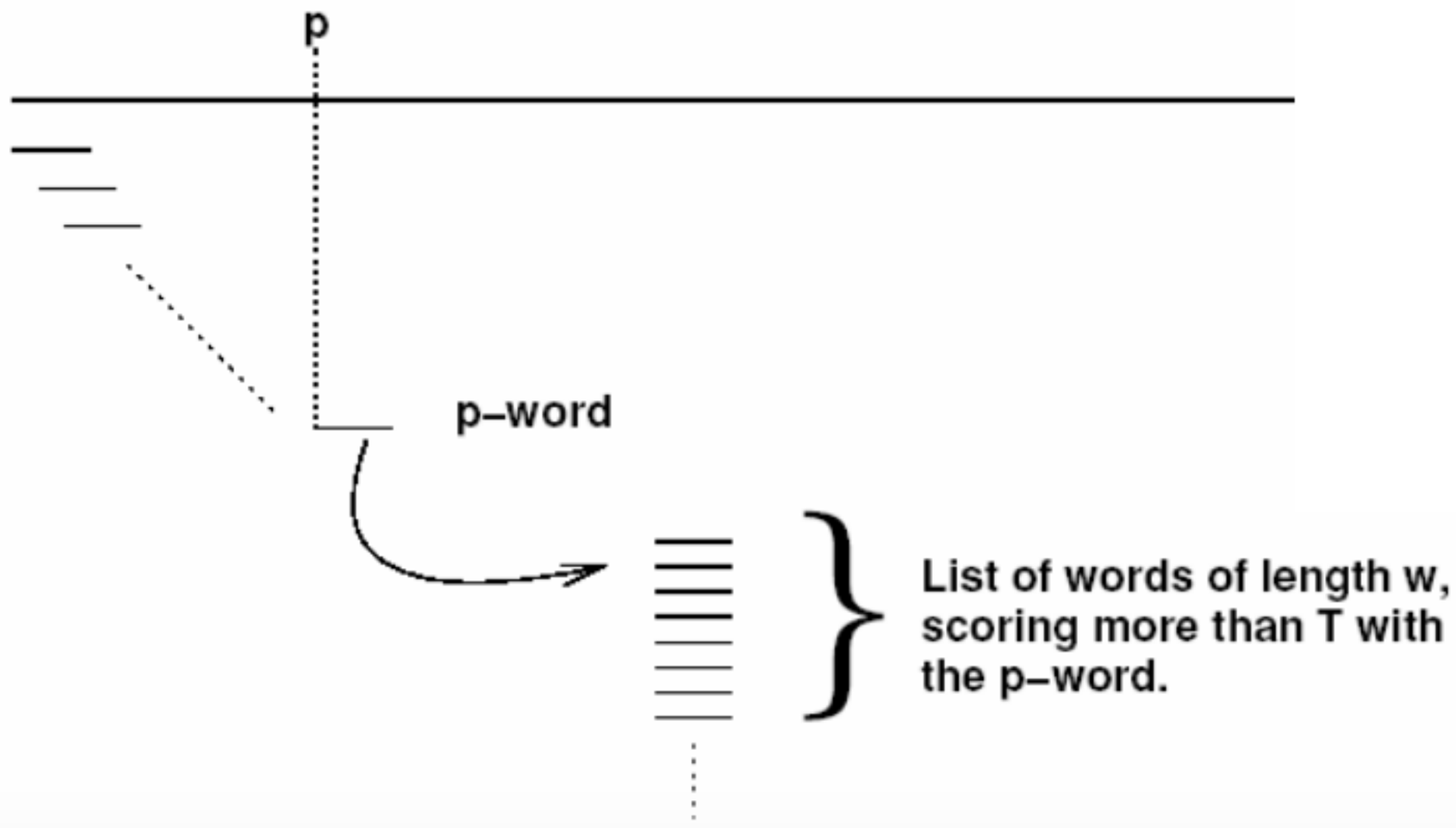
Neighbor words

Word	Score (BL-62)
GSK	15
GAK	12
GNK	12
GTK	12
GSR	12
GDK	11
GQK	11
GEK	11
GGK	11
GKK	11
GSQ	11
GSE	11

How to handle possible mismatches in words?

First step:

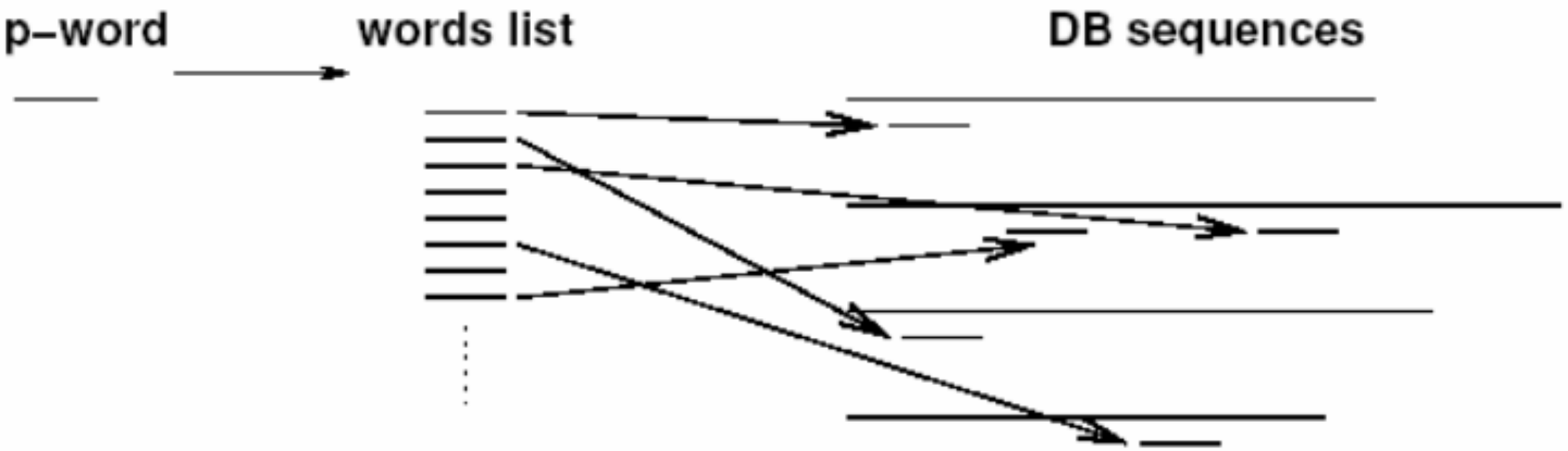
For each position p of the query, find the list or words of length w scoring more than T when paired with the word starting at p:



How to handle possible mismatches in words?

Second step:

For each words list, identify all exact matches with DB sequences:



How to handle possible mismatches in words?

Third step:

For each word match («hit»), extend ungapped alignment in both directions. Stop when S decreases by more than X from the highest value reached by S.

HSP = High Scoring Segment Pair

Parameters

- Word length: 3 for protein, 11 for DNA/RNA
- Thresholds **T** and **S**:
  - BLAST minimizes time spent on database sequences whose similarity with the query has little chance of exceeding this cutoff **S**.
  - Main strategy: seek only segment pairs (one from database, one query) that contain a word pair with score >= **T**
  - Intuition: If the sequence pair has to score above **S**, its most well matched word (of some predetermined small length) must score above **T**
    - Lower T => Fewer false negatives
    - Lower T => More pairs to analyze

Choosing threshold **S**

- BLAST may not find all segment pairs above threshold **S**
- Bounds on the error: not hard bounds, but statistical bounds
  - “Highly likely” to find the MSP

Choosing threshold **S**

- BLAST may not find all segment pairs above threshold S
- Bounds on the error: not hard bounds, but statistical bounds
  - “Highly likely” to find the MSP

- Is the score high enough to provide evidence of homology?
- Are the scores of alignments of random sequences higher than this score?
- What are is the expected number of alignments between random sequences with score greater than this score?

Choosing threshold **S**

- BLAST may not find all segment pairs above threshold S
- Bounds on the error: not hard bounds, but statistical bounds
  - “Highly likely” to find the MSP

- Suppose the MSP has been calculated by BLAST (and suppose this is the true MSP)
- Suppose this observed MSP with a score S.
- What are the chances that the MSP score for two unrelated sequences would be >= S?
- If the chances are very low, then we can be confident that the two sequences must not have been unrelated

Statistics: Question

- Given two random sequences of lengths m and n
- What is the probability that they will produce an MSP score of >= S ?

Statistics: intuition

Given a binary 0/1 sequence and a query string of k consecutive ones

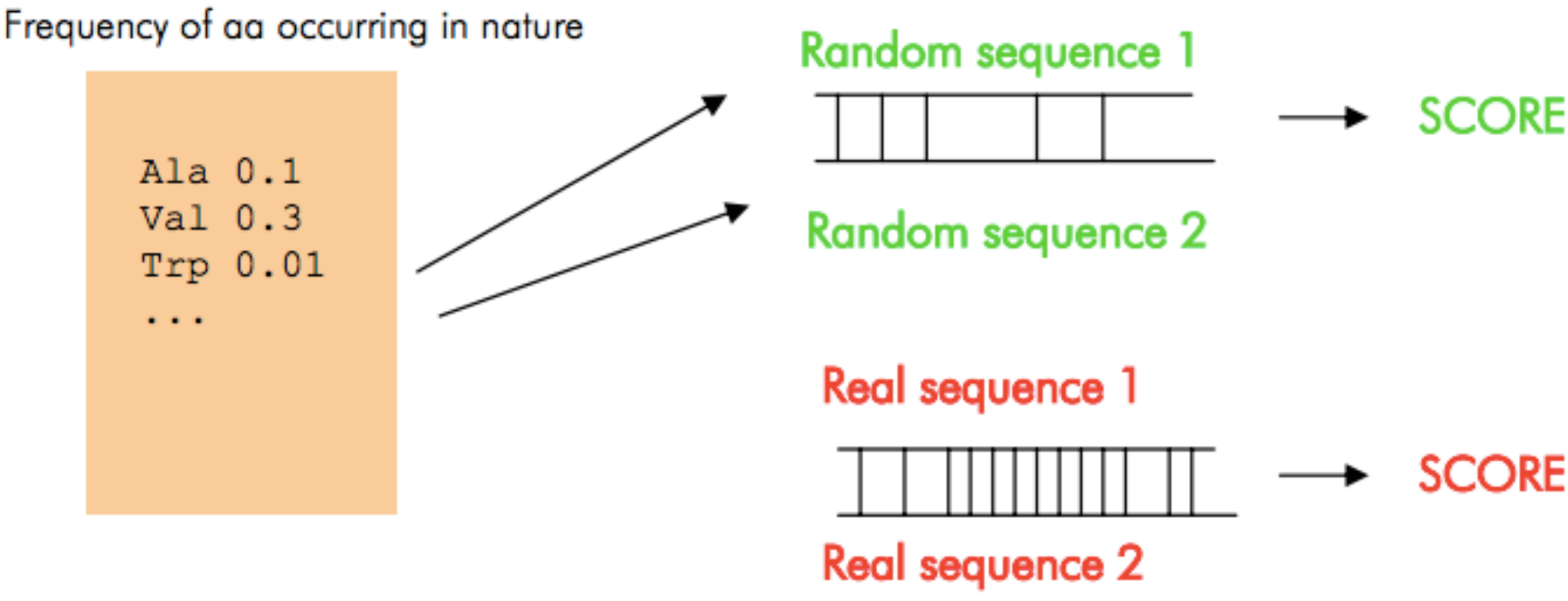
- Probability in a sequence of length k: $1/2^k$
- Probability in a sequence of length k+1?
  - 1 - (1 - $1/2^k$)^2
- How about the probability in a sequence of length k+n?
  - 1 - (1 - $1/2^k$)$^{n+1}$
- The longer the sequence, the more likely you are going to get k ones by chance!

Statistics: more intuition

The probability will depend on:

- How long is are the sequences (the longer the easier to get a local score above threshold by chance)
- Scoring matrix
- Distribution of amino acids in each sequence

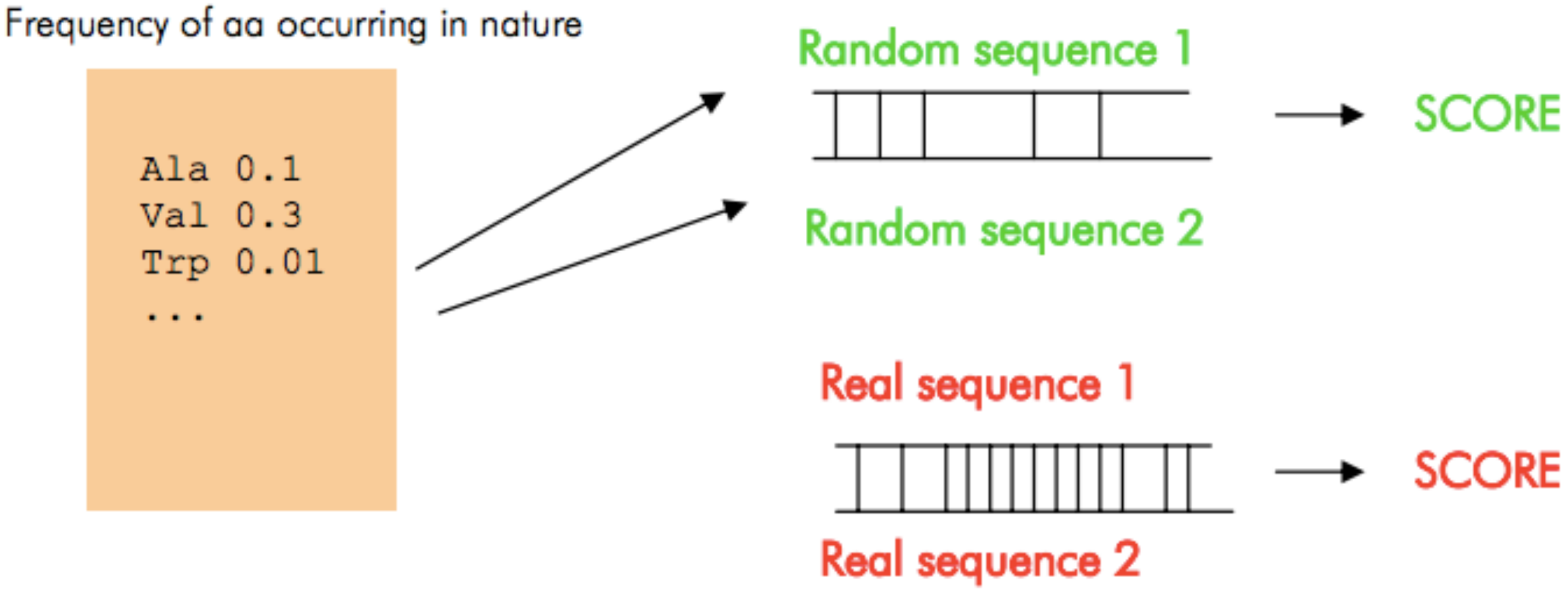
Statistics: Intuition



Approach

⇒ Evaluate the probability that a score between random or unrelated sequences will reach the score found between two real sequences of interest:

If that probability is very low, the alignment score between the real sequences is significant.

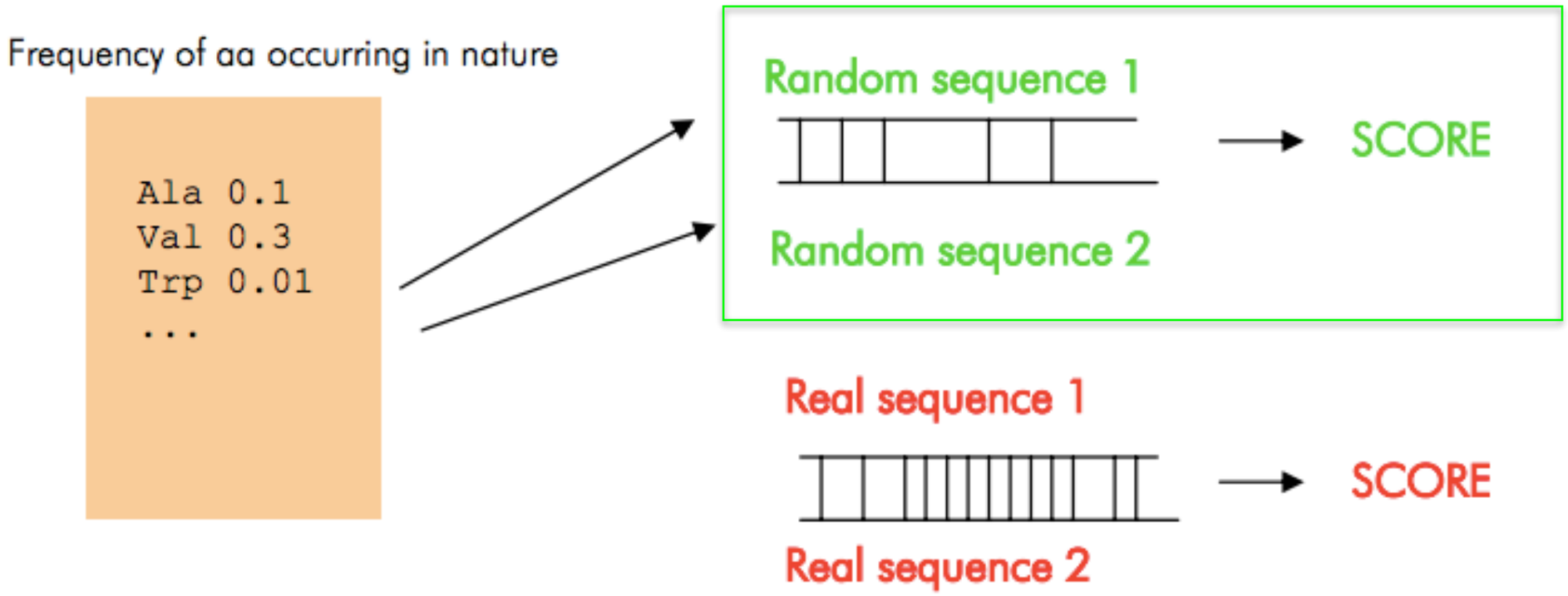


If SCORE > SCORE => the alignment between the real sequences is significant

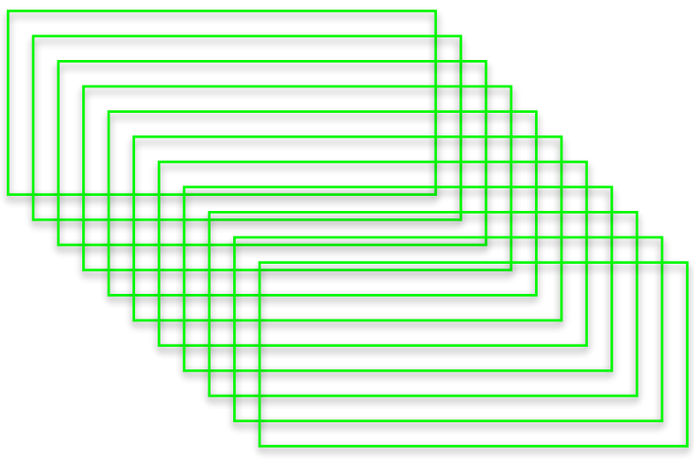
How to compute the probability?

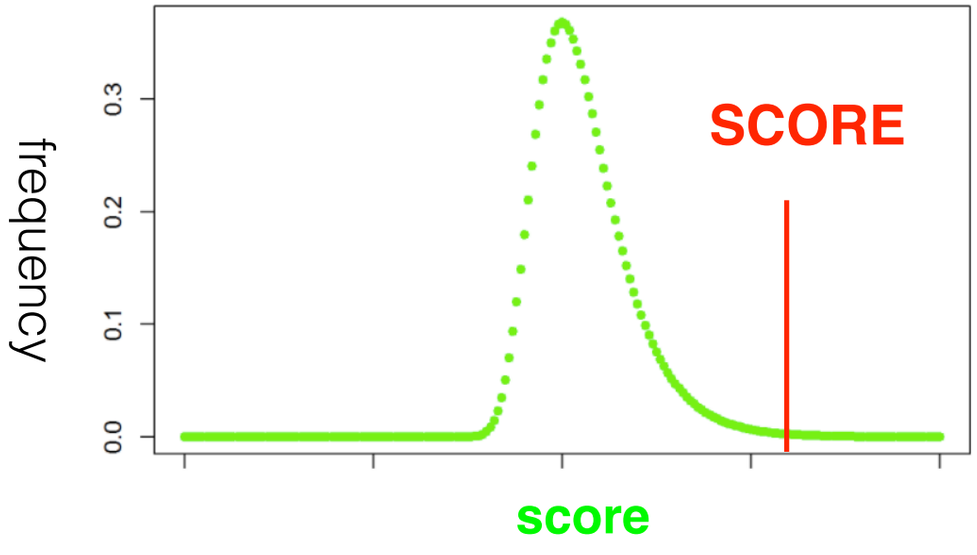
Simulation

1. Generate many random sequence pairs



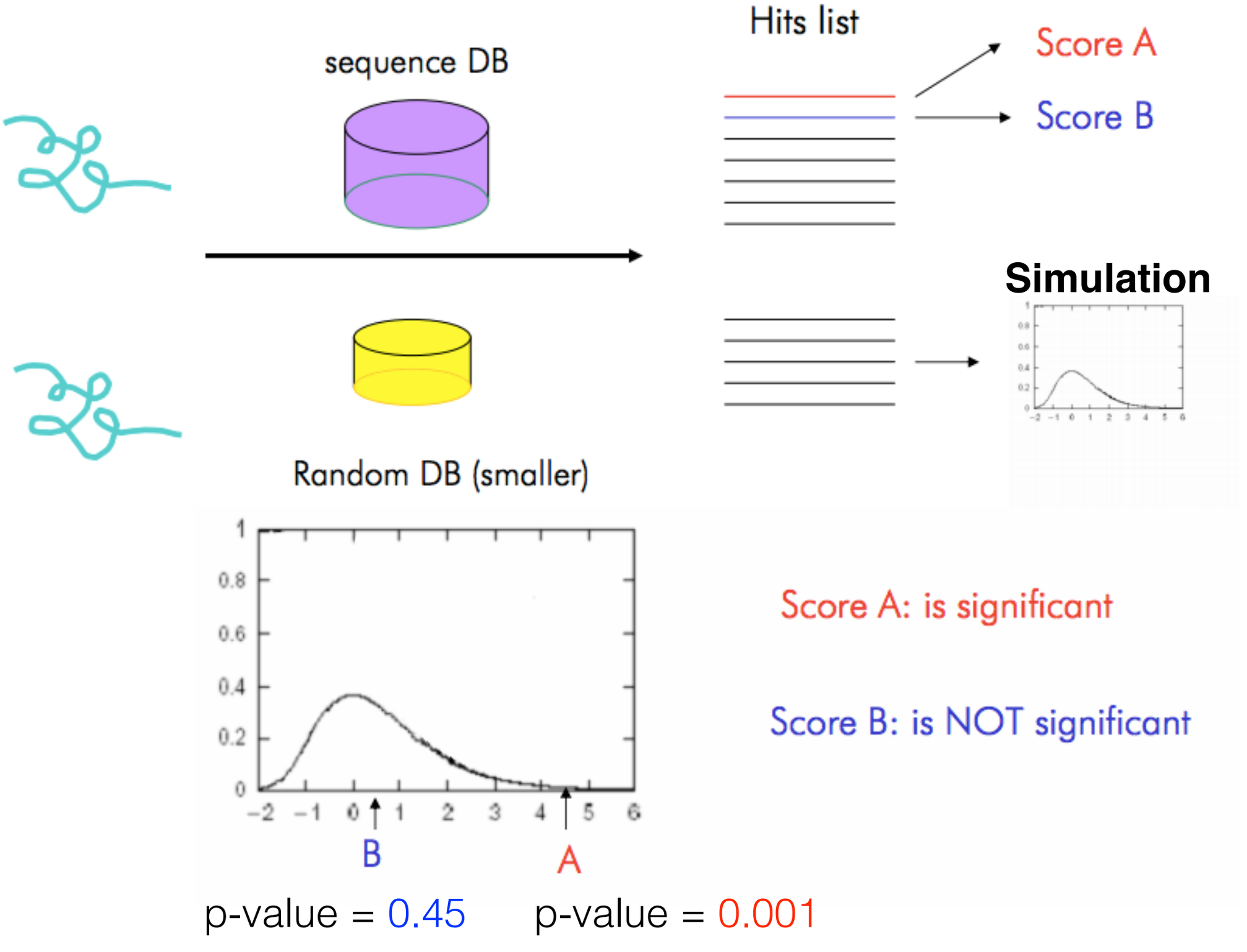
2. Compute the distribution of the SCOREs





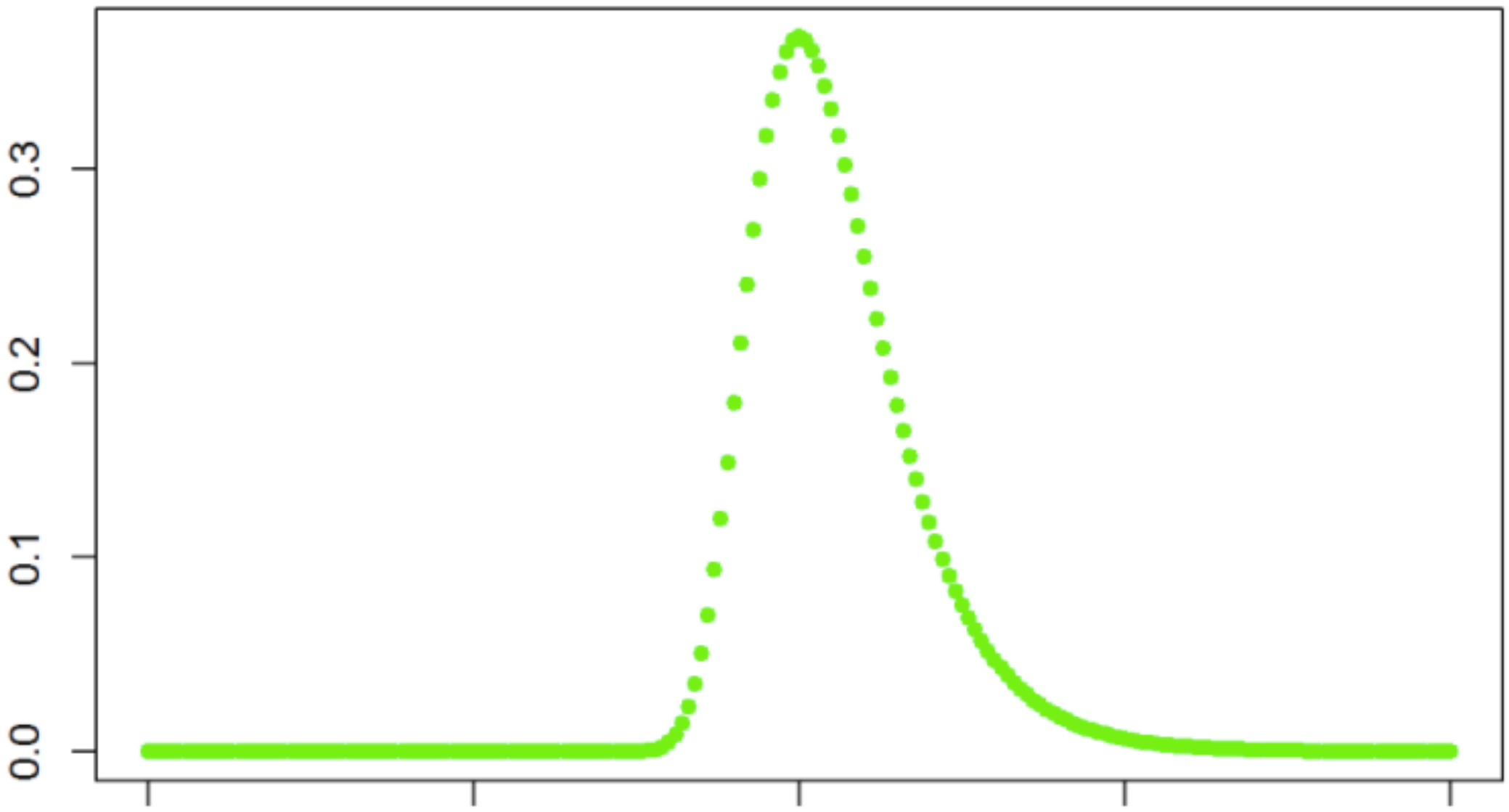
How to compute the p-value (probability)?

Statistical test



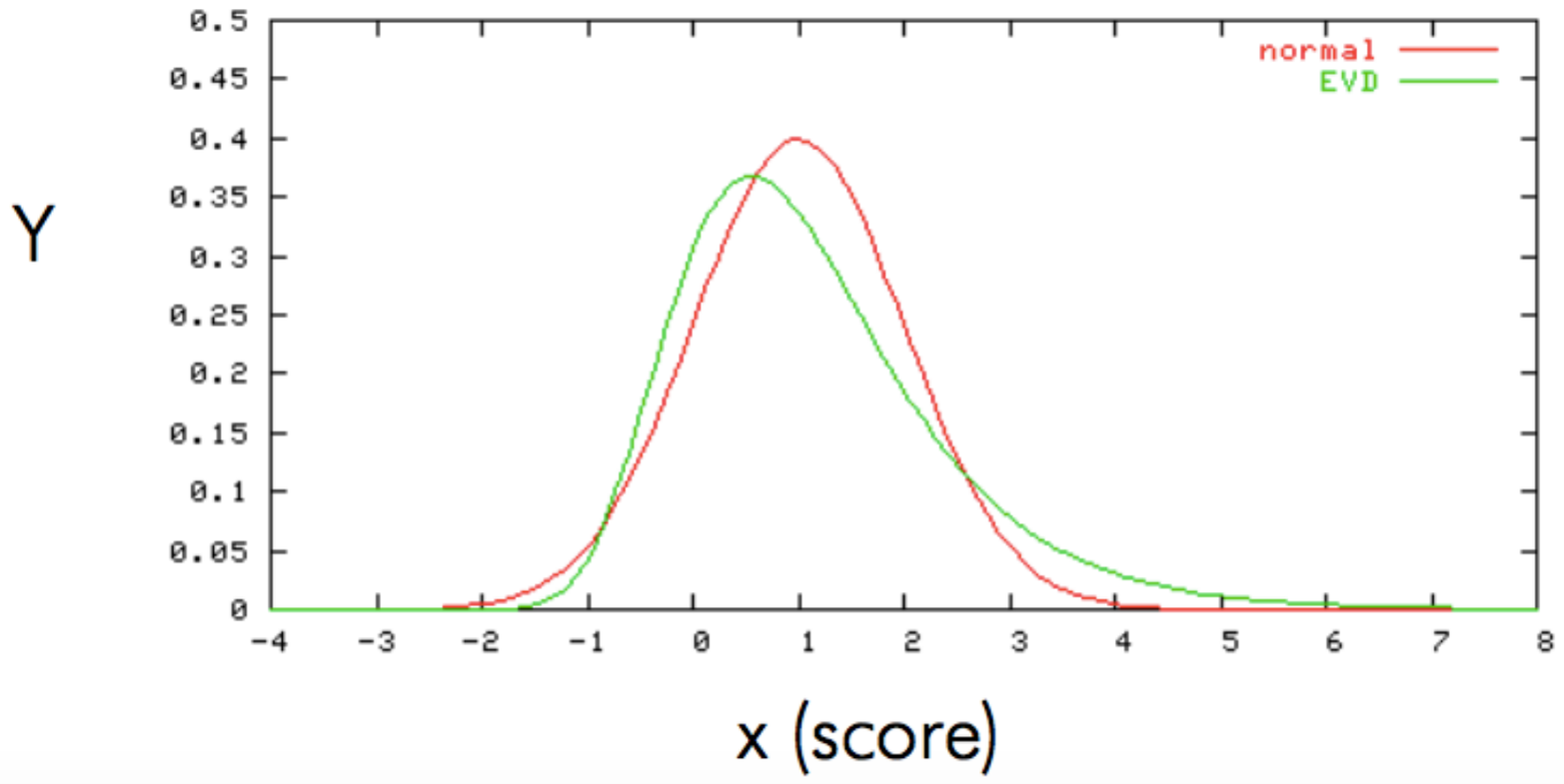
Is this efficient enough?

Another observation

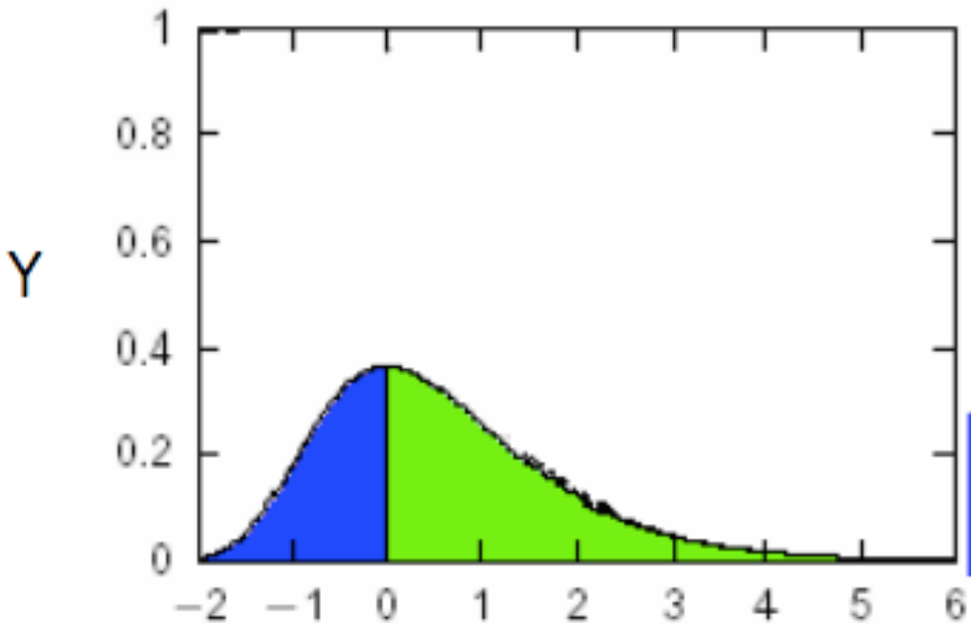


Extreme value distribution

Karlin and Altschul observed that in the framework of local alignments without gaps: the distribution of random sequence alignment scores follow an EVD.



Extreme value distribution



$$Y = \lambda \exp[-\lambda(x-\mu) - e^{-\lambda(x-\mu)}]$$

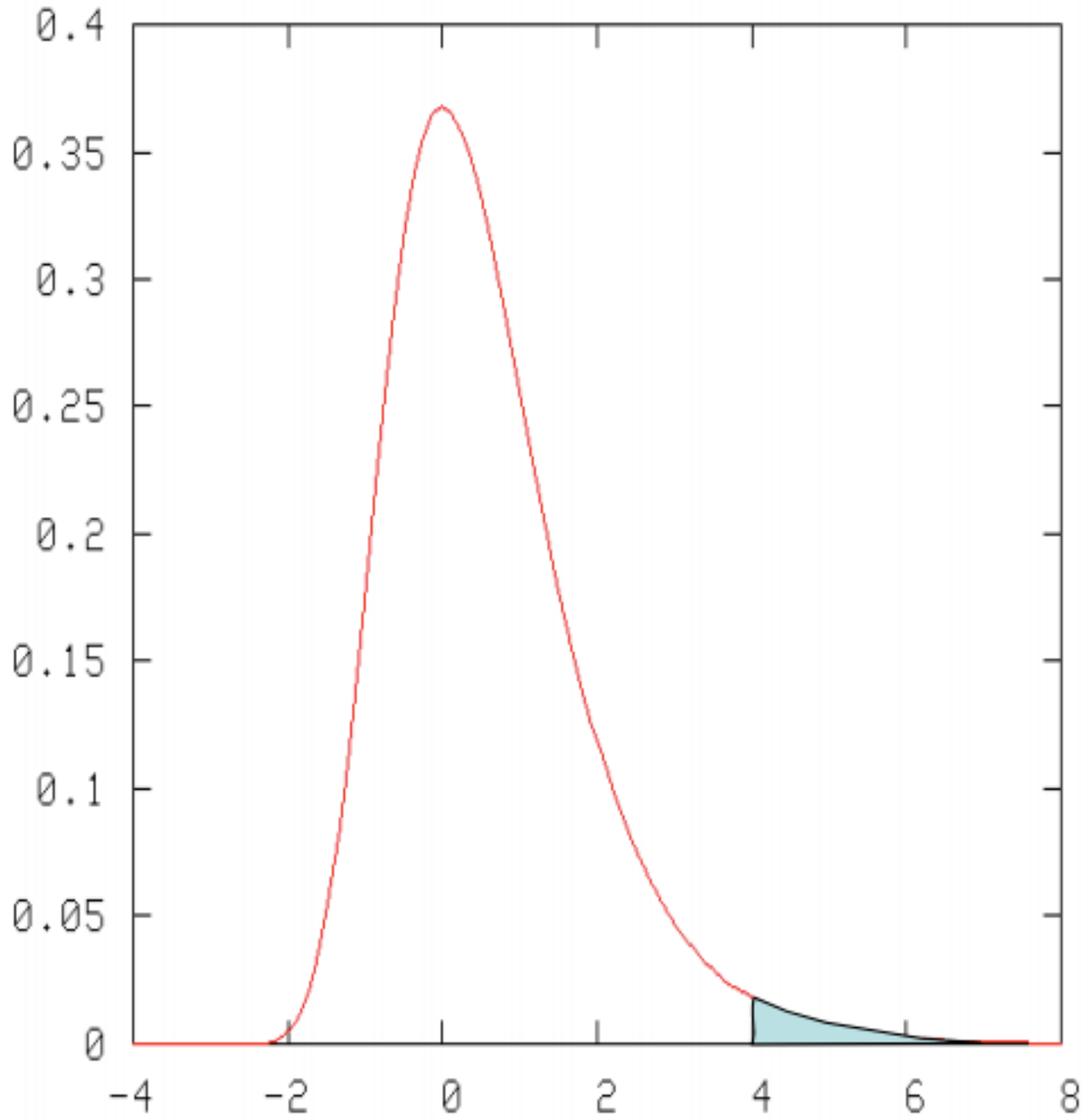
$$\int$$

$$P(Z < x) = \exp[-e^{-\lambda(x-\mu)}]$$

$$P(Z \geq x) = 1 - \exp[-e^{-\lambda(x-\mu)}]$$

P-value = the probability of obtaining a score equal or greater than *x* by chance

Compute a p-value



- The probability of observing a score >=4 is the area under the curve to the right of 4.
- <u>For an *Unscaled EVD:*</u>

$$P(S \geq x) = 1 - e^{(-e^{-x})}$$

$$P(S \geq 4) = 1 - e^{(-e^{-4})}$$

$$P(S \geq 4) = 0.018149$$

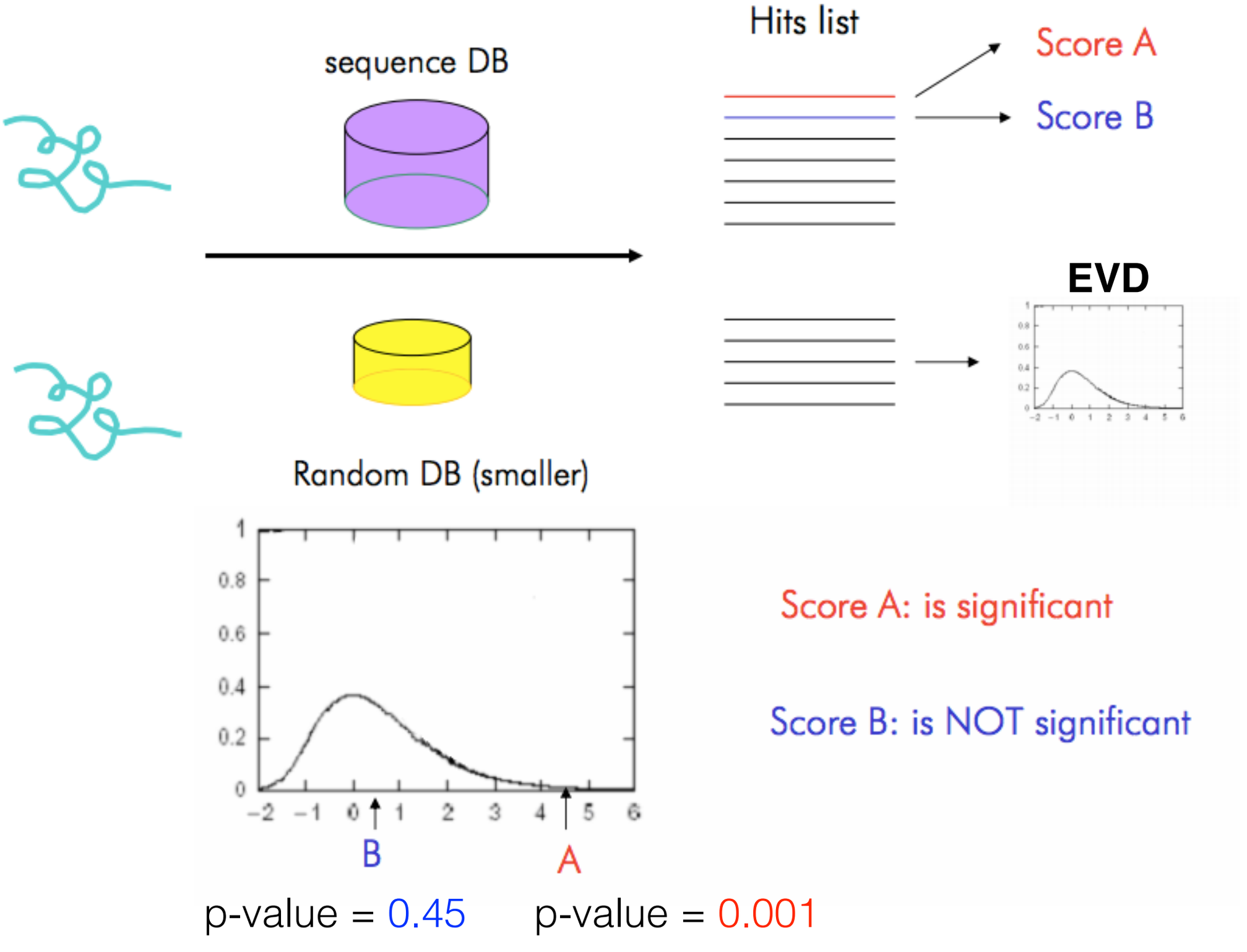
Parameters

$$P(Z \geq x) = 1 - exp[-e^{-\lambda(x-\mu)}] \qquad (1)$$

μ, λ : parameters depend on the length and composition of the sequences and on the scoring system: μ is the mode (highest point) of the distribution and λ is the decay parameter

-They can me estimated by making many alignments of random or shuffled sequences.

Statistical test



Significance: P-value and E-value

In a database of size N: P x N = E

- P-value:

Probability that an alignment with this score occurs by chance in a database of size N.

The closer the P-value is towards 0, the better the alignment

- E-value:

Number of matches with this score one can expect to find by chance in a database of size N.

The closer the E-value is towards 0, the better the alignment

→ Smaller E-value, more significant in statistics

Bigger E-value , by chance

E[# occurrences of a string of length m in reference of length L] ~ $L/4^m$

Parameters

$$P(Z \geq x) = 1 - exp[-e^{-\lambda(x-\mu)}] \qquad (1)$$

μ, λ : parameters depend on the length and composition of the sequences and on the scoring system: μ is the mode (highest point) of the distribution and λ is the decay parameter
-They can me estimated by making many alignments of random or shuffled sequences.
- For alignments without gaps they can be calculated from the scoring matrix and then :

$$P(Z \geq x) = 1 - exp[-Kmne^{-\lambda x}] \qquad (2)$$

K: is a constant that depend on the scoring matrix values and the frequencies of the different residues in the sequences.
m,n : sequence lengths

E-value

Approximation:

if x is very small, then 1-exp(-x) can be approximated by x

Therefore,

$$P(Z>=x) \sim e^{-\lambda(x-\mu)} = Kmn\, e^{-\lambda x}$$

So E-value = DatabaseLength * p-value

$$\text{E-value} = KNme^{-\lambda x}$$

where N is the database size (not the aligned length n)