University of Antwerp

DNA Motif search using Genetic Algorithms

Hura Viktor

Madmar Mounir

Darkaoui Mohamed

Co-regulation

```
> Gene 1 Promoter
CAAAACCCTCAAATACATTTTAGAAACACAATTTCAGGATATAAAAAGTTAAATTCATCTAGTTATACAA

> Gene 2 Promoter
TCTTTTCTGAATCTGTATAAAAACTTTTATTCTGTAGATGGTGGCTGTAGGAATCTGTCACACAGCATGA

> Gene 3 Promoter
CCACGTGGTTAGTGGCAACCTGGTGACCCCCCTTCCTGTGATTTTTATAAAAAGAGCAGCCGGCATCGTT

> Gene 4 Promoter
GGAGAGTGTTTATAAAAAGATGACTACAGTCAAACCAGGTACAGGATTCACACTCAGGGAACACGTGTGG

> Gene 5 Promoter
TCACCATCAAACCTGAATCAAGGCAATGAGCAGGTATAAAAAGCCTGGATAAGGAAACCAAGGCAATGAG
```

DNA Motif

All genes that have the motif in their sequence will be regulated by the same transcription factor.
Often this means that they will have similar expression patterns.

Degeneracy

```
> Gene 1 Promoter
CAAAACCCTCAAATACATTTTAGAAACACAATTTCAGGATATTAAAAGTTAAATTCATCTAGTTATACAA

> Gene 2 Promoter
TCTTTTCTGAATCTGAATAAATACTTTTATTCTGTAGATGGTGGCTGTAGGAATCTGTCACACAGCATGA

> Gene 3 Promoter
CCACGTGGTTAGTGGCAACCTGGTGACCCCCCTTCCTGTGATTTTTACAAATAGAGCAGCCGGCATCGTT

> Gene 4 Promoter
GGAGAGTGTTTTTAAGAAGATGACTACAGTCAAACCAGGTACAGGATTCACACTCAGGGAACACGTGTGG

> Gene 5 Promoter
TCACCATCAAACCTGAATCAAGGCAATGAGCAGGTATACATAGCCTGGATAAGGAAACCAAGGCAATGAG
```

In many cases, the DNA signal is not absolute but some error tolerance is allowed.
How do we represent the (common) motif sequence?
How do we decide what is a match and what isn't?

Gibbs motif sampling

Optimize starting from a random start in n-1 sequences

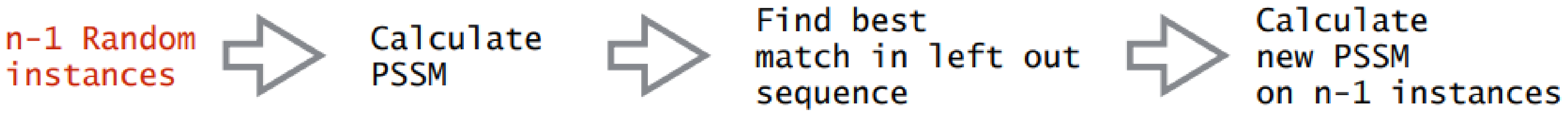
```
> Gene 1 Promoter
CAAAACCCTCAAATACATTTTAGAAACACAATTTCAGGATATTAAAAGTTAAATTCATCTAGTTATACAA

> Gene 2 Promoter
TCTTTTCTGAATCTGAATAAATACTTTTATTCTGTAGATGGTGGCTGTAGGAATCTGTCACACAGCATGA

> Gene 3 Promoter
CCACGTGGTTAGTGGCAACCTGGTGACCCCCCTTCCTGTGATTTTTACAAATAGAGCAGCCGGCATCGTT

> Gene 4 Promoter
GGAGAGTGTTTTTAAGAAGATGACTACAGTCAAACCAGGTACAGGATTCACACTCAGGGAACACGTGTGG

> Gene 5 Promoter
TCACCATCAAACCTGAATCAAGGCAATGAGCAGGTATACATAGCCTGGATAAGGAAACCAAGGCAATGAG
```



n-1 Random instances

Calculate PSSM

Find best match in left out sequence

Calculate new PSSM on n-1 instances

Recursion until optimum is reached

Consensus sequence

Known instances but unknown motif

Known instances	TATTAAAA
	AATAAATA
	TACAAATA
	TTTAAGAA
	TATACATA
Motif	TATAAATA

List the most frequent base at each position

```
                                       TATAAAAA
CAAAACCCTCAAATACATTTTAGAAACACAATTTCAGGATATTAAAAGTTAAATTCATCTAGTTATACAA
```

```
               TATAAAAA
TCTTTTCTGAATCTGAATAAATACTTTTATTCTGTAGATGGTGGCTGTAGGAATCTGTCACACAGCAA
```

```
                                           TATAAAAA
CCACGTGGTTAGTGGCAACCTGGTGACCCCCCTTCCTGTGATTTTTACAAATAGAGCAGCCGGGTT
```

Motif

Genome [] =
list of letters

Example

Genome [] = [‘A’, ‘C’, ‘G’, ‘C’, ‘A’, ‘T’]

Mutation

Motif

Genome [] = [‘A’, ‘G’, ‘G’, ‘C’, ‘A’, ‘T’]

Each gene has a {mutationRate} % chance to mutate

Crossover

P1

Genome [] = [‘A’, ‘C’, ‘G’, ‘C’, ‘A’, ‘T’]

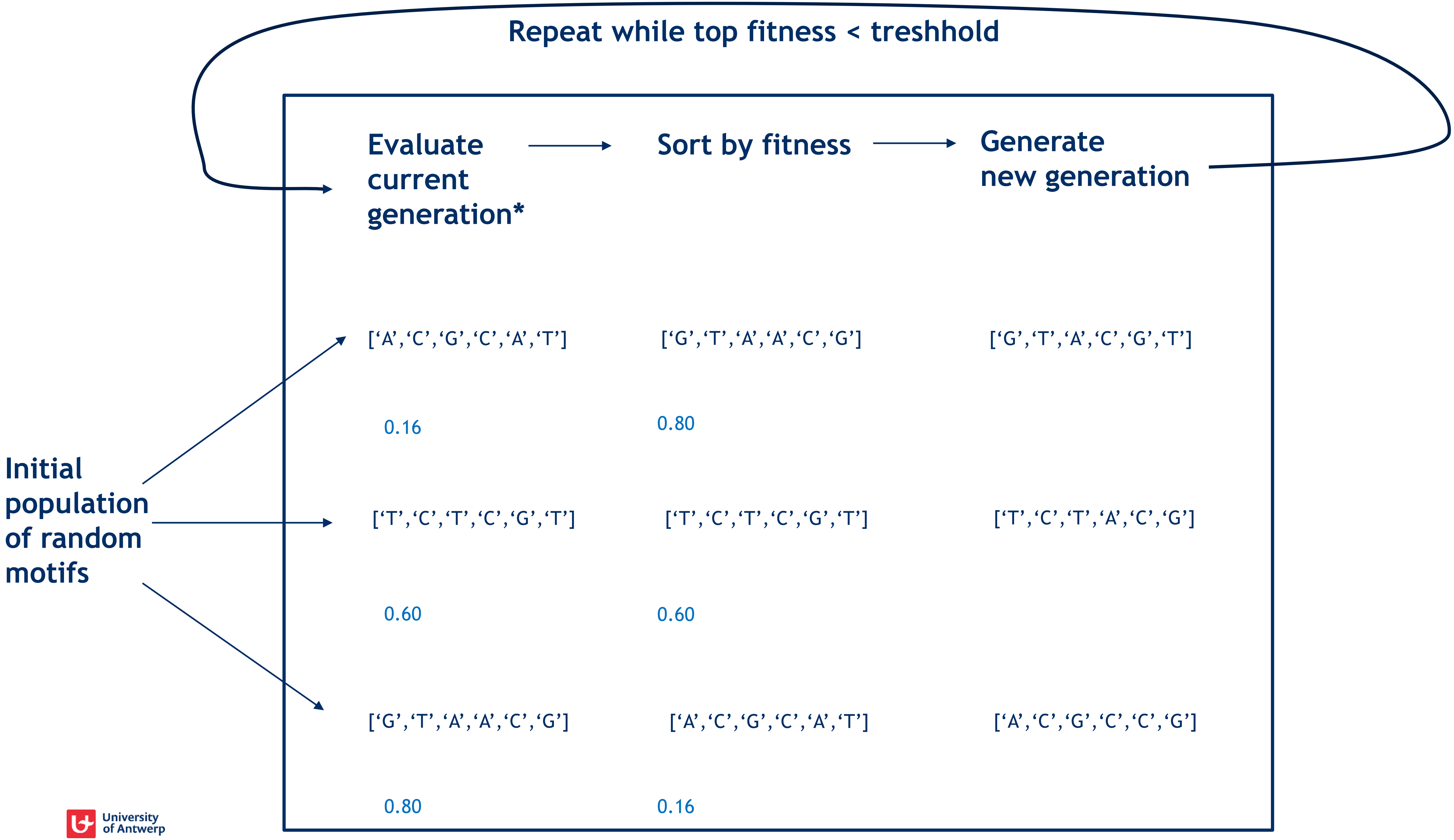
P2

Genome [] = [‘T’, ‘G’, ‘T’, ‘A’, ‘A’, ‘G’]

Midpoint(random) = 2

Child

Genome [] = [‘A’, ‘C’, ‘T’, ‘A’, ‘A’, ‘G’]



Repeat while top fitness < treshhold

Initial population of random motifs

Evaluate current generation*	Sort by fitness	Generate new generation
[‘A’,‘C’,‘G’,‘C’,‘A’,‘T’]	[‘G’,‘T’,‘A’,‘A’,‘C’,‘G’]	[‘G’,‘T’,‘A’,‘C’,‘G’,‘T’]
0.16	0.80	
[‘T’,‘C’,‘T’,‘C’,‘G’,‘T’]	[‘T’,‘C’,‘T’,‘C’,‘G’,‘T’]	[‘T’,‘C’,‘T’,‘A’,‘C’,‘G’]
0.60	0.60	
[‘G’,‘T’,‘A’,‘A’,‘C’,‘G’]	[‘A’,‘C’,‘G’,‘C’,‘A’,‘T’]	[‘A’,‘C’,‘G’,‘C’,‘C’,‘G’]
0.80	0.16	

TATAAAAA 7/8

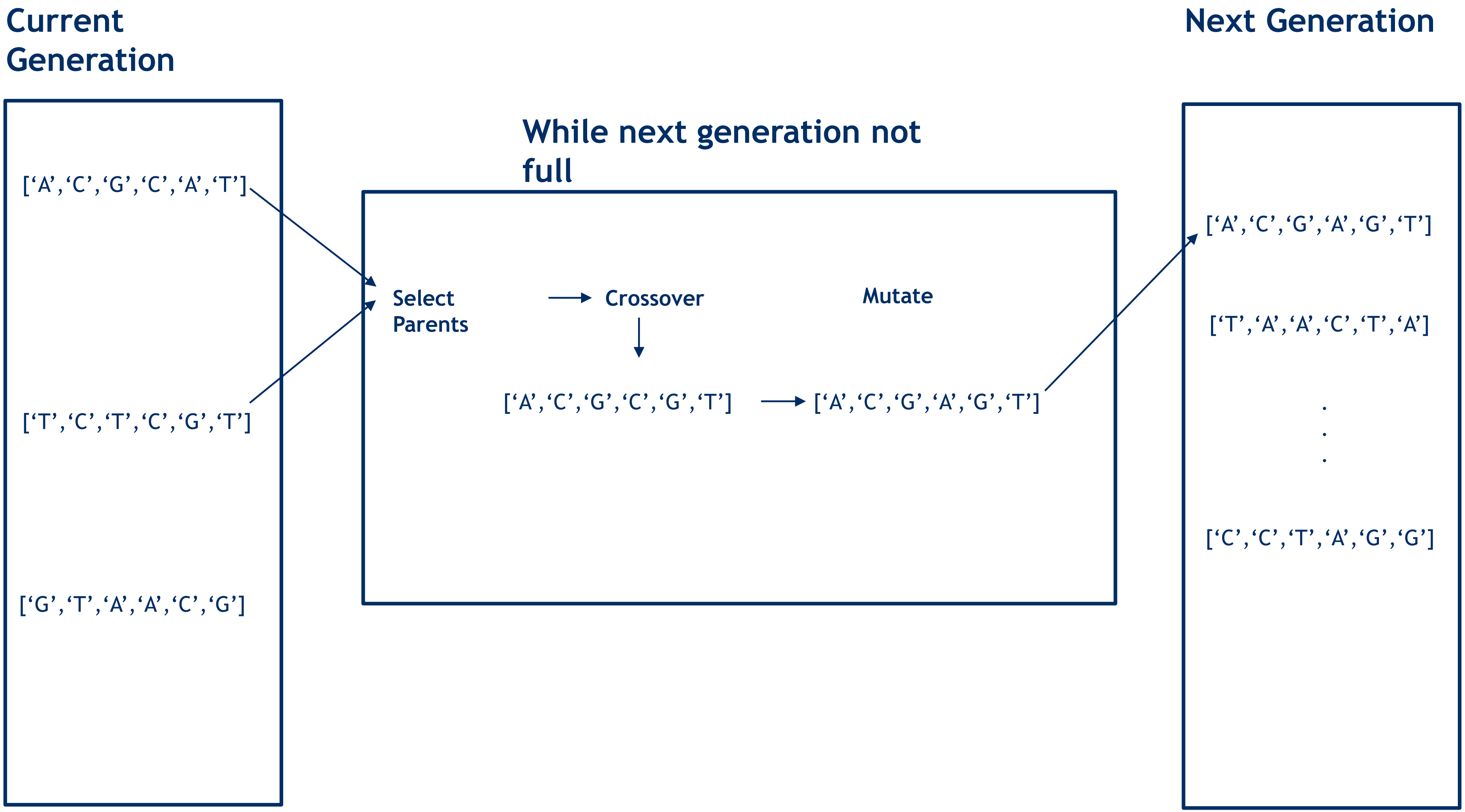
CAAAACCCTCAAATACATTTTAGAAACACAATTTCAGGATATTAAAAGTTAAATTCATCTAGTTATACAA

TATAAAAA 6/8

TCTTTTCTGAATCTGAATAAATACTTTTATTCTGTAGATGGTGGCTGTAGGAATCTGTCACACAGCAA

TATAAAAA 6/8

CCACGTGGTTAGTGGCAACCTGGTGACCCCCCTTCCTGTGATTTTTACAAATAGAGCAGCCGGGTT



Current Generation

['A','C','G','C','A','T']

['T','C','T','C','G','T']

['G','T','A','A','C','G']

While next generation not full

Select Parents → Crossover → ['A','C','G','C','G','T'] → ['A','C','G','A','G','T']

Mutate

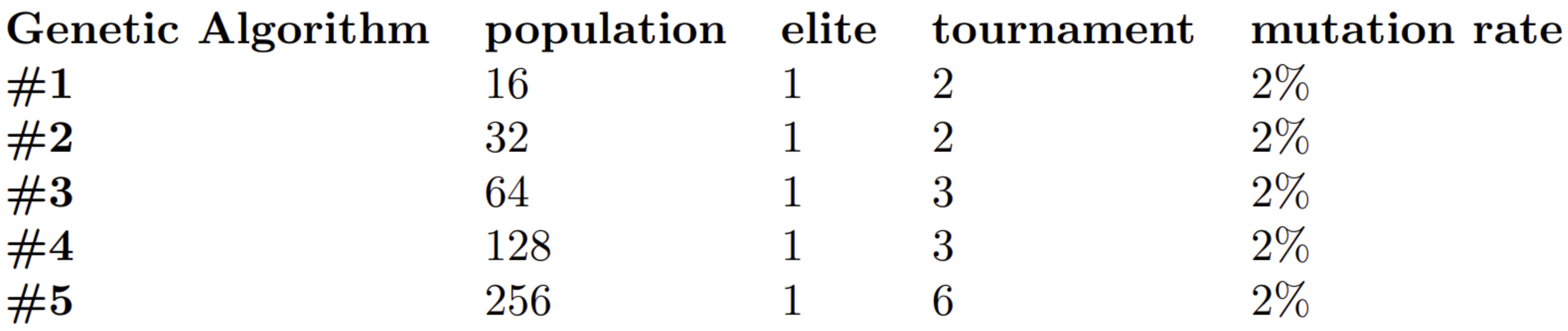
Next Generation

['A','C','G','A','G','T']

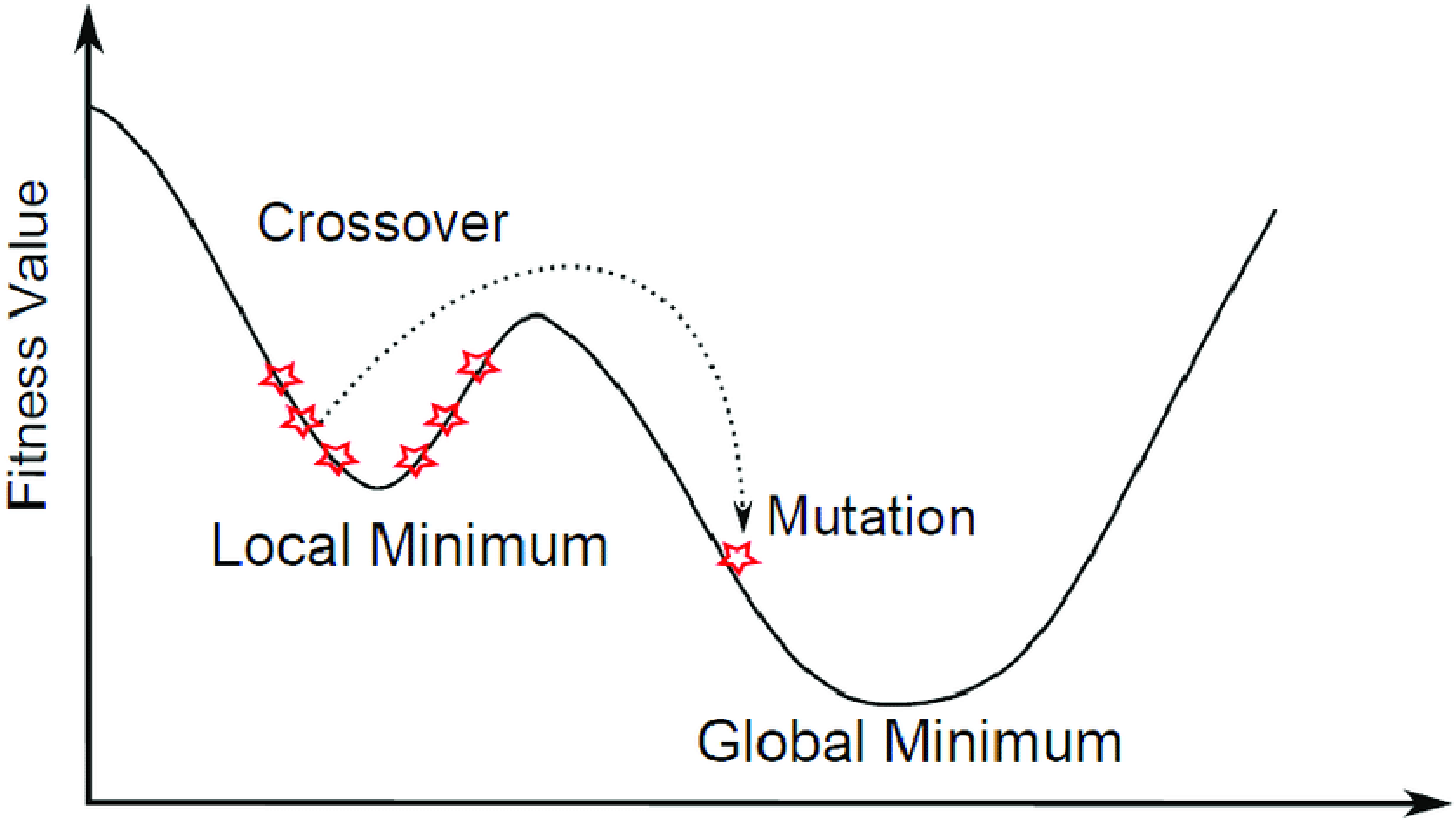
['T','A','A','C','T','A']

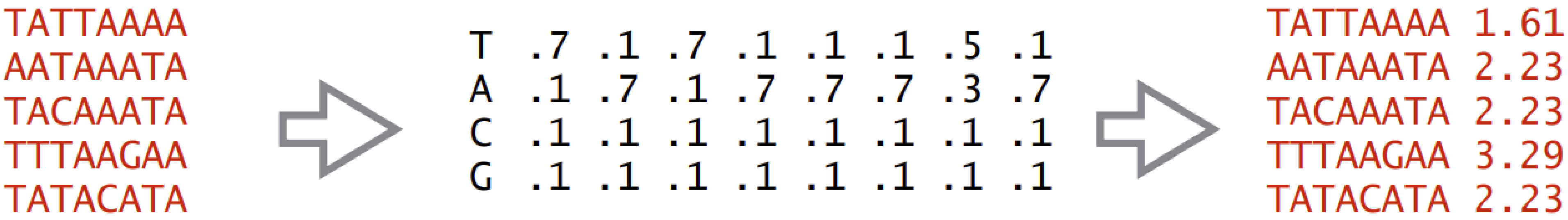
.
.
.

['C','C','T','A','G','G']



Genetic Algorithm	**population**	**elite**	**tournament**	**mutation rate**
#1	16	1	2	2%
#2	32	1	2	2%
#3	64	1	3	2%
#4	128	1	3	2%
#5	256	1	6	2%


Fitness Value
Crossover
Mutation
Local Minimum
Global Minimum

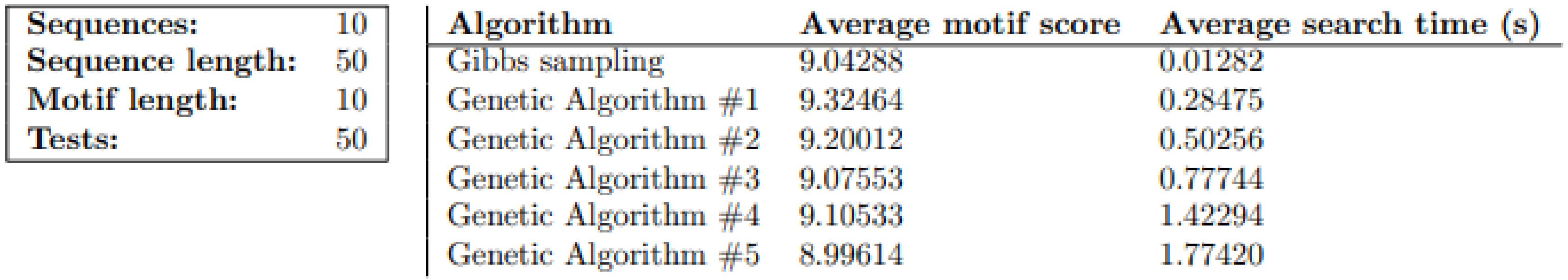


TATTAAAA
AATAAATA
TACAAATA
TTTAAGAA
TATACATA

T	.7	.1	.7	.1	.1	.1	.5	.1
A	.1	.7	.1	.7	.7	.7	.3	.7
C	.1	.1	.1	.1	.1	.1	.1	.1
G	.1	.1	.1	.1	.1	.1	.1	.1

TATTAAAA	1.61
AATAAATA	2.23
TACAAATA	2.23
TTTAAGAA	3.29
TATACATA	2.23

Table 2: Results with less sequences and lower sequence length.



Sequences:	10
Sequence length:	50
Motif length:	10
Tests:	50

Algorithm	**Average motif score**	**Average search time (s)**
Gibbs sampling	9.04288	0.01282
Genetic Algorithm #1	9.32464	0.28475
Genetic Algorithm #2	9.20012	0.50256
Genetic Algorithm #3	9.07553	0.77744
Genetic Algorithm #4	9.10533	1.42294
Genetic Algorithm #5	8.99614	1.77420

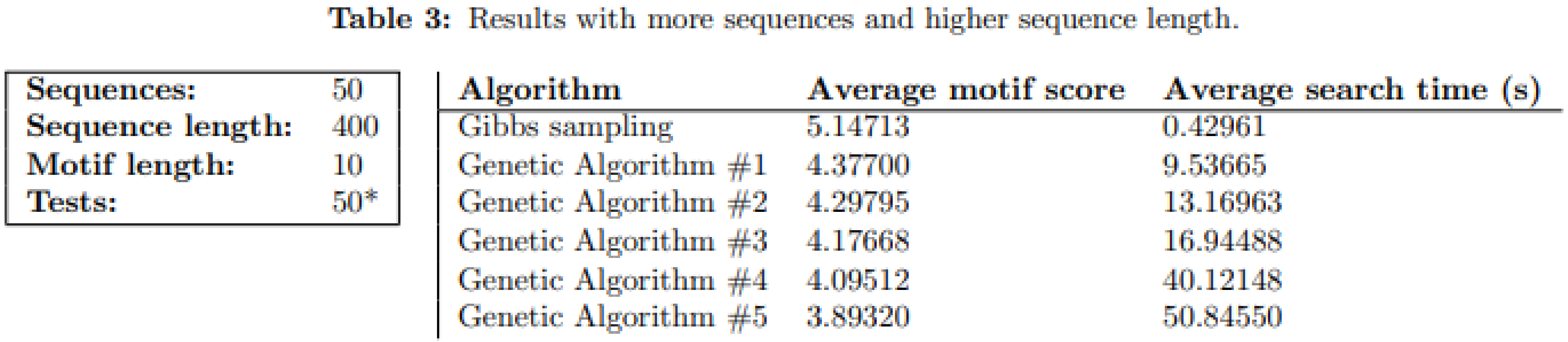


Table 3: Results with more sequences and higher sequence length.

Sequences:	50
Sequence length:	400
Motif length:	10
Tests:	50*

Algorithm	**Average motif score**	**Average search time (s)**
Gibbs sampling	5.14713	0.42961
Genetic Algorithm #1	4.37700	9.53665
Genetic Algorithm #2	4.29795	13.16963
Genetic Algorithm #3	4.17668	16.94488
Genetic Algorithm #4	4.09512	40.12148
Genetic Algorithm #5	3.89320	50.84550

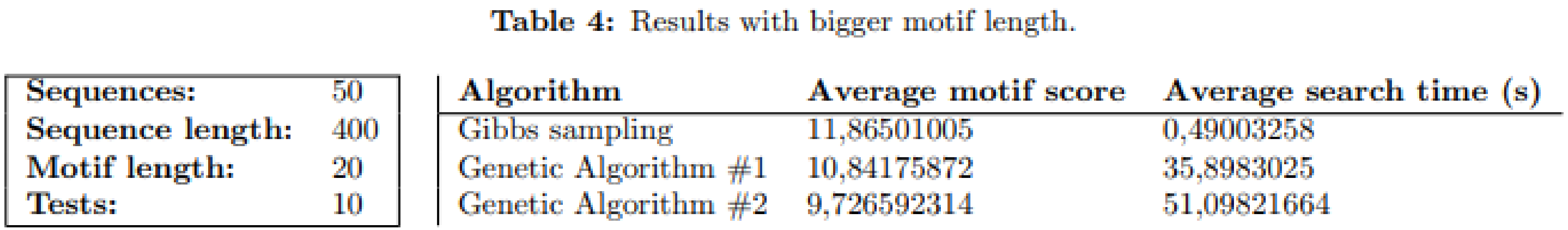


Table 4: Results with bigger motif length.

Sequences:	50
Sequence length:	400
Motif length:	20
Tests:	10

Algorithm	**Average motif score**	**Average search time (s)**
Gibbs sampling	11,86501005	0,49003258
Genetic Algorithm #1	10,84175872	35,8983025
Genetic Algorithm #2	9,726592314	51,09821664

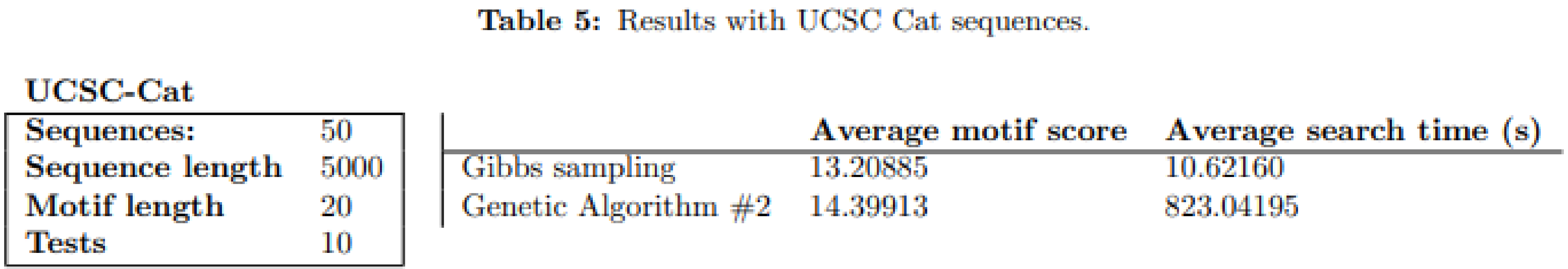


Table 5: Results with UCSC Cat sequences.

UCSC-Cat

Sequences:	50
Sequence length	5000
Motif length	20
Tests	10

	Average motif score	**Average search time (s)**
Gibbs sampling	13.20885	10.62160
Genetic Algorithm #2	14.39913	823.04195

Conclusies