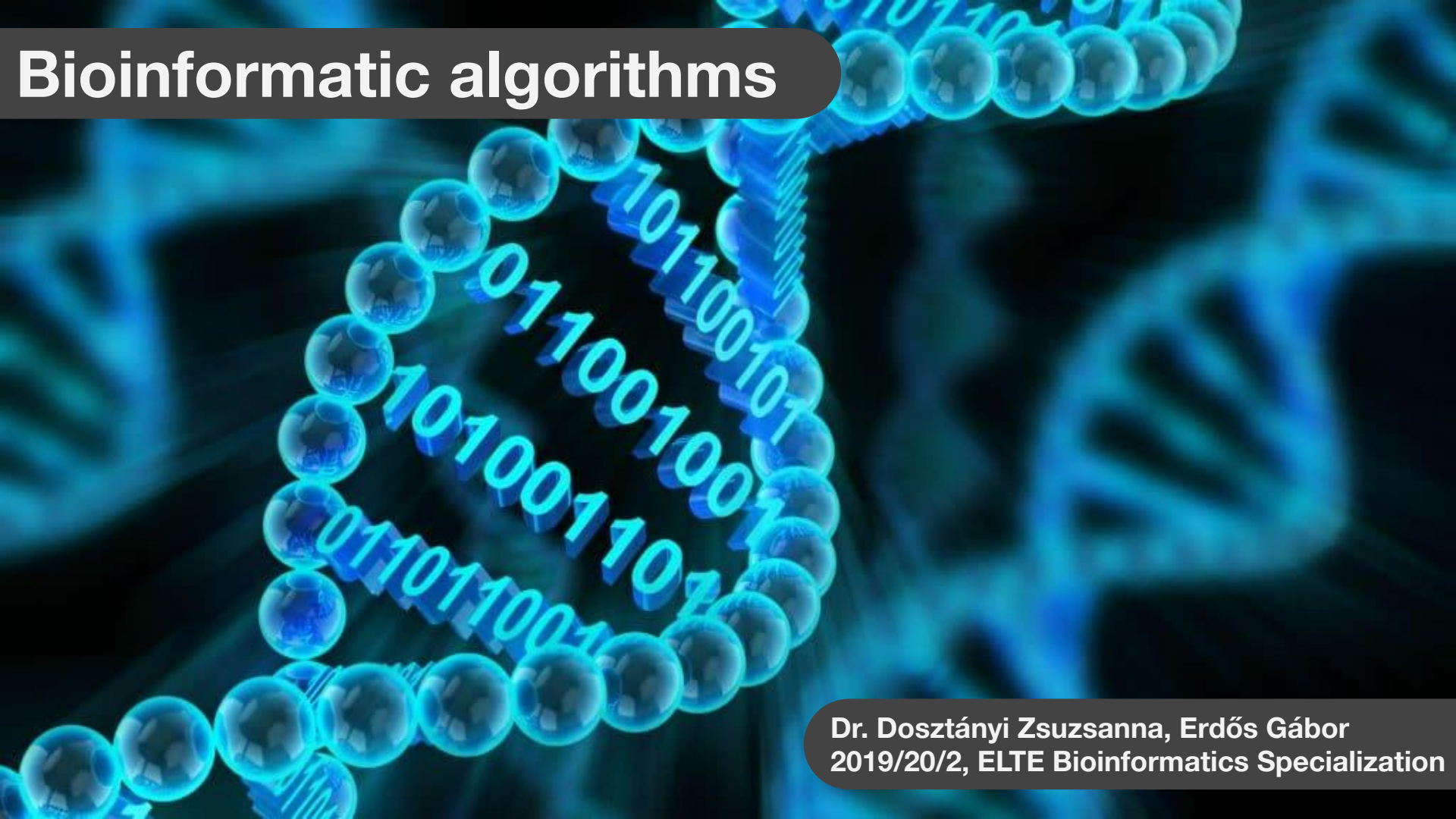
Bioinformatic algorithms



Dr. Dosztányi Zsuzsanna, Erdős Gábor
2019/20/2, ELTE Bioinformatics Specialization

Transcription factors

Now we can find the origin of replication, and the region where the polymerase likely starts the replication. However there are multiple other “hidden messages” in the DNA. For example a transcription factor called NF-kB that activates various immunity genes in flies. We dont expect the region where to transcription factor binds to appear multiple times in short segments, however we expect it to appear frequently throughout the whole genom.

```
1  "atgaccgggatactgatAAAAAAAAGGGGGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg"
2  "acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataAAAAAAAAGGGGGGGa"
3  "tgagtatccctgggatgacttAAAAAAAAGGGGGGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga"
4  "gctgagaattggatgAAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga"
5  "tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAAGGGGGGGcttatag"
6  "gtcaatcatgttcttgtgaatggatttAAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa"
7  "cggttttggcccttgttagaggccccgtAAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat"
8  "aacttgagttAAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta"
9  "ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAAGGGGGGGaccgaaagggaag"
10 "ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAAGGGGGGGa"
```

Transcription factors

Now we can find the origin of replication, and the region where the polymerase likely starts the replication. However there are multiple other “hidden messages” in the DNA. For example a transcription factor called NF-kB that activates various immunity genes in flies. We dont expect the region where to transcription factor binds to appear multiple times in short segments, however we expect it to appear frequently throughout the whole genom.

```
1  "atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg"
2  "acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa"
3  "tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga"
4  "gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga"
5  "tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag"
6  "gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa"
7  "cggttttggcccttgttagaggccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat"
8  "aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta"
9  "ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag"
10 "ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa"
```

A brute force algorithm

Given a collection of strings *Dna* and an integer *d*, a *k*-mer is a **(*k*,*d*)-motif** if it appears in every string from *Dna* with at most *d* mismatches. For example, the implanted 15-mer in the strings above represents a (15,4)-motif.

Implanted Motif Problem: *Find all* (*k*, *d*)*-motifs in a collection of strings.*

- **Input:** A collection of strings *Dna*, and integers *k* and *d*.
- **Output:** All (*k*, *d*)-motifs in *Dna*.

Brute force (also known as **exhaustive search**) is a general problem-solving technique that explores all possible solution candidates and checks whether each candidate solves the problem. Such algorithms require little effort to design and are guaranteed to produce a correct solution, but they may take an enormous amount of time, and the number of candidates may be too large to check.

The Median String Problem

Consider **AgAAgAAAGGttGGG** and **cAAtAAAAcGGGGcG**, each of which differs from **AAAAAAAAGGGGGGG** by four mismatches. Although these 15-mers look similar to the correct motif **AAAAAAAAGGGGGGG**, they are not so similar when compared to each other, having eight mismatches.

A more appropriate problem formulation would score individual instances of motifs depending on how similar they are to an "ideal" motif.

Given a *k*-mer *Pattern* and a longer string *Text*, we use *d*(*Pattern*, *Text*) to denote the minimum Hamming distance between *Pattern* and any *k*-mer in *Text*.

$$d(Pattern, Text) = \min_{\text{all k-mers Pattern' in Text}} HammingDistance(Pattern, Pattern')$$

A *k*-mer in *Text* that achieves the minimum Hamming distance with *Pattern* is denoted *Motif*(*Pattern*, *Text*).

Motif(GATTCTCA, GCAAAGACGCTGACCAA) = GACGCTGA.

The Median String Problem

Given a *k*-mer *Pattern* and a set of strings *Dna* = {Dna_1, … , Dna_t}, we define *d*(*Pattern*, *Dna*) as the sum of distances between *Pattern* and all strings in *Dna*

$$d(Pattern, Dna) = \sum_{i=1}^{t} d(Pattern, Dna_i).$$

For example, for the strings *Dna* shown below, *d*(AAA, *Dna*) = 1 + 1 + 2 + 0 + 1 = 5.

Dna

ttaccttAAC	1
gATAtctgtc	1
ACGgcgttcg	2
ccctAAAgag	0
cgtcAGAggt	1

Median String Problem: *Find a median string.*

- **Input:** A collection of strings *Dna* and an integer *k*.
- **Output:** A *k*-mer *Pattern* that minimizes *d*(*Pattern*, *Dna*) among all possible choices of *k*-mers

Profile matrices

Another option to describe a motif is to create a so called profile matrix or position specific scoring matrix (PSSM). A profile describes how likely is for a given nucleotide (or amino acid) to appear in a given position.

We can use the profile to “score” new motifs.

Profile matrix: *Create a profile matrix from a collection of motifs*

- **Input:** A collection of strings *Dna*
- **Output:** A profile matrix

Motifs

	T	C	G	G	G	G	g	T	T	T	t	t
	c	C	G	G	t	G	A	c	T	T	a	C
	a	C	G	G	G	G	A	T	T	T	t	C
	T	t	G	G	G	G	A	c	T	T	t	t
	a	a	G	G	G	G	A	c	T	T	C	C
	T	t	G	G	G	G	A	c	T	T	C	C
	T	C	G	G	G	G	A	T	T	c	a	t
	T	C	G	G	G	G	A	T	T	c	C	t
	T	a	G	G	G	G	A	a	c	T	a	C
	T	C	G	G	G	t	A	T	a	a	C	C

Score(Motifs) $3 + 4 + 0 + 0 + 1 + 1 + 1 + 5 + 2 + 3 + 6 + 4 = 30$

Count(Motifs)

A:	2	2	0	0	0	0	9	1	1	1	3	0
C:	1	6	0	0	0	0	0	4	1	2	4	6
G:	0	0	10	10	9	9	1	0	0	0	0	0
T:	7	2	0	0	1	1	0	5	8	7	3	4

Profile(Motifs)

A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4

Consensus(Motifs) T C G G G G A T T T C C

Profile matrices

Lets consider the second and the last column. They both contribute 4 to the final score.

Do they have equal contribution?

Entropy is a measure of the uncertainty of a probability distribution ($p1$, ..., pN), and is defined as follows:

$$H(p_1, \ldots, p_N) = -\sum_{i=1}^{N} p_i \cdot \log_2 p_i$$

Note: Technically, $\log_2(0)$ is undefined, but in the computation of entropy, we assume that $0 \cdot \log_2(0)$ is equal to 0.

Calculate the entropy of each column

Motifs	T	C	G	G	G	G	g	T	T	T	t	t
	c	C	G	G	t	G	A	c	T	T	a	C
	a	C	G	G	G	G	A	T	T	T	t	C
	T	t	G	G	G	G	A	c	T	T	t	t
	a	a	G	G	G	G	A	c	T	T	C	C
	T	t	G	G	G	G	A	c	T	T	C	C
	T	C	G	G	G	G	A	T	T	c	a	t
	T	C	G	G	G	G	A	T	T	c	C	t
	T	a	G	G	G	G	A	a	c	T	a	C
	T	C	G	G	G	t	A	T	a	a	C	C
Score(Motifs)	3	+ 4	+ 0	+ 0	+ 1	+ 1	+ 1	+ 5	+ 2	+ 3	+ 6	+ 4 = 30
Count(Motifs)	A:	2	2	0	0	0	0	9	1	1	1	3	0
	C:	1	6	0	0	0	0	0	4	1	2	4	6
	G:	0	0	10	10	9	9	1	0	0	0	0	0
	T:	7	2	0	0	1	1	0	5	8	7	3	4
Profile(Motifs)	A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
	C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
	G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
	T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4
Consensus(Motifs)	T	C	G	G	G	G	A	T	T	T	C	C

Profile matrices

Given a profile matrix *Profile*, we can evaluate the probability of every *k*-mer in a string *Text* and find a ***Profile*-most probable** *k*-mer in *Text*, i.e., a *k*-mer that was most likely to have been generated by *Profile* among all *k*-mers in *Text*. For example, **ACGGGGATTACC** is the *Profile*-most probable 12-mer in GGT**ACGGGGATTACC**T. Indeed, every other 12-mer in this string has probability 0. In general, if there are multiple *Profile*-most probable *k*-mers in *Text*, then we select the first such *k*-mer occurring in *Text*.

***Profile*-most Probable *k*-mer Problem:** *Find a Profile-most probable k-mer in a string.*

- **Input:** A string *Text*, an integer *k*, and a 4 × *k* matrix *Profile*.
- **Output:** A *Profile*-most probable *k*-mer in *Text*.

Motifs	T	C	G	G	G	G	g	T	T	T	t	t
	c	C	G	G	t	G	A	c	T	T	a	C
	a	C	G	G	G	G	A	T	T	T	t	C
	T	t	G	G	G	G	A	c	T	T	t	t
	a	a	G	G	G	G	A	c	T	T	C	C
	T	t	G	G	G	G	A	c	T	T	C	C
	T	C	G	G	G	G	A	T	T	c	a	t
	T	C	G	G	G	G	A	T	T	c	C	t
	T	a	G	G	G	G	A	a	c	T	a	C
	T	C	G	G	G	t	A	T	a	a	C	C
Score(Motifs)	3	+ 4	+ 0	+ 0	+ 1	+ 1	+ 1	+ 5	+ 2	+ 3	+ 6	+ 4 = 30
Count(Motifs) A:	2	2	0	0	0	0	9	1	1	1	3	0
C:	1	6	0	0	0	0	0	4	1	2	4	6
G:	0	0	10	10	9	9	1	0	0	0	0	0
T:	7	2	0	0	1	1	0	5	8	7	3	4
Profile(Motifs) A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4
Consensus(Motifs)	T	C	G	G	G	G	A	T	T	T	C	C