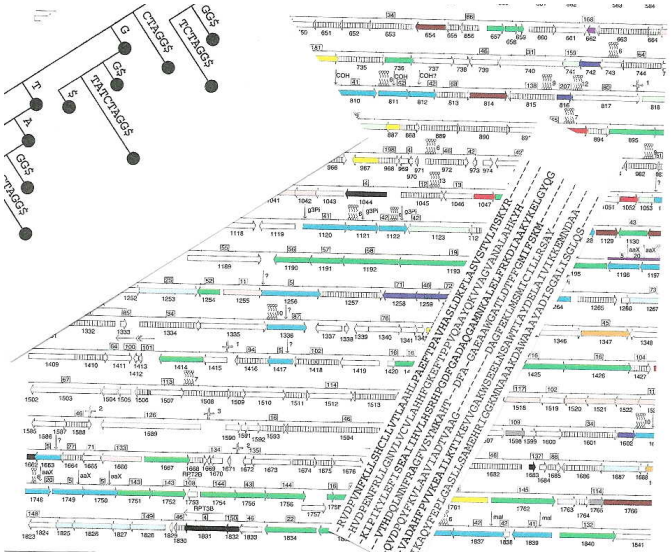
Lecture 8:Motifs and Motifs finding

(with a section on Chip-Seq)

Principles of Computational Biology

Teresa Przytycka, PhD



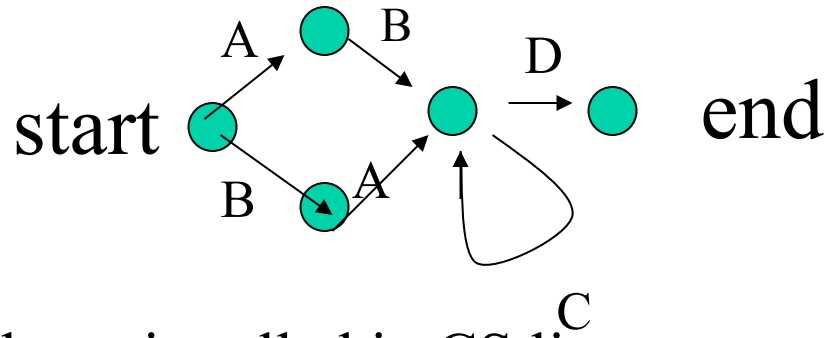
Motifs

- Motif is a region (a subsequence) of protein or DNA sequence that has a specific structure
- Motifs are candidates for functionally important sites
- Presence of a motif may be used as a base of protein classification

Representation of motifs

- Profile or sequence logos
- Regular expression

Describing patterns using regular expressions

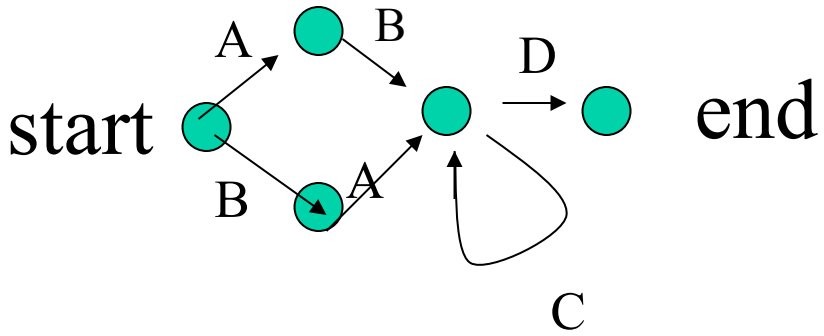


A graph like one above is called in CS literature a finite automaton can be used do describe a sequence family (CS literature such a set of sequences is called a language):

Take any path from "start" to "end" and as you go print the letters that label the edges you used. Any sequence that can be printed in this way will be called generated (CS term: accepted) by the automaton.

E.g.: ABCCCCD; BACCD;…..

Regular expressions



A finite automaton can be translated to so called regular expressions:

Notation:
[choice1, choice2,….] = a set of choices in a brunching point ,
- = "followed by"
* = repeat 0 or more times

E.g. The regular expression describing automaton above:

[A-B , B-A]-C*-D

PROSITE

- A data base of regular expression that describe protein motifs
- Developed since 1988
- 1999 – authors recognize that some protein families are characterized by profiles than regular expression and extended the data base to contain profiles
- Profiles are generated from multiple sequence alignments

PROSITE patterns

- PROSITE fingerprints are described by regular expressions
- Rules:
  - Each position is separated by a hyphen
  - One character denotes residuum at a given position
  - […] denoted a set of allowed amino acids
  - (n) denotes repeat of n times
  - (n,m) denoted repeat between n and m inclusive
  - X – any character

  Example [EDQH]-x-K-x-[DN]-G-x-R-[GACV]

  Ex. ATP/GTP binding motive [SG]-X(4)-G-K-[DT]
- There is a number of programs that allow to search databases for PROSITE patterns

Finding motifs

- Method I: extracted from multiple sequence alignment .
  - EMOTIF
  - PRINTS
- Method II: Gibbs sampling – a method that allows to find motifs in the absence of multiple sequence alignment

  Reference: Lawrence,.C.E. et al (1993) Science 263, 208-214
- Method III: Exhaustive or dedicated search

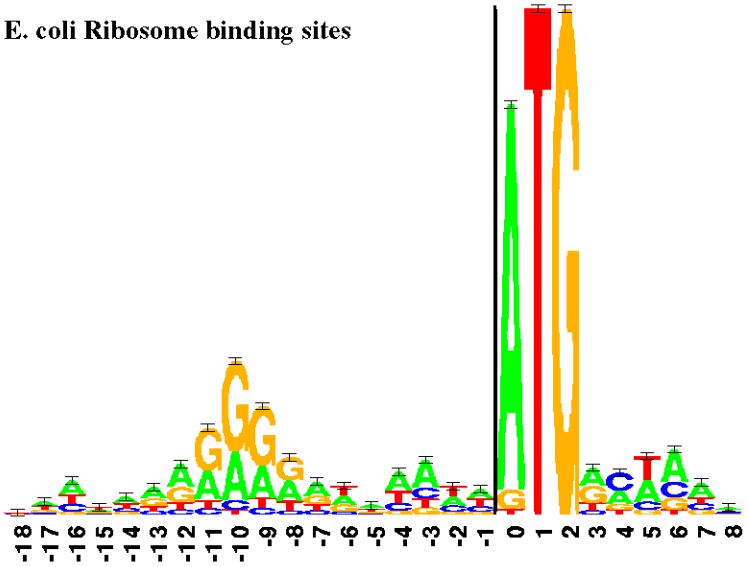
Recognition of transcription factor binding sites

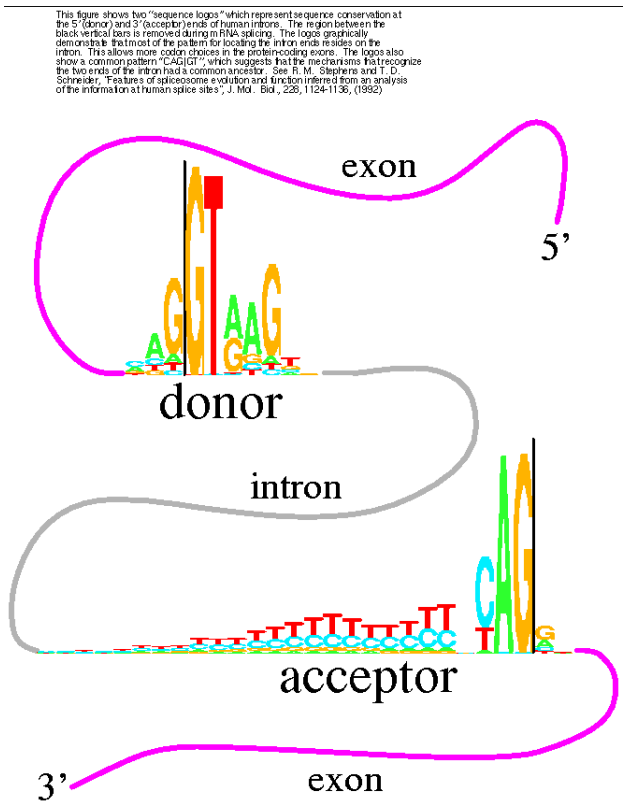
- Transcription Factors = proteins that bind DNA, typically upstream or close to the transcription start site and regulate the expression of the gene by activating or inhibiting the transcription machinery
- Little is know about most of transcription factors it particular what binding sites most of them are.
- Co-regulated genes – genes to which are regulated by the same transcription factor

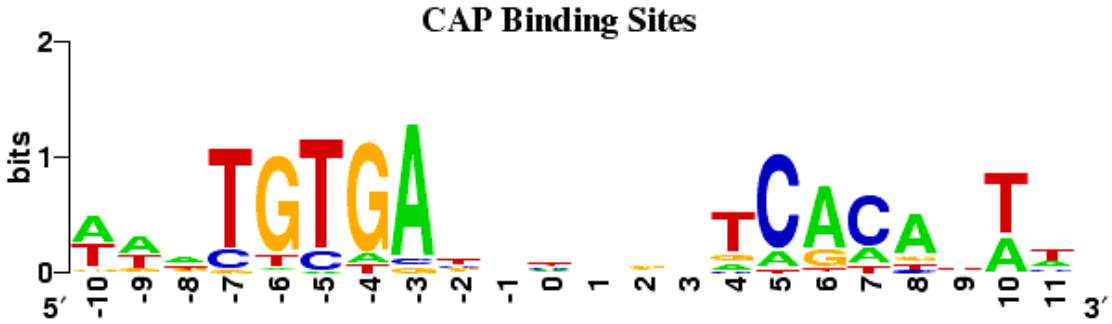
Typical setting for computational finding of transcription binding sites

- Give is collection of regulatory regions of genes that are believed to be co-regulated.
- Goal – fund sort DNA motifs that are overrepresented.
- So what is the problem?
  - Binding motifs are typically short
  - They have significant variability
- There is a large number of other algorithms: (AlignACE, MEME, Weeder, YMF…)

Examples of binding sites profiles

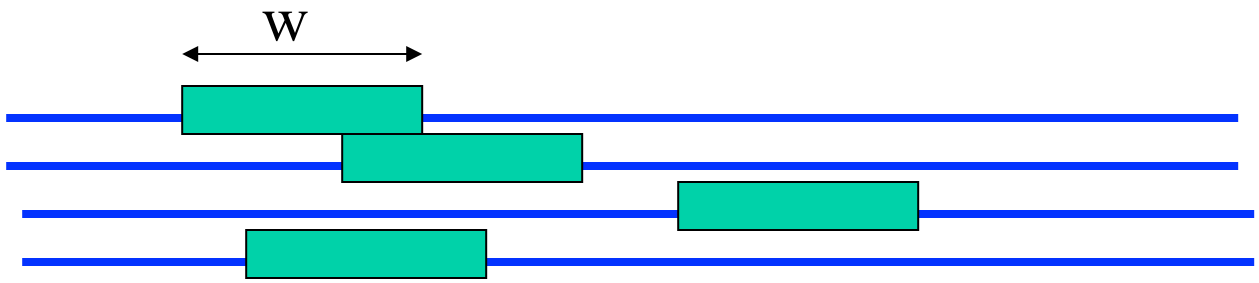






Finding Motifs with Gibbs Sampling Method

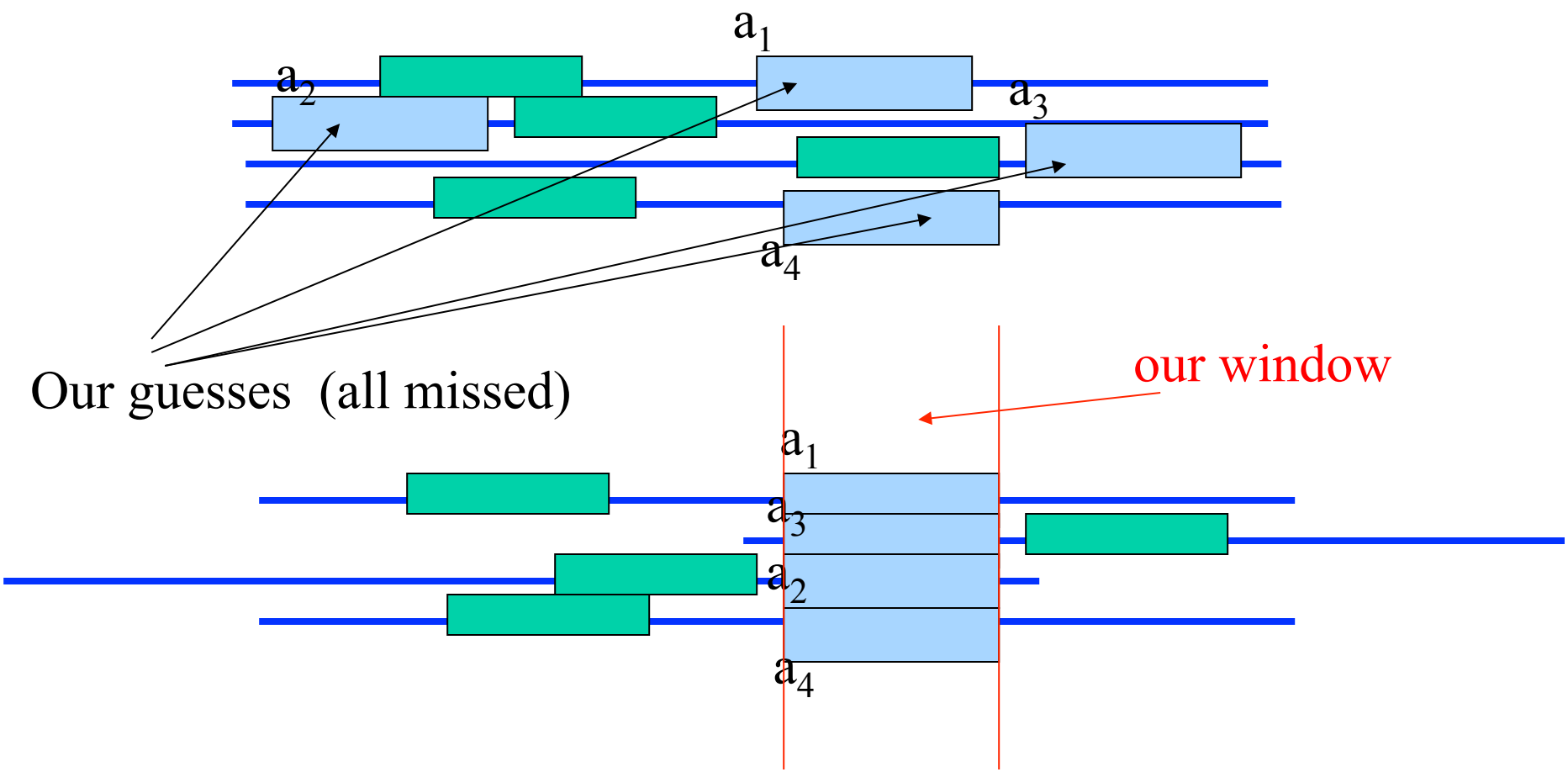
- Assumption: Given is a set of sequences that are believed to share (one) common motif
- Motif is assumed to have length w



Idea: Look for a signal using a Monte Carlo method

Initialization: Make a guess

- Choose randomly at each sequence a candidate position for the motif



The basic algorithm

- Initialization: Choose randomly at each sequence a candidate position for the motif
- Iterate the following two steps until convergence:
  - Predictive update: detecting current signal from the motif identified so far
  - Sampling: improving the signal

Predictive update step – leave one out

- One of N sequences, say z, is removed (randomly or in specific order)
- The pattern frequencies (position specific scoring matrix) and background probabilities are computed with z excluded

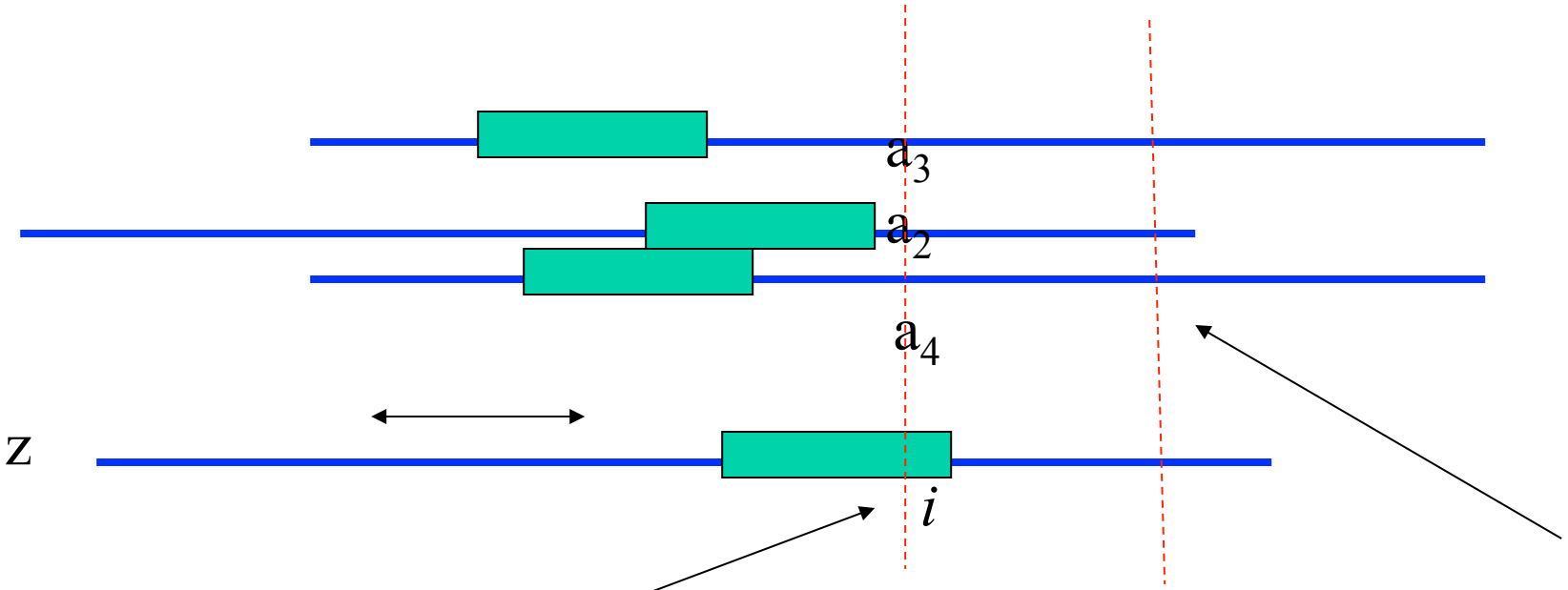
Two evolving data structures maintained by the algorithm

- Pattern description in the form of probabilistic model of residue frequencies
  - $q_{i1},\ldots q_{i20}$; i = 1,…w (q_{ik} is the frequency of amino-acid k on position i of the pattern)
  - $p_1,\ldots p_{20}$; background frequency
- Local alignment description
  - $a_1,\ldots a_N$; N-number of sequences;
  - a_i – beginning of the pattern in the i^{th} sequence

Sampling Step

- Every possible sequence x of length w is aligned with the profile in the window
- Calculate probability Q_x of generating x according to probability distribution defined by current pattern description (profile) (q_{i1},… q_{i20} ; i=1..w)
  - e.g, probability of ATCA = $q_{1A}\, q_{2T}\, q_{3C}\, q_{4A}$
- Calculate probability P_x of generating x according to background probability distribution p_1,… p_{20}
- Assign weight $A_x = Q_x / P_x$ to the sequence x
- Choose with probability weighted by A_x the sequence x to be aligned with current patter

Sampling Step



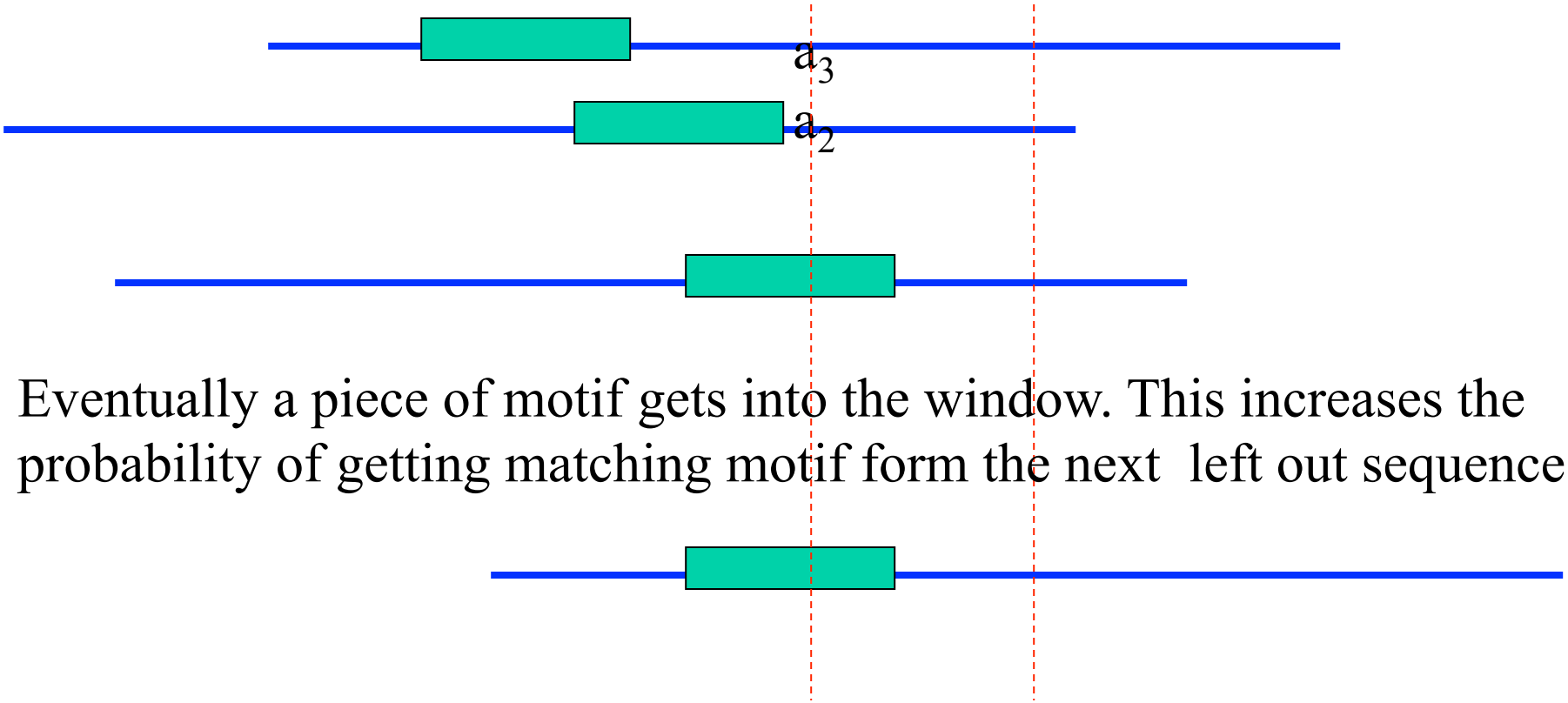
•*Try all possible alignments of z to the profile defined by the pattern we found so far.*

•*Each position i has some probability p_i of being good (if no pattern all position should be equally likely)*

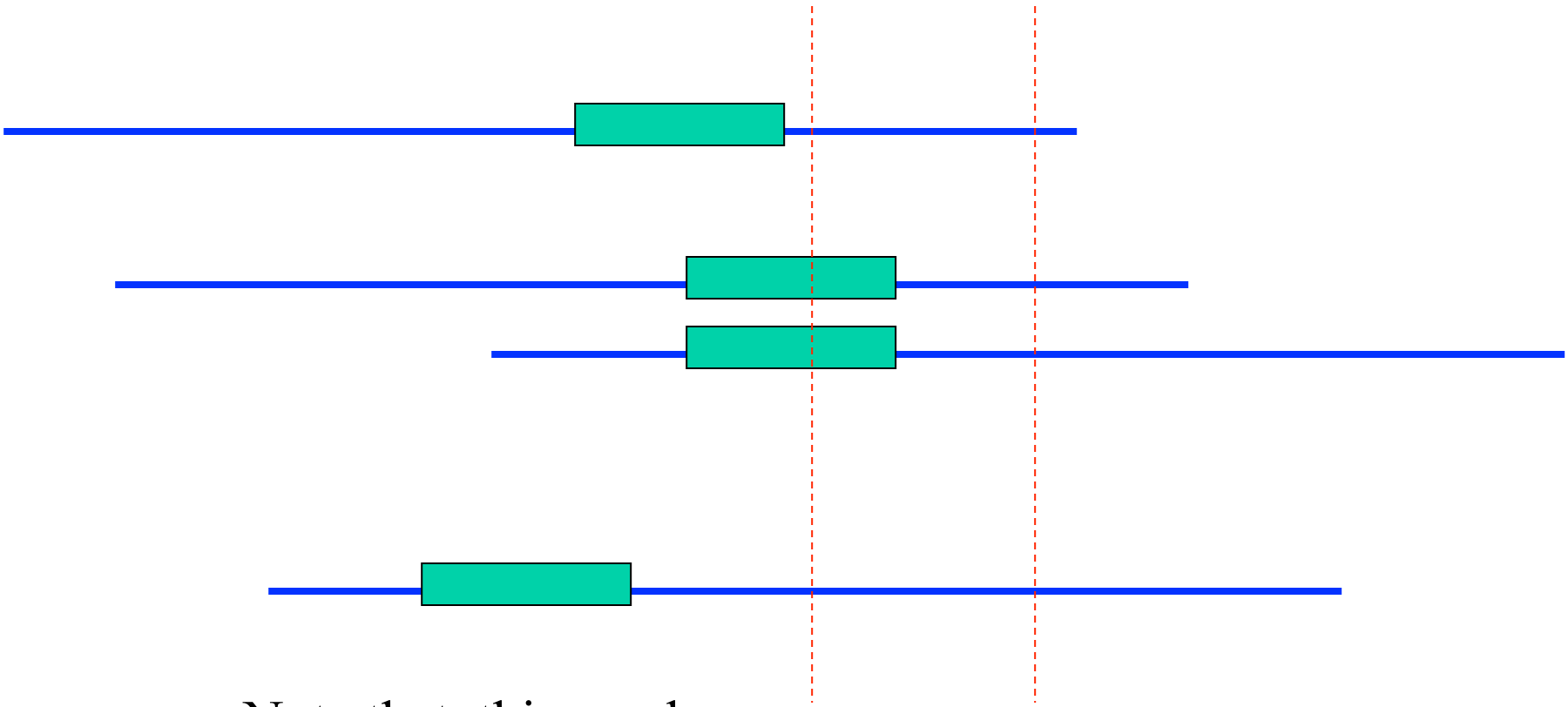
•*Chose fragment from i to i+w to be be aligned to the window with probability p_i.*

We know the Profile of this alignment (still no pattern found so this profile should correspond to random a.a. distribution)

Why it is supposed to work

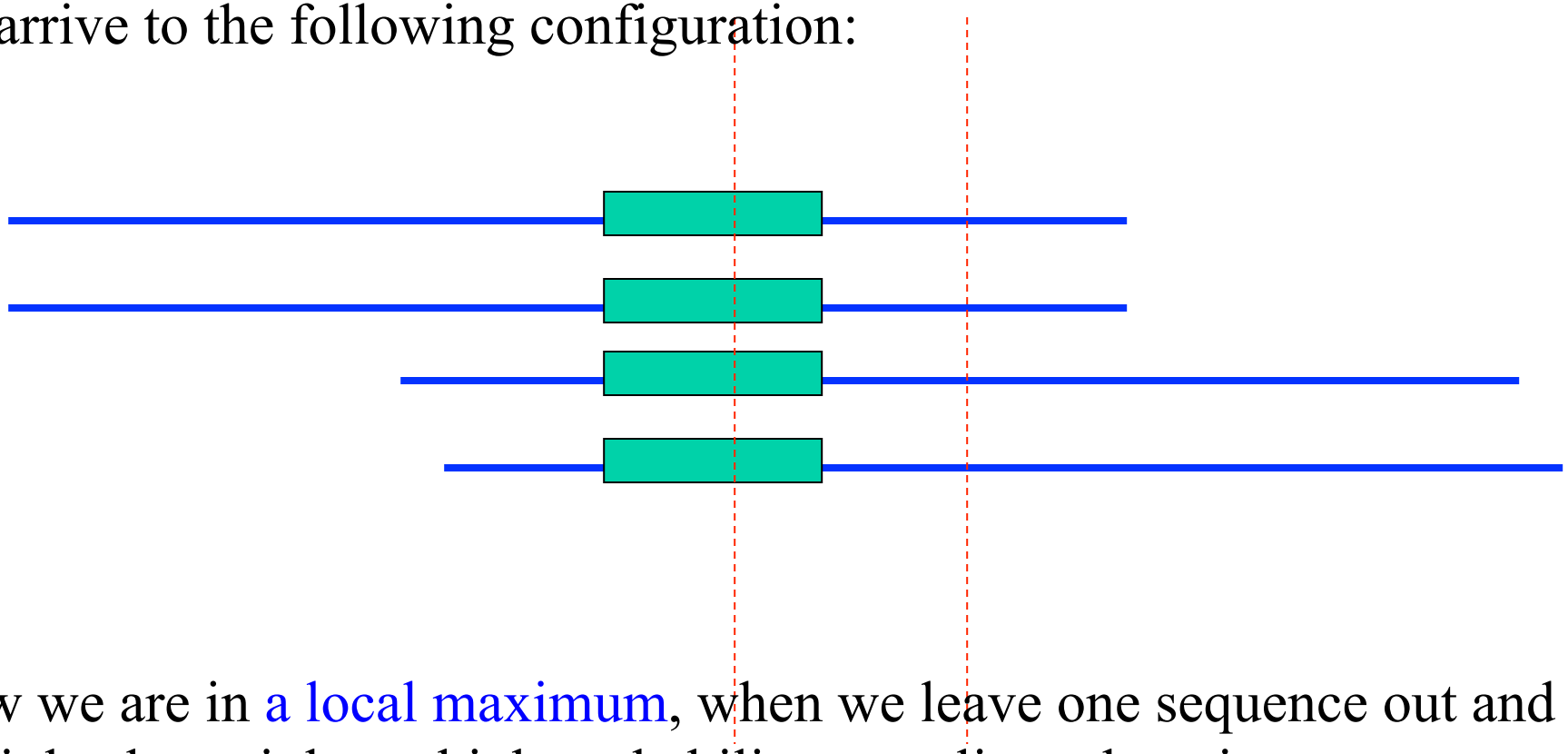


Eventually a piece of motif gets into the window. This increases the probability of getting matching motif form the next left out sequence

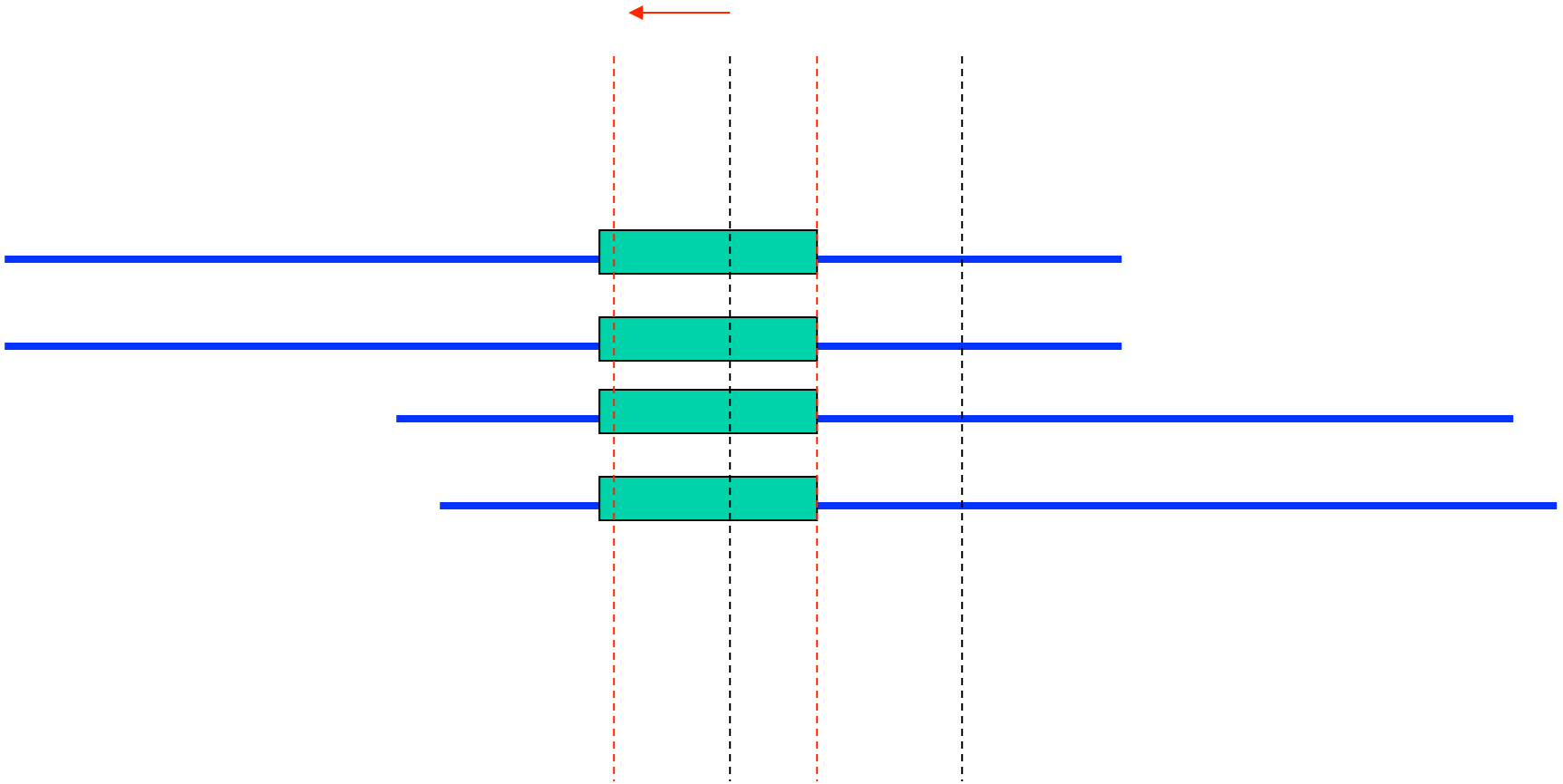


Note that this one has a good chance to fit the pattern in the window

Finally after a number of iteration we have a good chance to arrive to the following configuration:

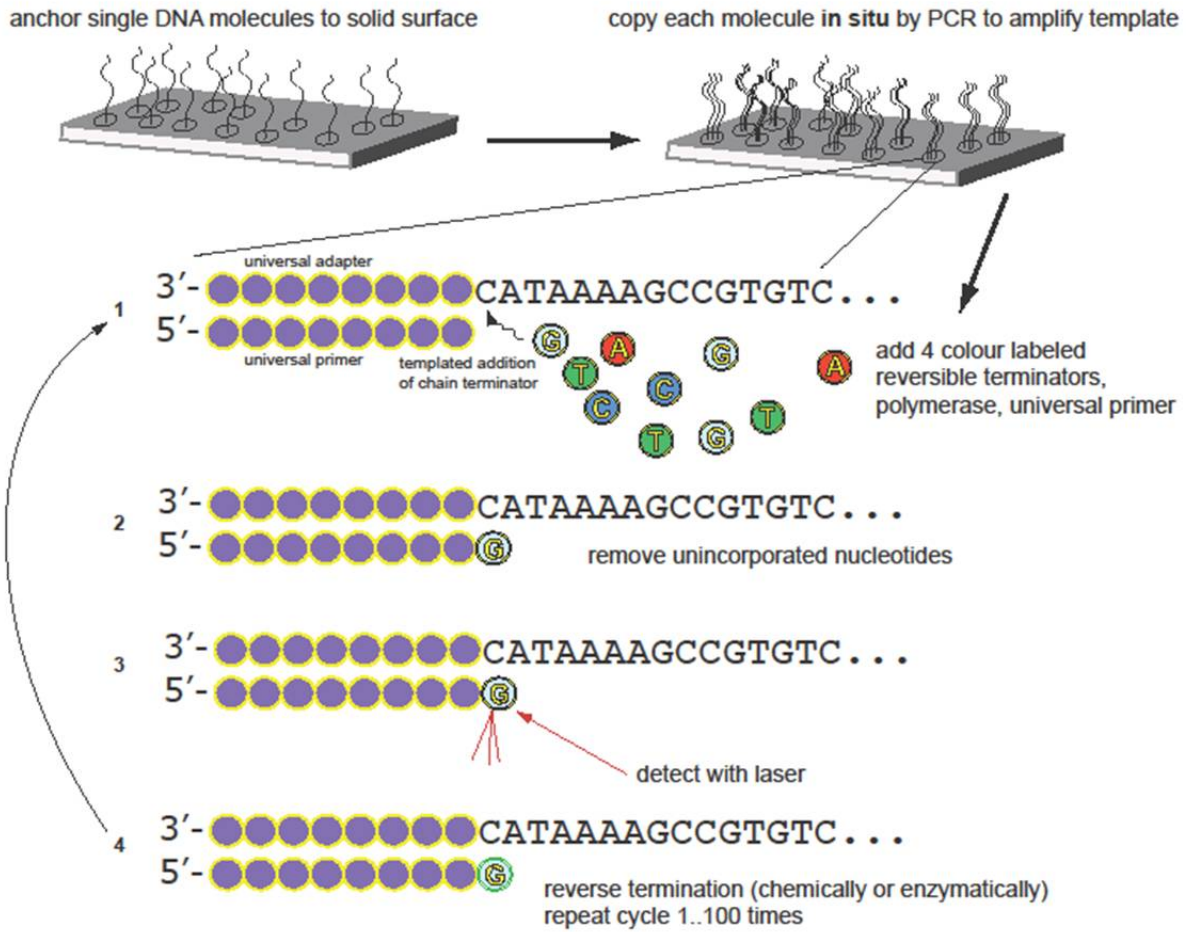


Now we are in a local maximum, when we leave one sequence out and put it back out it has a high probability to realign where it was.
When no further improvement is observed we assume that have a pattern or a part of it in the window and we try to move the window slightly to the sides to discover the rest of the motif



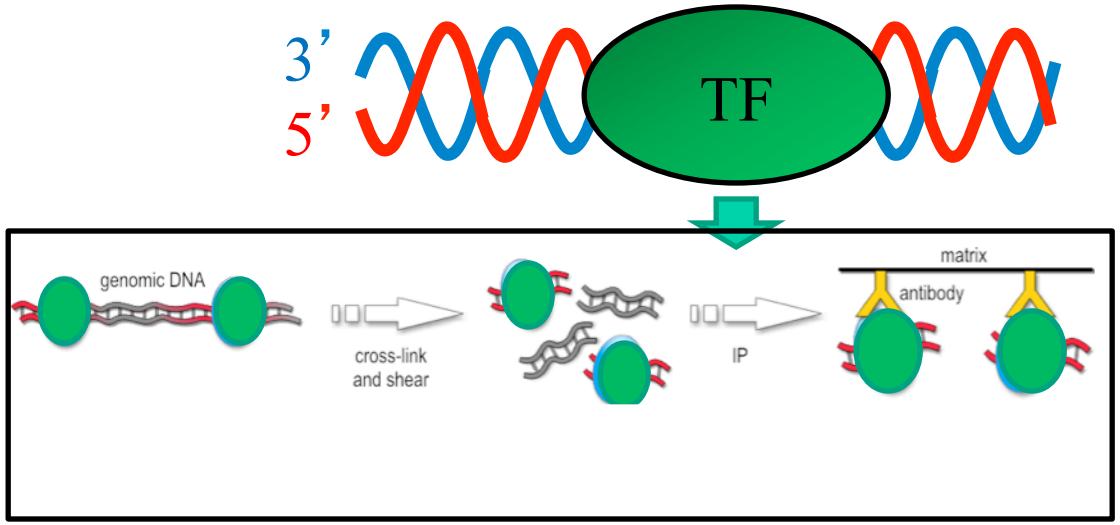
High throughput sequencing technologies (next generation sequencing)

DNA molecules are physically bound to a surface, and sequenced in parallel.



http://molonc.bccrc.ca/wp-content/uploads/2009/09/next-gen-seq-fig-1.jpg

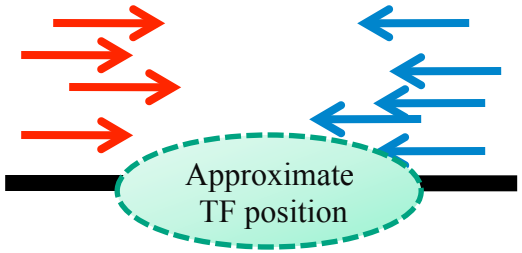
How this facilitates identification of binding motifs



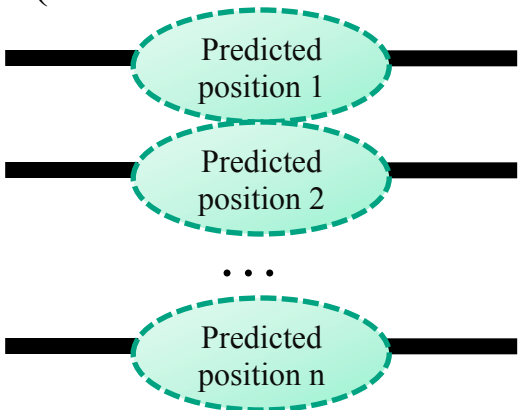
Sequence DNA fragments

Search for motifs in predicted regions (

Mapping of sequenced tags to the genome



Reference genome



Aligning short reads

Challenge – alignment a large number of short reads
(hundreds of billions of bases total)
Programs:
-SOAP, Maq, Bowtie, RMAP, ZOOM, SHRiMP, Eland
Each tool builds a hash table of short oligomers present in either the reads (SHRiMP, Maq, RMAP, and ZOOM) or the reference (SOAP).
Eland - a commercial alignment program from Illumina

Bowtie:
Ben Langmead, Cole Trapnell, Mihai Pop and Steven L Salzberg

Table 1

Table 1

Bowtie alignment performance versus SOAP and Maq

	Platform	CPU time	Wall clock time	Reads mapped per hour (millions)	Peak virtual memory footprint (megabytes)	Bowtie speed-up	Reads aligned (%)
Bowtie -v 2	Server	15 m 7 s	15 m 41 s	33.8	1,149	-	67.4
SOAP		91 h 57 m 35 s	91 h 47 m 46 s	0.10	13,619	351×	67.3
Bowtie	PC	16 m 41 s	17 m 57 s	29.5	1,353	-	71.9
Maq		17 h 46 m 35 s	17 h 53 m 7 s	0.49	804	59.8×	74.7
Bowtie	Server	17 m 58 s	18 m 26 s	28.8	1,353	-	71.9
Maq		32 h 56 m 53 s	32 h 58 m 39 s	0.27	804	107×	74.7

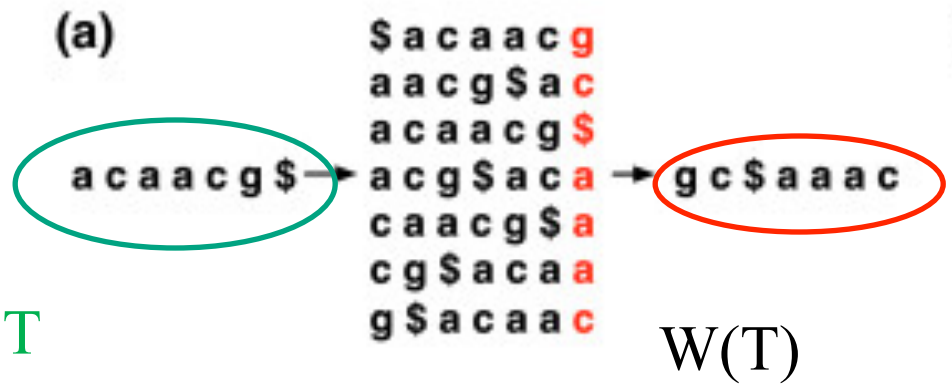
The performance and sensitivity of Bowtie v0.9.6, SOAP v1.10, and Maq v0.6.6 when aligning 8.84 M reads from the 1,000 Genome project (National Center for Biotechnology Information Short Read Archive: SRR001115) trimmed to 35 base pairs. The 'soap.contig' version of the SOAP binary was used. SOAP could not be run on the PC because SOAP's memory footprint exceeds the PC's physical memory. For the SOAP comparison, Bowtie was invoked with '-v 2' to mimic SOAP's default matching policy (which allows up to two mismatches in the alignment and disregards quality values). For the Maq comparison Bowtie is run with its default policy, which mimics Maq's default policy of allowing up to two mismatches during the first 28 bases and enforcing an overall limit of 70 on the sum of the quality values at all mismatched positions. To make Bowtie's memory footprint more comparable to Maq's, Bowtie is invoked with the '-z' option in all experiments to ensure only the forward or mirror index is resident in memory at one time. CPU, central processing unit.

Langmead *et al. Genome Biology* 2009 **10**:R25 doi:10.1186/gb-2009-10-3-r25

Burrows-Wheeler indexing

- originally developed within the context of data compression
- allows large texts to be searched efficiently in a small memory footprint.

The character $ is appended to T,where $ is not in T and is lexicographically less than all characters in T.
The Burrows-Wheeler matrix of T is constructed as the matrix whose rows comprise all cyclic rotations of T$.
The rows are then sorted lexicographically. BWT(T) is the sequence of characters in the rightmost column of the Burrows-Wheeler matrix
BWT(T) has the same length as the original text T.



This matrix has a property called 'last first (LF) mapping

The ith occurrence of character X in the last column corresponds to the same text character as the ith occurrence of X in the first.

This matrix has a property called 'last first (LF) mapping

The ith occurrence of character X in the last column corresponds to the same text character as the ith occurrence of X in the first.

UNPERMUTE repeatedly applies the last first (LF) mapping to recover the original text (in red on the top line) from the Burrows-Wheeler transform (in black in the rightmost column).

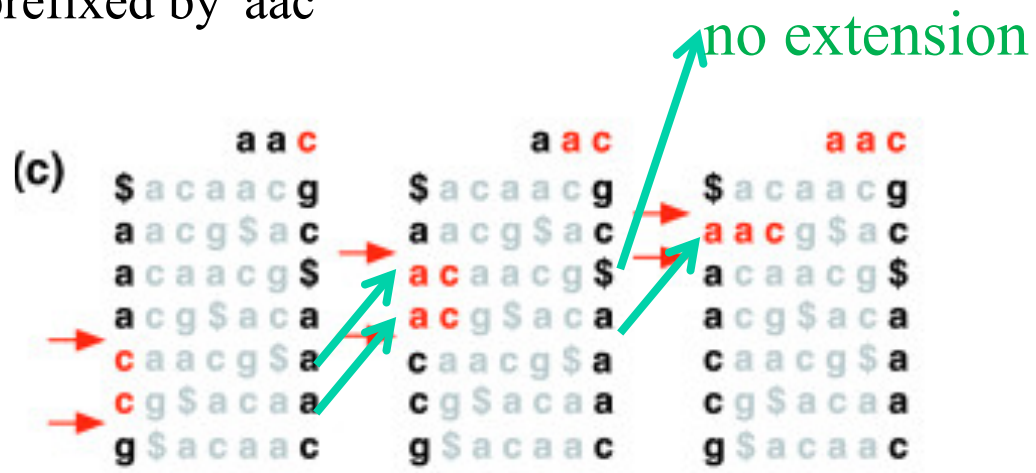
EXACTMATCH algorithm calculates the range of matrix rows beginning with successively longer suffixes of the query.

Because the matrix is sorted lexicographically, rows beginning with a given sequence appear consecutively.

At each step of EXACTMATCH, the size of the range either shrinks or remains the same. When the algorithm completes, rows beginning with S0 (the entire query) correspond to exact occurrences of the query in the text.

If the range is empty, the text does not contain the query.

Below steps taken by EXACTMATCH to identify the range of rows, and thus the set of reference suffixes, prefixed by 'aac'



Each character in a read has a numeric quality value, with lower values indicating a higher likelihood of a sequencing error. Bowtie alignment policy allows a limited number of mismatches and prefers (no guarantees) alignments where the sum of the quality values at all mismatched positions is low.

There is no exact match for query 'ggta' but there is a one-mismatch alignment when 'a' is replaced by 'g'. Boxed pairs of numbers denote ranges of matrix rows beginning with the suffix observed up to that point. A red X marks where the algorithm encounters an empty range and either aborts (as in EXACTMATCH) or backtracks (as in the inexact algorithm). A green check marks where the algorithm finds a nonempty range delimiting one or more occurrences of a reportable alignment for the query.

Tricks and heuristics

If only one error is allowed once can build second index for alignment in the reverse direction and such two error free alignments would “meet” at the position of the error

The is there is error-free prefix + error + error-free postfix

This doesn’t work for more than one error and Bowties finds the best solution it can do in the k = 200 steps (a parameter of the alignment, estimated experimentally) and gives up.

Index building

- Kärkkäinen J: **Fast BWT in small space by blockwise suffix sorting.** *Theor Comput Sci* 2007, **387:**249-257.

Bowtie index building performance

Physical memory target (GB)	Actual peak memory footprint (GB)	Wall clock time
16	14.4	4 h 36 m
8	5.84	5 h 5 m
4	3.39	7 h 40 m
2	1.39	21 h 30 m

Performance results and memory footprints of running the Bowtie v0.9.6 indexer on the whole human genome (National Center for Biotechnology Information build 36.3, contigs). Runs were performed on the server platform. The indexer was run four times with different upper limits on memory usage. See Additional data file 1 (Supplementary Discussion 3 and Supplementary Table 1) for details.

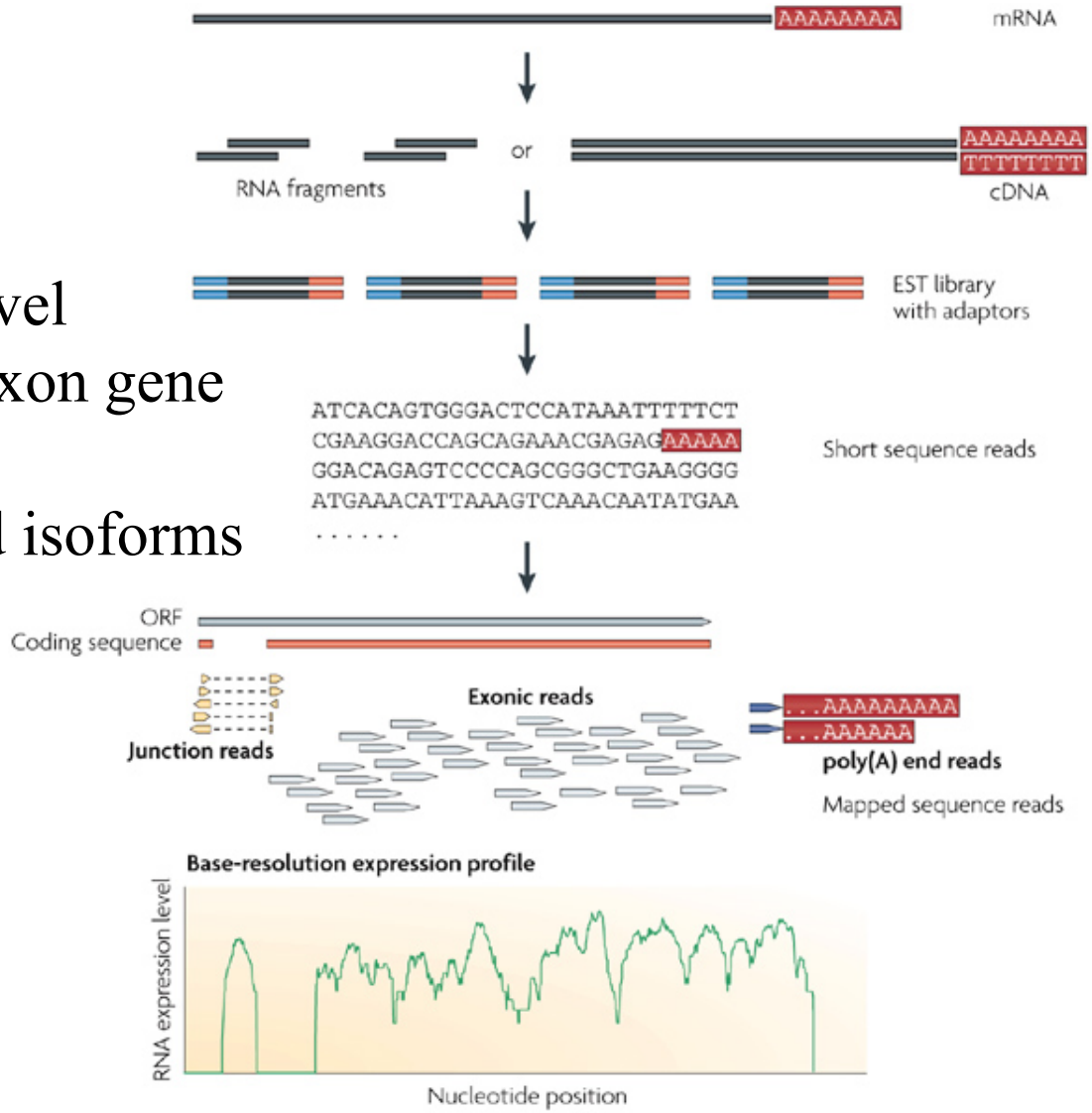
Langmead *et al. Genome Biology* 2009 **10**:R25 doi:10.1186/gb-2009-10-3-r25

Other algorithms that use BW transform

- Li H, Durbin R (2009). "Fast and accurate short read alignment with Burrows–Wheeler Transform". *Bioinformatics*. PMID 19451168.
- Li R, et al (2009). "SOAP2: an improved ultrafast tool for short read alignment". *Bioinformatics*. PMID 19497933.

RNA-Seq provides means to measure transcriptome data experimentally,

Applications:
-Measuring mRNA level
- Uncovering intron-exon gene structure
-Detecting transcribed isoforms



RNA-Seq: a revolutionary tool for transcriptomics
Zhong Wang, Mark Gerstein & Michael Snyder; Jan 2009

Nature Reviews | Genetics