Algorithms in Bioinformatics Spring 2023 Lecture 6

Jason Ernst

University of California, Los Angeles

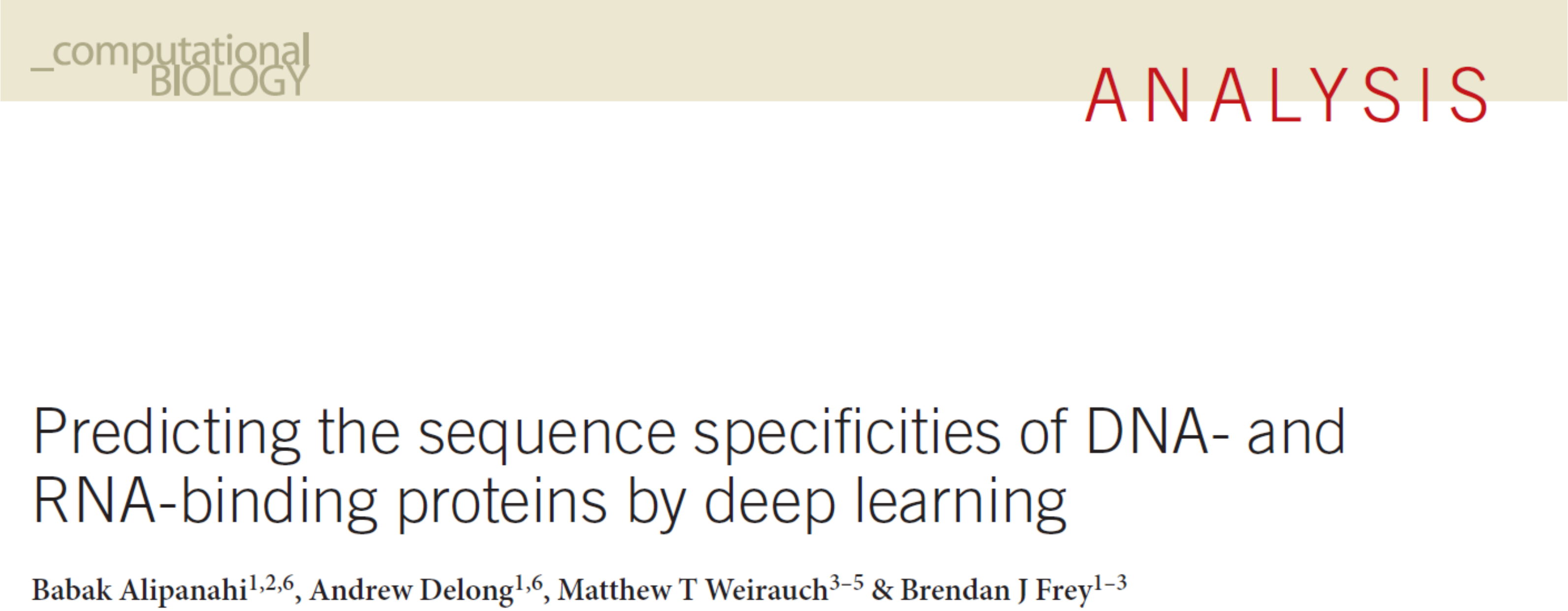
Announcements

- HW3 - chapter 8 due 4/25
- Project 1a - due 4/27
- Discussion sections Friday - focus will be on chapter 8

Announcements

- HW3 - chapter 8 due 4/25
- Project 1a - due 4/27
- Discussion sections Friday - focus will be on chapter 8

Paper 3



_computational BIOLOGY

ANALYSIS

Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning

Babak Alipanahi[1,2,6], Andrew Delong[1,6], Matthew T Weirauch[3–5] & Brendan J Frey[1–3]

Focus on first p.1-13 of supplementary note for computational methods details

Question due Thur 5/4
Responses due Tue 5/9

Motifs

Lecture 6

April 20th, 2023

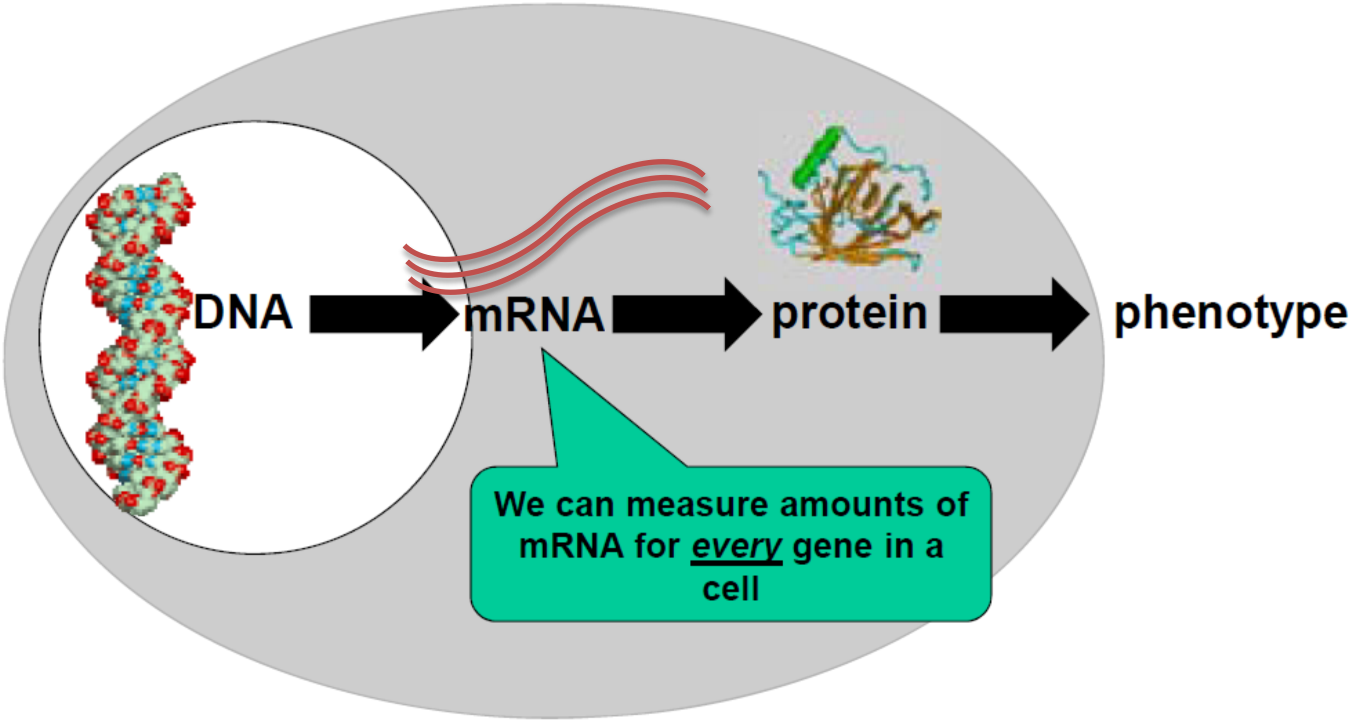
Topics

- Motif background and representations
- De novo motif discovery

Topics

- Motif background and representations
- De novo motif discovery

Central Dogma



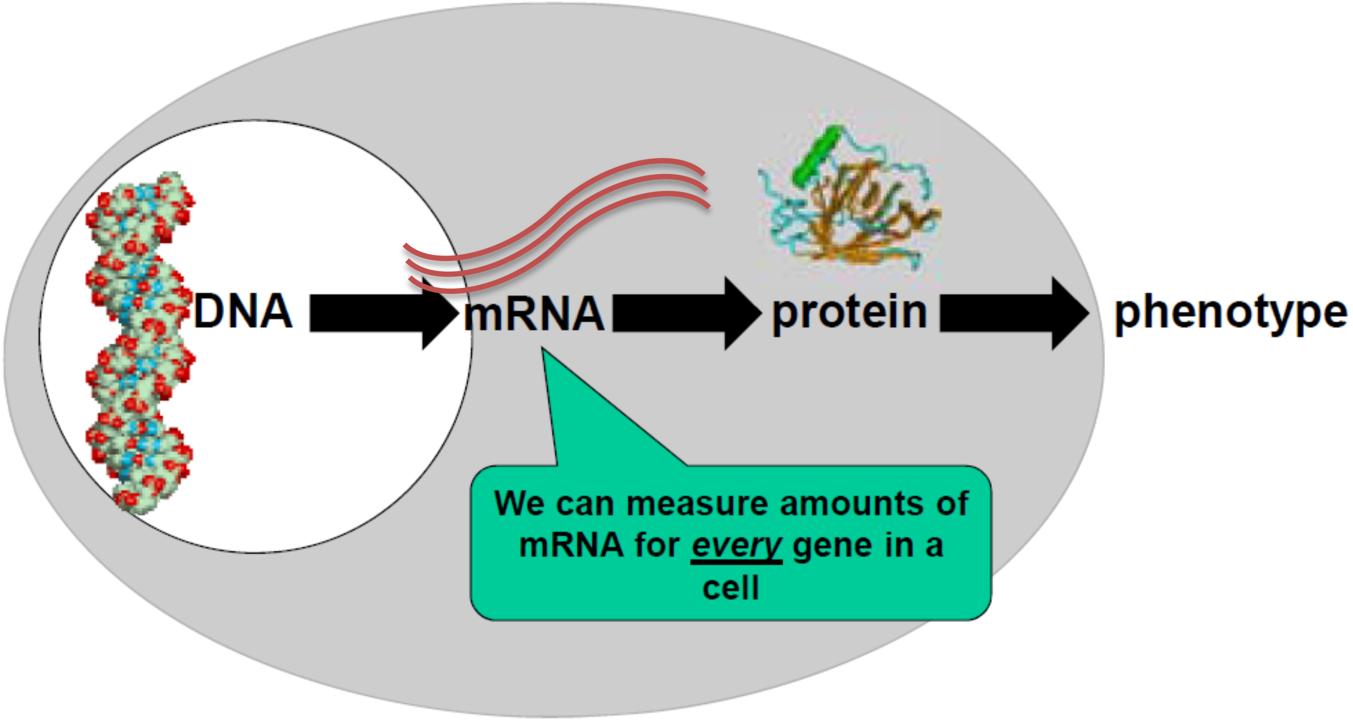
DNA → mRNA → protein → phenotype

We can measure amounts of mRNA for *every* gene in a cell

Image adapted from Manolis Kellis

Central Dogma

- The cell needs to regulate the process of going from DNA to mRNA.

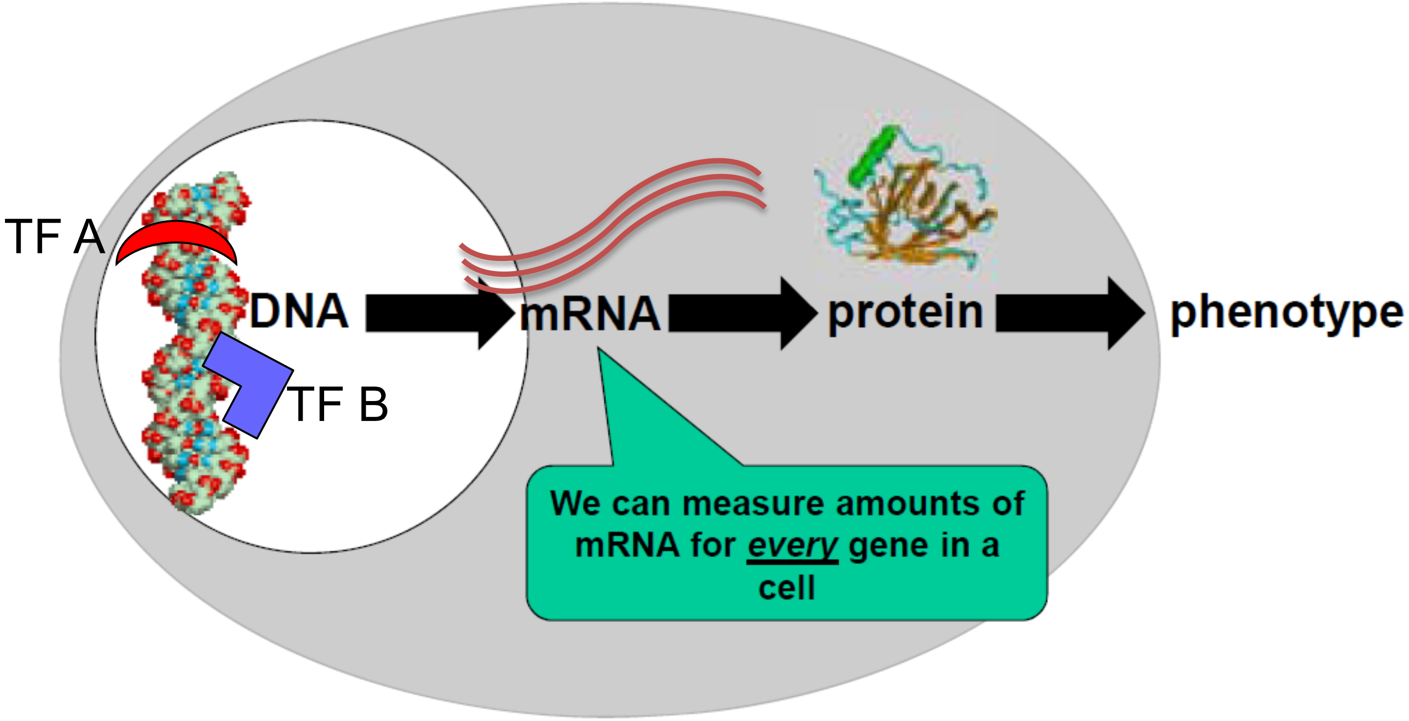


We can measure amounts of mRNA for ***every*** gene in a cell

Image adapted from Manolis Kellis

Central Dogma

- The cell needs to regulate the process of going from DNA to mRNA.
- Transcription factors (TFs) binding DNA play a major role in controlling this process to activate or repress gene expression. Thousands of TFs in human.

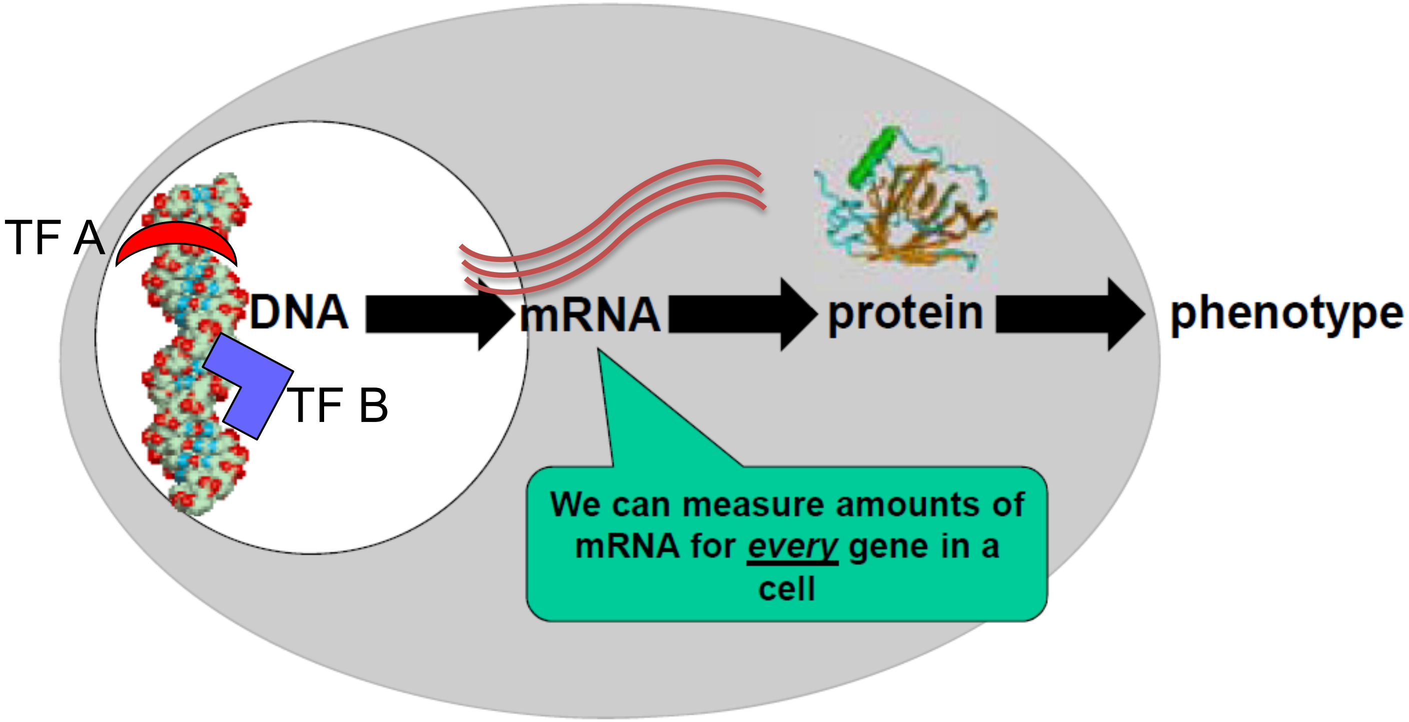


We can measure amounts of mRNA for ***every*** gene in a cell

Image adapted from Manolis Kellis

Central Dogma

- The cell needs to regulate the process of going from DNA to mRNA.
- Transcription factors (TFs) binding DNA play a major role in controlling this process to activate or repress gene expression. Thousands of TFs in human.



TF A

DNA

TF B

mRNA

protein

phenotype

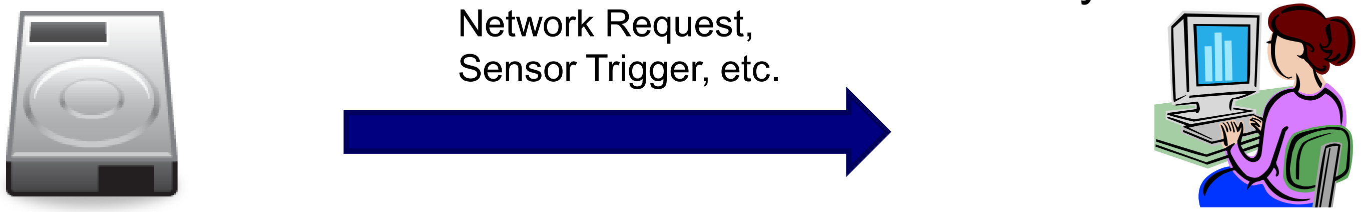
We can measure amounts of mRNA for ***every*** gene in a cell

Image adapted from Manolis Kellis

Gene Expression and its Regulation

Static code always stored

Dynamic code execution

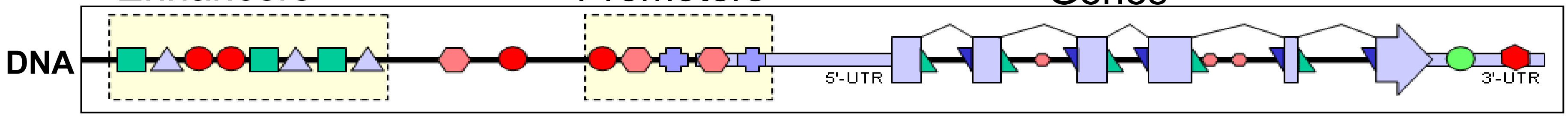


User Input,
Network Request,
Sensor Trigger, etc.

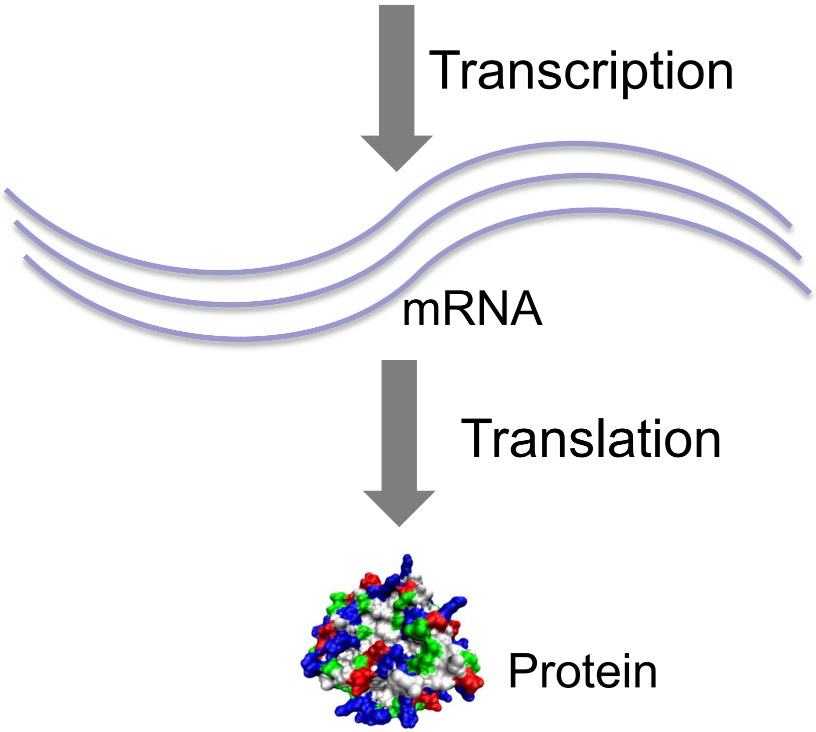
Enhancers

Promoters

Genes



Transcription Factors (TFs)



Transcription

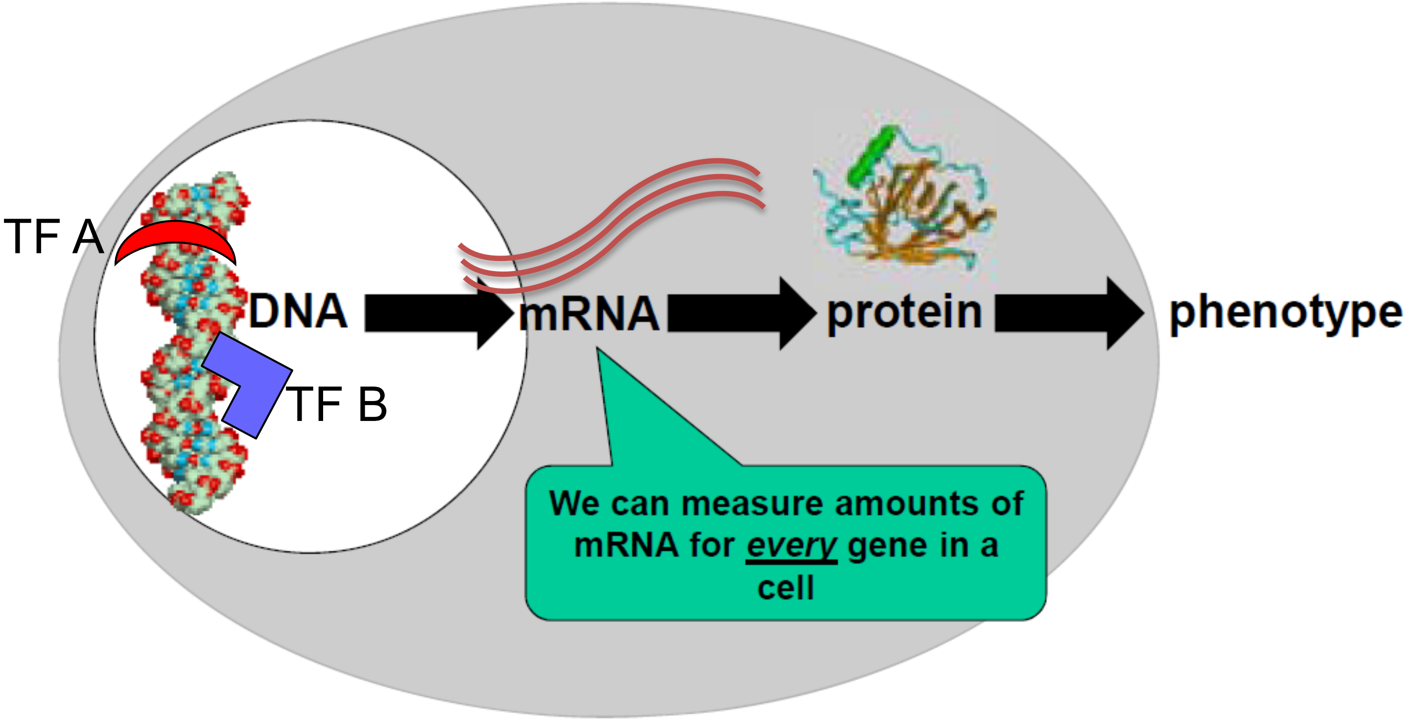
mRNA

Translation

Protein

Central Dogma

- The cell needs to regulate the process of going from DNA to mRNA.
- Transcription factors (TFs) binding DNA play a major role in controlling this process to activate or repress gene expression. Thousands of TFs in human.
- How does a transcription factor know to only bind specific locations in the genome?

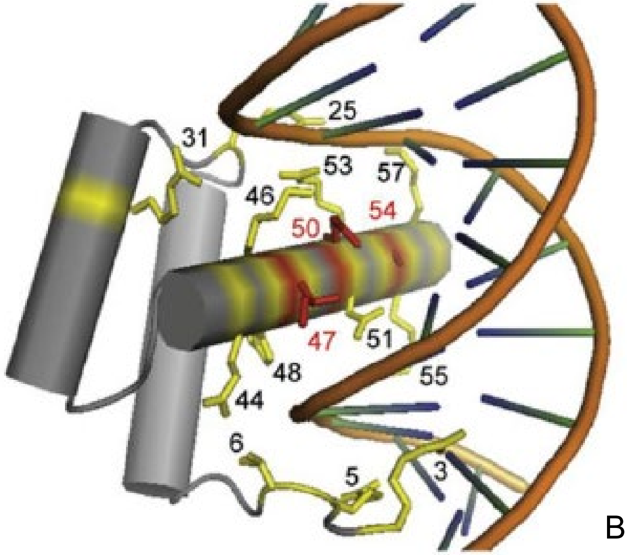


We can measure amounts of mRNA for *every* gene in a cell

Image adapted from Manolis Kellis

Transcription factor binding to DNA

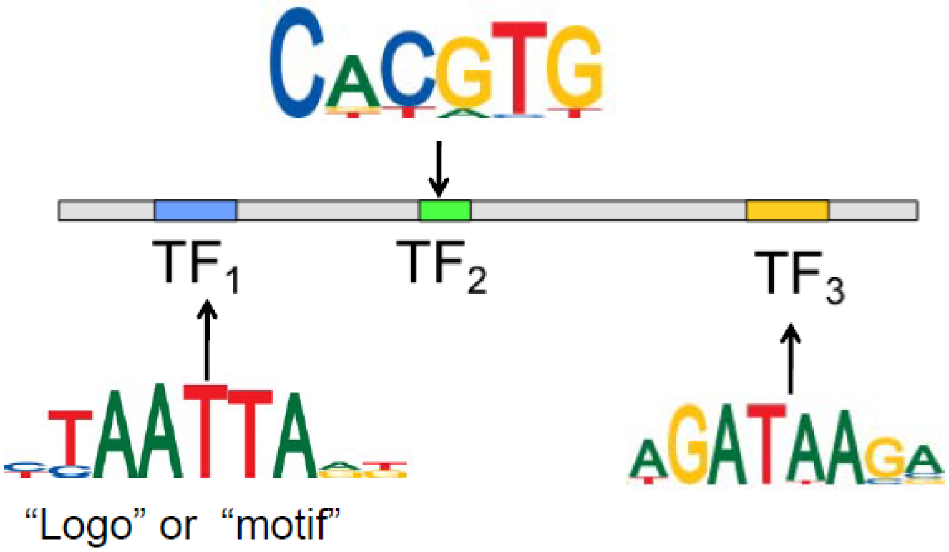
- Binding domain of transcription factors will preferentially recognize specific short DNA sequences based on biophysical constraints
- Preferences will differ between transcription factors



Berger et al, Cell 2008

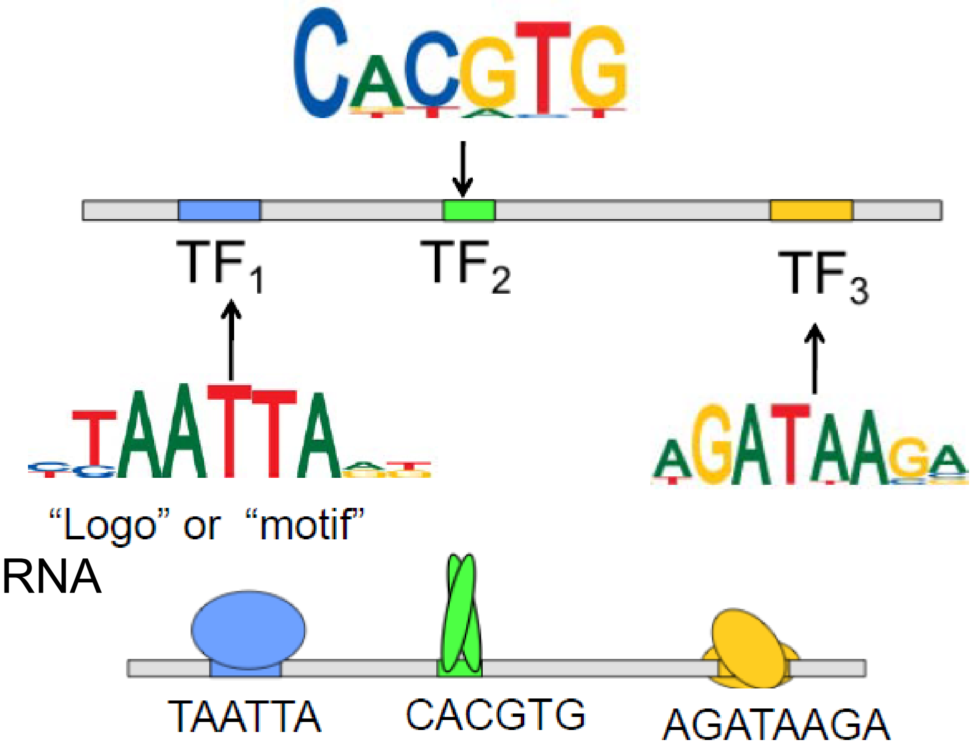
DNA-binding domain of *Engrailed*

Transcription factors recognize sequence motifs in genome



Based on slide from Manolis Kellis

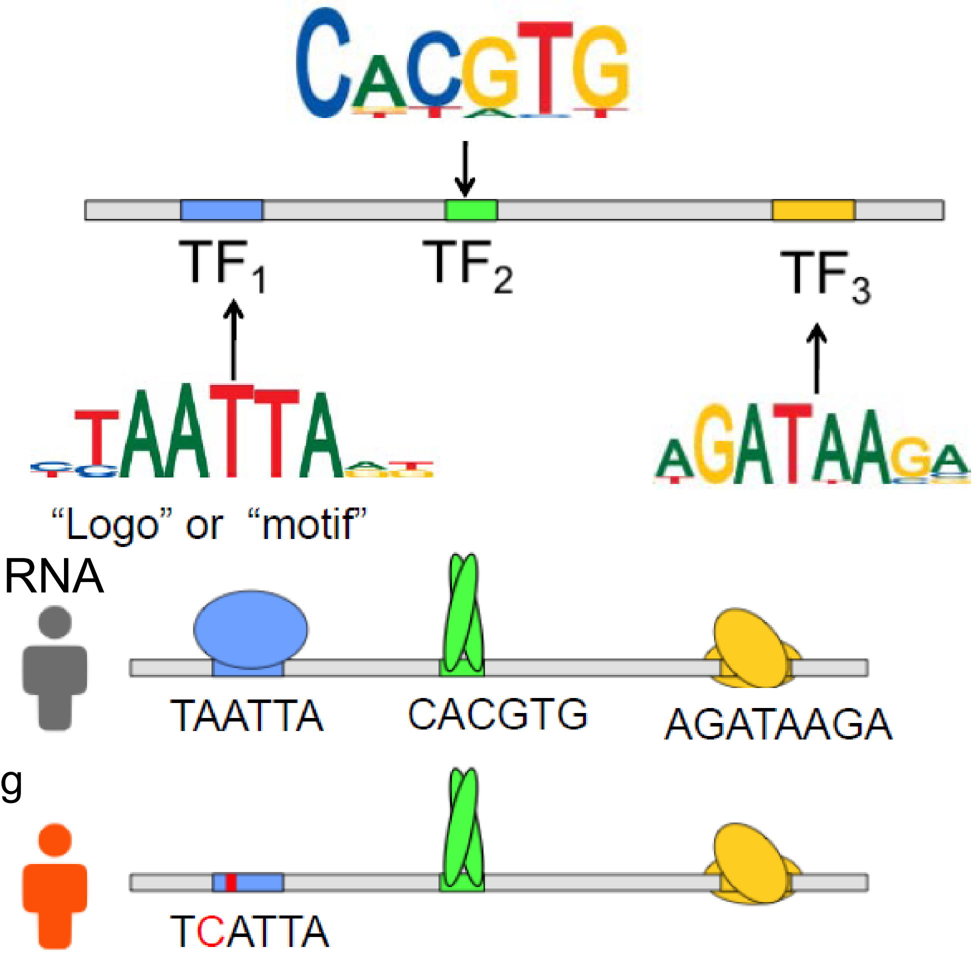
Understanding TF binding important to interpreting sequence variants



Binding can activate or repress production of mRNA

Based on slide from Manolis Kellis

Understanding TF binding important to interpreting sequence variants



Binding can activate or repress production of mRNA

Important for understanding non-coding variants associated with disease

Based on slide from Manolis Kellis

How to represent motifs

Possible ideas:

- K-mer

Define a motif to be a single k-mer:
e.g. ACTAGGAT

Question: What are potential advantages or disadvantages of this approach?

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood

(*k*,d)-motifs – a k-mer and all k-mers with at most *d* mismatches

(CAT,1): AAT,GAT,TAT,CCT,CGT,CTT,CAA,CAC,CAG,CAT

Question: What are potential advantages or disadvantages of this approach?

How to represent motifs

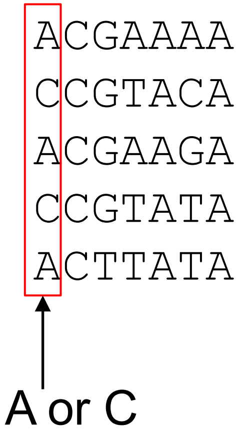
Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



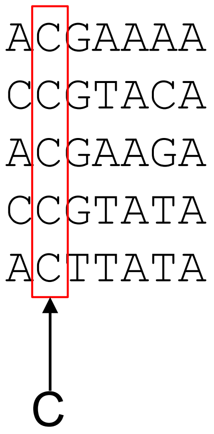
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

A or C

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



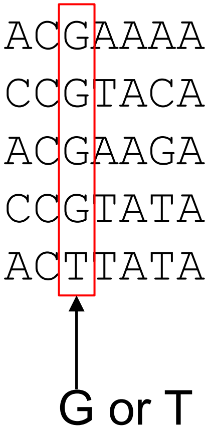
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

C

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



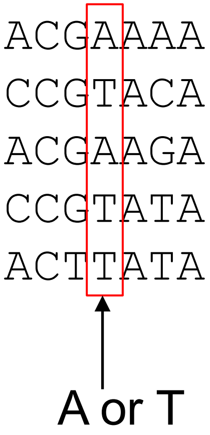
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

G or T

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



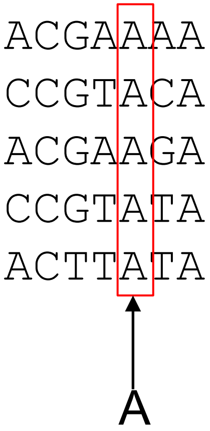
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

A or T

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



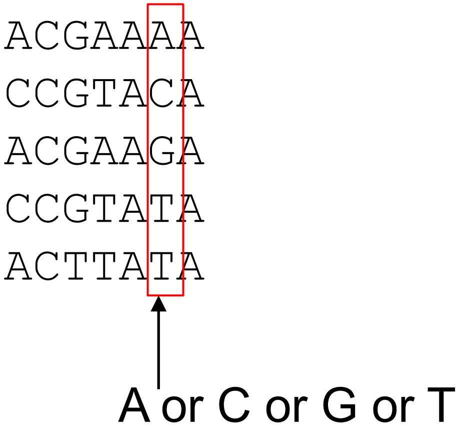
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

A

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



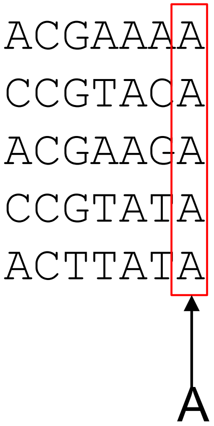
```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

A or C or G or T

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

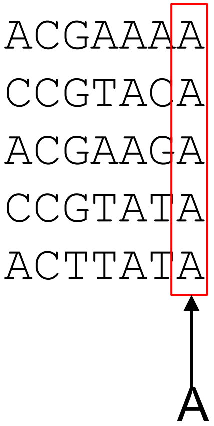
A

Question: How many non-empty combinations of four nucleotides are possible?

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes



```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

A

Question: How many non-empty combinations of four nucleotides are possible?

$$2^4 - 1 = 15$$

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes

```
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
=======
MCKWANA
```

Base set	IUPAC nucleotide code
A	A
C	C
G	G
T	T
A or G	R
C or T	Y
G or C	S
A or T	W
G or T	K
A or C	M
C or G or T	B
A or G or T	D
A or C or T	H
A or C or G	V
A or C or G or T	N

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes
- Positional weight matrix (PWM; profile matrix)

```
1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes
- Positional weight matrix (PWM; profile matrix)

```
1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: What are potential advantages or disadvantages of this approach?

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes
- Positional weight matrix (PWM; profile matrix)

```
1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: What assumptions are being made when representing an alignment as a positional weight matrix?

How to represent motifs

Possible ideas:

- K-mer
- K-mer neighborhood
- Degenerate sequence codes
- Positional weight matrix (PWM; profile matrix)

```
1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
```

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: What assumptions are being made when representing an alignment as a positional weight matrix?

Assuming independence. Also fixed spacing.

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: How should we score agreement of this sequence with the PWM?

CCGTATA

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	$\boxed{1}$	1/5	$\boxed{1}$
C	$\boxed{2/5}$	$\boxed{1}$	0	0	0	1/5	0
G	0	0	$\boxed{4/5}$	0	0	1/5	0
T	0	0	1/5	$\boxed{3/5}$	0	$\boxed{2/5}$	0

Question: How should we score agreement of this sequence with the PWM?

CCGTATA

$$\frac{2}{5} \times 1 \times \frac{4}{5} \times \frac{3}{5} \times 1 \times \frac{2}{5} \times 1 = \frac{48}{625}$$

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: How should we score agreement of this sequence with the PWM?

CCGTATA

$$\frac{2}{5}\times 1\times\frac{4}{5}\times\frac{3}{5}\times\ 1\ \times\frac{2}{5}\times 1=\frac{48}{625}$$

Question: What additional assumption are we implicitly making here?

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	$\boxed{1}$	1/5	$\boxed{1}$
C	$\boxed{2/5}$	$\boxed{1}$	0	0	0	1/5	0
G	0	0	$\boxed{4/5}$	0	0	1/5	0
T	0	0	1/5	$\boxed{3/5}$	0	$\boxed{2/5}$	0

Question: How should we score agreement of this sequence with the PWM?

CCGTATA

$$\frac{2}{5} \times 1 \times \frac{4}{5} \times \frac{3}{5} \times 1 \times \frac{2}{5} \times 1 = \frac{48}{625}$$

Question: What additional assumption are we implicitly making in the scoring here?

Each nucleotide is a priori equally likely

Background Models

- Probability can be evaluated relative to background model

$$log\,\frac{P(sequence|PWM)}{P(sequence|Background)}$$

If we assume a uniform background distribution over nucleotides, then each is assumed to occur with probability 0.25

CCGTATA

$$\log(\frac{\frac{48}{625}}{0.25^7})=7.14$$

If we assume G or C occur with probability 0.2 and As and Ts with probability 0.3

$$\log(\frac{\frac{48}{625}}{0.2^3\times0.3^4})=7.08$$

Background Models

- Probability can be evaluated relative to background model

$$log\,\frac{P(sequence|PWM)}{P(sequence|Background)}$$

If we assume a uniform background distribution over nucleotides, then each is assumed to occur with probability 0.25

CCGTATA

$$\log(\frac{\frac{48}{625}}{0.25^7})=7.14$$

If we assume G or C occur with probability 0.2 and As and Ts with probability 0.3

$$\log(\frac{\frac{48}{625}}{0.2^3\times 0.3^4})=7.08$$

For simplicity we will assume a uniform background which will make the denominator the same for all sequences of a fixed length

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

Question: How should we score agreement of this sequence with the PWM?

ACTTATA

$$\frac{3}{5} \times 1 \times \frac{1}{5} \times \frac{3}{5} \times 1 \times \frac{2}{5} \times 1 = \frac{18}{625}$$

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	$\boxed{3/5}$	0	0	2/5	$\boxed{1}$	1/5	$\boxed{1}$
C	2/5	$\boxed{1}$	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	$\boxed{1/5}$	$\boxed{3/5}$	0	$\boxed{2/5}$	0

ACTTATA

$$\frac{3}{5} \times 1 \times \frac{1}{5} \times \frac{3}{5} \times 1 \times \frac{2}{5} \times 1 = \frac{18}{625}$$

Question: How can we run into problems using this PWM for scoring?

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

ACTTATC

$$\frac{3}{5} \times 1 \times \frac{1}{5} \times \frac{3}{5} \times \ 1 \ \times \frac{2}{5} \times 0 = 0$$

Any sequence that has a nucleotide not previously observed in a position will always get a score of 0

Scoring with a positional weight matrix

	1	2	3	4	5	6	7
A	3/5	0	0	2/5	1	1/5	1
C	2/5	1	0	0	0	1/5	0
G	0	0	4/5	0	0	1/5	0
T	0	0	1/5	3/5	0	2/5	0

ACTTATC

$$\frac{3}{5} \times 1 \times \frac{1}{5} \times \frac{3}{5} \times \ 1 \ \times \frac{2}{5} \times 0 = 0$$

Any sequence that has a nucleotide not previously observed in a position will always get a score of 0

Question: What can be done to address this?

PWM based on pseudo-counts

Add one observation for each nucleotide at each position.
Could also add fractional or more than one observation

1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
AAAAAAA
CCCCCCC
GGGGGGG
TTTTTTT

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

PWM based on pseudo-counts

Add one observation for each nucleotide at each position.
Could also add fractional or more than one observation

1234567
ACGAAAA
CCGTACA
ACGAAGA
CCGTATA
ACTTATA
AAAAAAA
CCCCCCC
GGGGGGG
TTTTTTT

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

ACTTATC

$$\frac{4}{9}\times\frac{6}{9}\times\frac{2}{9}\times\frac{4}{9}\times\frac{6}{9}\times\frac{3}{9}\times\frac{1}{9}$$
$$= 0.000723$$

Scanning a sequence with a PWM

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

ACTTATCGA

Scanning a sequence with a PWM

Score each position and record matches above some threshold that depends on the PWM

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

ACTTATCGA

$$\frac{4}{9}\times\frac{6}{9}\times\frac{2}{9}\times\frac{4}{9}\times\frac{6}{9}\times\frac{3}{9}\times\frac{1}{9}$$
$$=0.000723$$

Scanning a sequence with a PWM

Score each position and record matches above some threshold that depends on the PWM

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

ACTTATCGA

$$\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{3}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{1}{9}$$
$$= 0.00000753$$

Scanning a sequence with a PWM

Score each position and record matches above some threshold that depends on the PWM

	1	2	3	4	5	6	7
A	4/9	1/9	1/9	3/9	6/9	2/9	6/9
C	3/9	6/9	1/9	1/9	1/9	2/9	1/9
G	1/9	1/9	5/9	1/9	1/9	2/9	1/9
T	1/9	1/9	2/9	4/9	1/9	3/9	1/9

AC$\boxed{\text{TTATCGA}}$

$$\frac{1}{9}\times\frac{1}{9}\times\frac{1}{9}\times\frac{4}{9}\times\frac{1}{9}\times\frac{2}{9}\times\frac{6}{9}$$
$$= 0.0000100$$

Libraries of Hundreds PWMs exist

- Derived from aligned sets of short curated sequence from small-scale experiments
- Discovered de novo from high-throughput experiments

Libraries of Hundreds PWMs exist

- Derived from aligned sets of short curated sequence from small-scale experiments
- Discovered de novo from high-throughput experiments

One scan set of sequences for libraries of available PWMs and compute statistical enrichments

Topics

- Motif background and representations
- De novo motif discovery

De Novo Motif Discovery

Problem: Give a collection of sequences identify motifs *de novo*

Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA

De Novo Motif Discovery

Problem: Give a collection of sequences identify motifs *de novo*

Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA

Do you see any shared pattern in the above set of sequences?

De Novo Motif Discovery

Problem: Give a collection of sequences identify motifs *de novo*

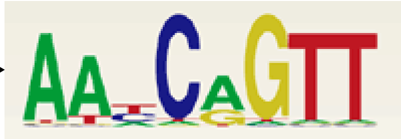
Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA

Do you see any shared pattern in the above set of sequences?

De Novo Motif Discovery

Problem: Give a collection of sequences identify motifs *de novo*

Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA

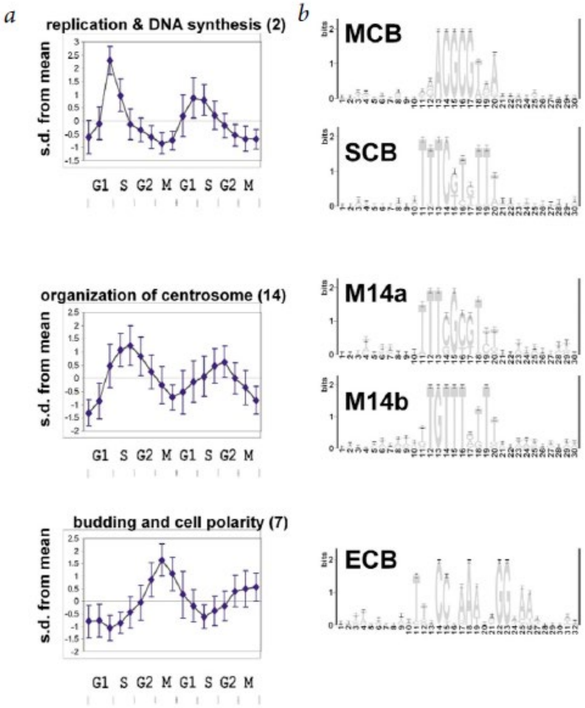


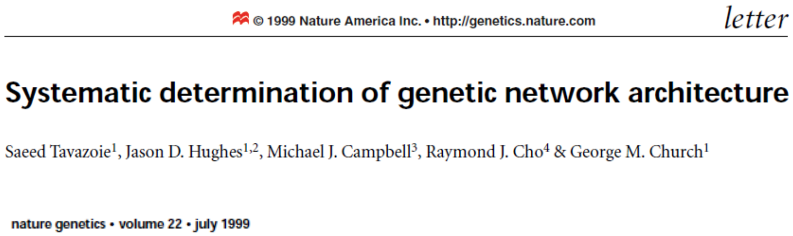
Do you see any shared pattern in the above set of sequences?

Adapted from D'haeseleer 2006

Examples of sets of sequences for motif discovery

- Promoter regions of co-expressed genes





© 1999 Nature America Inc. • http://genetics.nature.com *letter*

Systematic determination of genetic network architecture

Saeed Tavazoie[1], Jason D. Hughes[1,2], Michael J. Campbell[3], Raymond J. Cho[4] & George M. Church[1]

nature genetics • volume 22 • july 1999

Applied motif discovery on 600bp upstream of genes in the same k-means clusters

Examples of sets of sequences for motif discovery

- Promoter regions of co-expressed genes
- Locations of TF binding across the genome from a mapping experiment

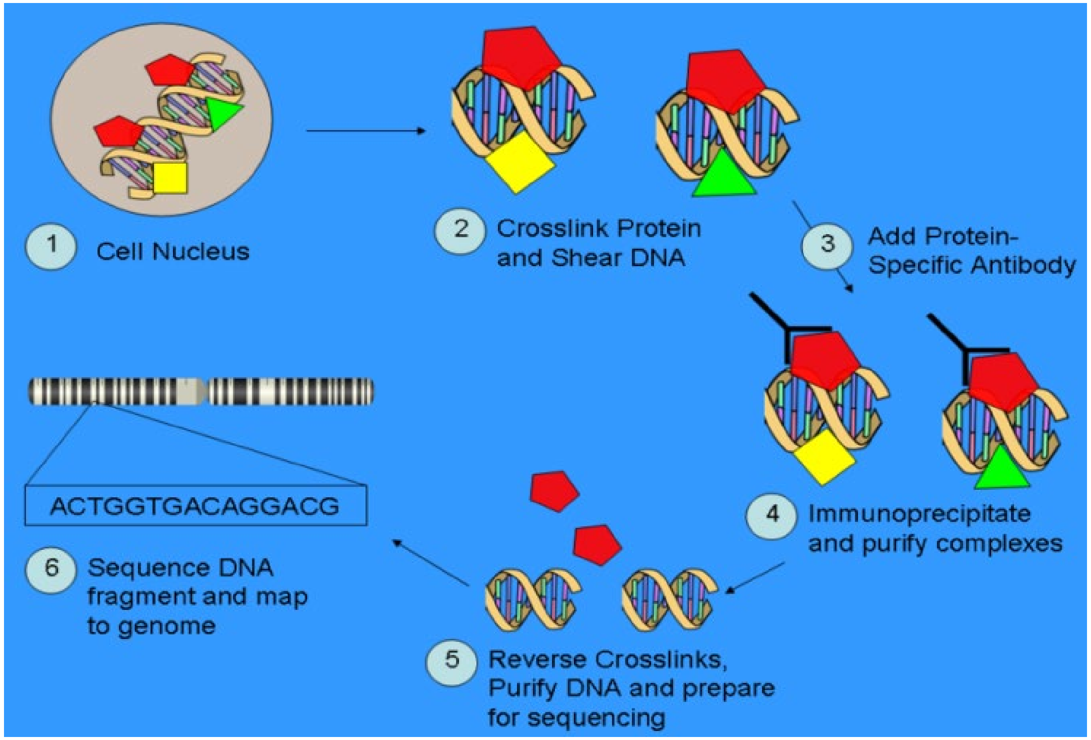
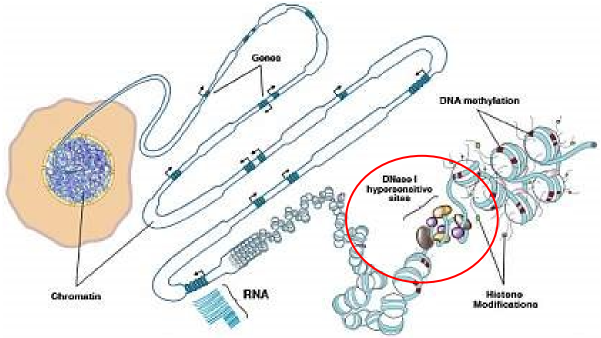

1 Cell Nucleus
2 Crosslink Protein and Shear DNA
3 Add Protein-Specific Antibody
4 Immunoprecipitate and purify complexes
5 Reverse Crosslinks, Purify DNA and prepare for sequencing
6 Sequence DNA fragment and map to genome
ACTGGTGACAGGACG

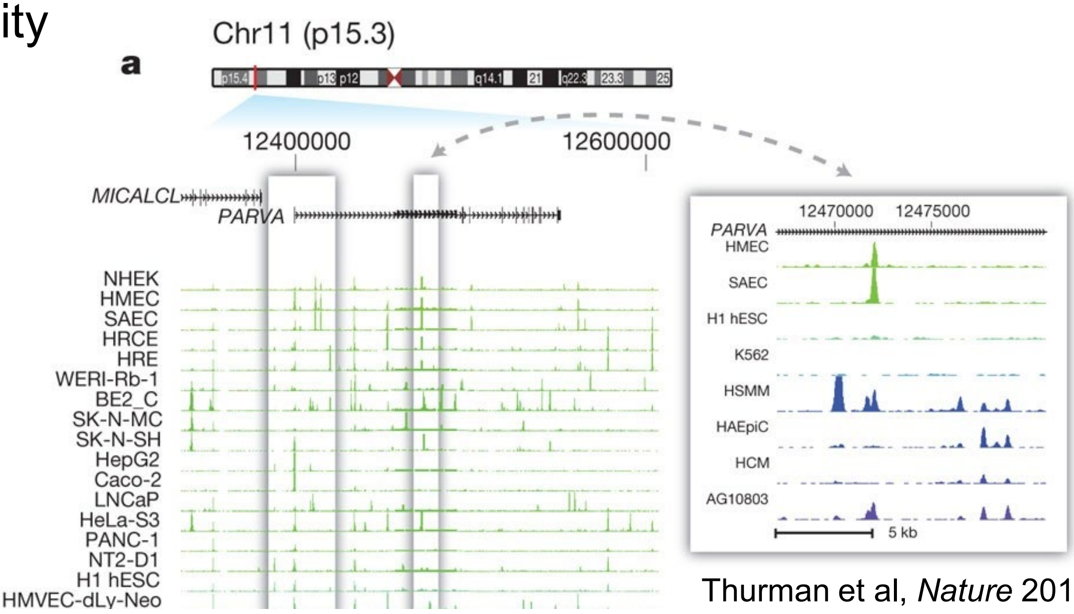
Image from Wikipedia

Examples of sets of sequences for motif discovery

- Promoter regions of co-expressed genes
- Locations of TF binding across the genome from a mapping experiment
- Regions across the genome where the DNA is accessible in a cell type from a mapping experiment

Mapped by DNase I hypersensitivity or ATAC-seq





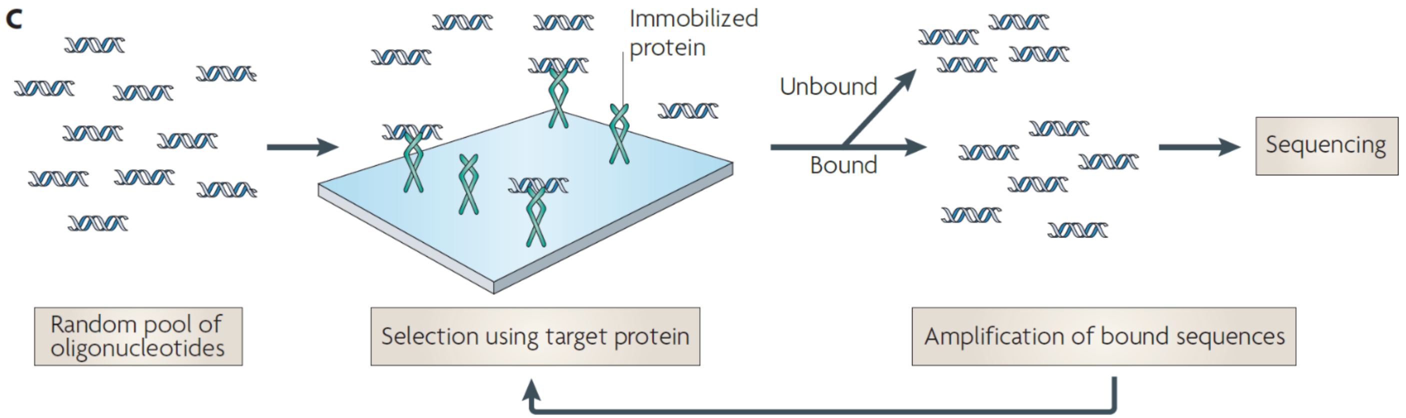
Thurman et al, *Nature* 2012

Image source : https://www.nih.gov/news-events/news-releases/nih-supported-researchers-map-epigenome-more-100-tissue-cell-types

Examples of sets of sequences for motif discovery

- Promoter regions of co-expressed genes
- Locations of TF binding across the genome from a mapping experiment
- Regions across the genome where the DNA is accessible in a cell type from a mapping experiment
- Experiments designed to measure TF binding specificity

High-throughput SELEX experiment


c
Immobilized protein
Unbound
Bound
Sequencing
Random pool of oligonucleotides
Selection using target protein
Amplification of bound sequences

Stormo and Zhao, Nature Reviews Genetics 2010

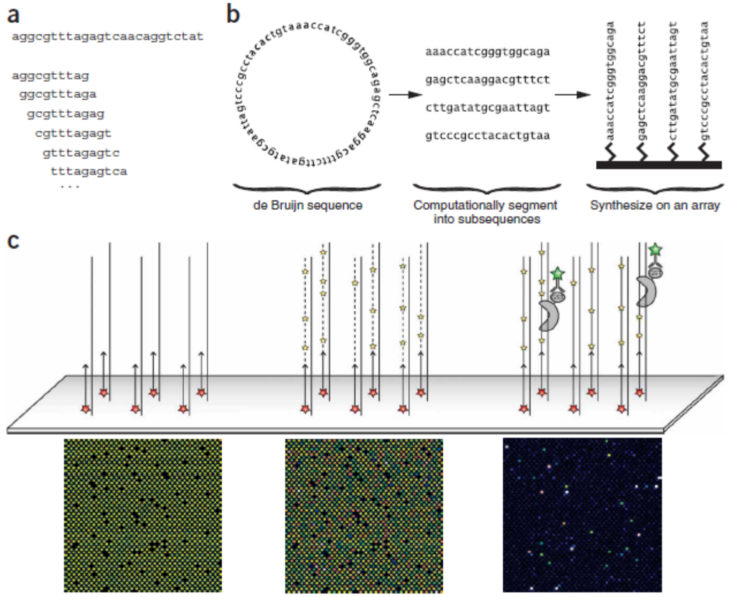
Examples of sets of sequences for motif discovery

- Promoter regions of co-expressed genes
- Locations of TF binding across the genome from a mapping experiment
- Regions across the genome where the DNA is accessible in a cell type from a mapping experiment
- Experiments designed to measure TF binding specificity

Protein binding microarray

Design properties:

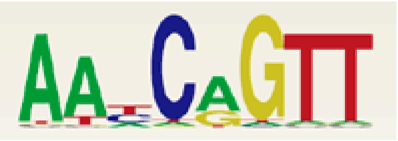
- 44,000 sequences of 35bp
- Sequences designed such all 10mers appear once
- All 8-mers appear 16 times



A formulation of the motif discovery problem

- Give an input motif of length *k* and set of *t* sequences
- Output a motif instance for each input sequence and a corresponding motif that optimizes some objective function
- Assumption each input sequence has one instance of the motif

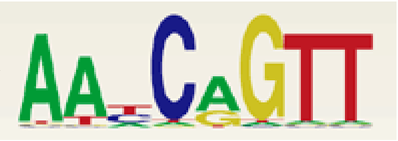
Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA



A formulation of the motif discovery problem

- Give an input motif of length *k* and set of *t* sequences
- Output a motif instance for each input sequence and a corresponding motif that optimizes some objective function
- Assumption each input sequence has one instance of the motif

Sequence 1	AATCAGTTATCTGTTGTATACCCGGAGTCC
Sequence 2	AGGTCGAATGAAACGTTCTTGCACGTACAT
Sequence 3	GAGATAACCGCTTGATATGACTCATTGCCA
Sequence 4	ATATTCCGGACGCTGTGACGATCCGGTTGT
Sequence 5	GAACGCAACCAGTTCAGTGCTTATCATGAA



Will depend on motif representation and scoring function.
Also will need a way to optimize the score.

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a k-mer string, how could we score motif instances?

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a k-mer string, how could we score motif instances?

Hamming distance – number of mis-matches
e.g. d(CAT,TAT) = 1

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a k-mer string, how could we score motif instances? Hamming distance – number of mis-matches e.g. d(CAT,TAT) = 1
- **Question:** What could our overall optimization function be?

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a k-mer string, how could we score motif instances? Hamming distance – number of mis-matches e.g. d(CAT,TAT) = 1
- **Question:** What could our overall optimization function be?

Minimize sum across all instances. “Median string problem.”

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a PWM, how could we score motif instances?

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a PWM, how could we score motif instances?

  Can use probabilities derived earlier or log of them

  Note: textbook uses simpler based on number of mismatches with consensus

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a PWM, how could we score motif instances?

  Can use probabilities derived earlier or log of them

- **Question:** How can we combine across multiple sequences?

Scoring a set of motif instances

- Will depend on motif representation
- Need to score individual instances and then combine the scores
- **Question:** If our motif representation was a PWM, how could we score motif instances?

  Can use probabilities derived earlier or log of them

- **Question:** How can we combine across multiple sequences

  Sum of the log probabilities

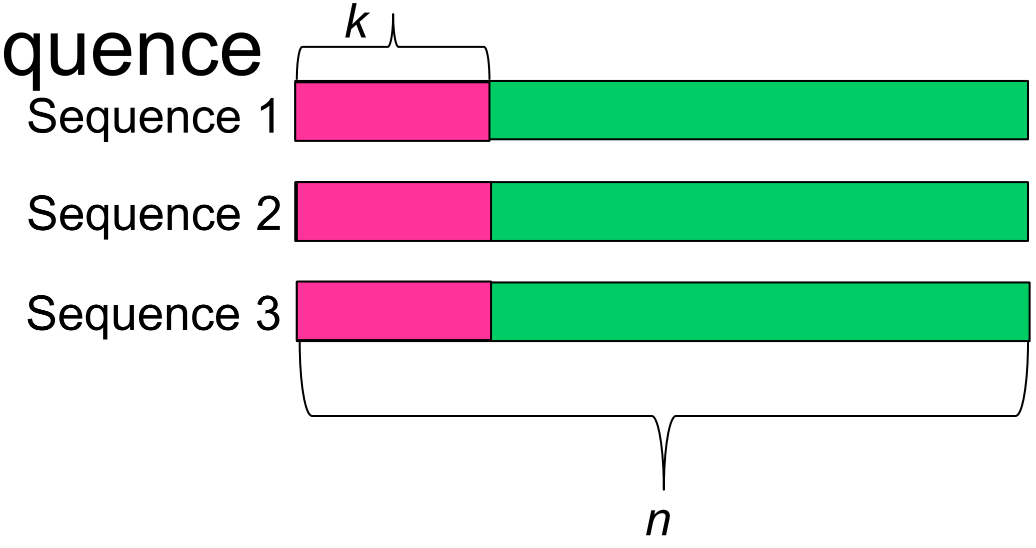
Optimization problem

- We need to find a motif instance from each sequence and corresponding motif
- What are some brute force strategies we could use for this problem?

Brute force approaches to motif discovery

- Idea 1: Try every possible combination of position in each sequence

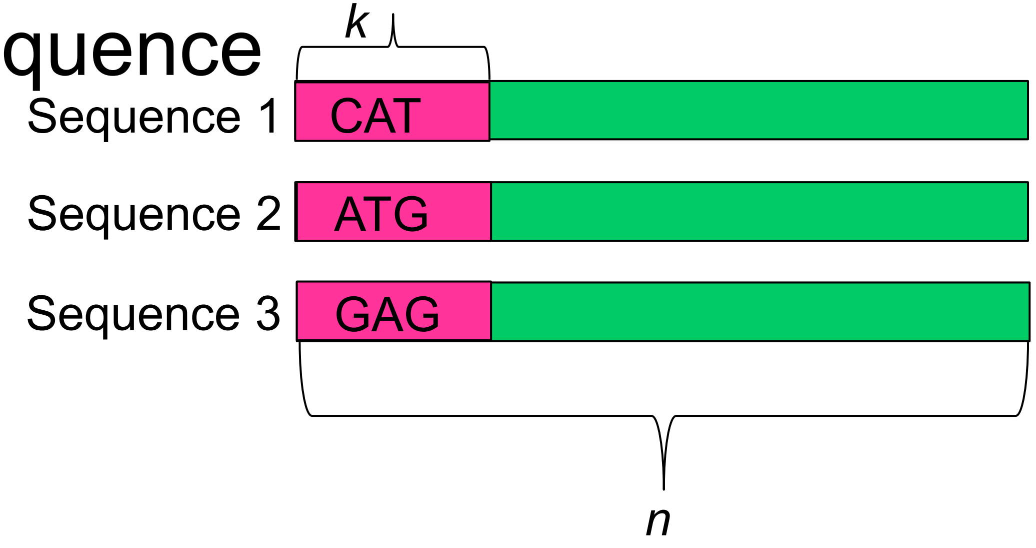
Suppose we have
t sequences
n nucleotides per sequence
k is length of motif



Brute force approaches to motif discovery

- Idea 1: Try every possible combination of position in each sequence

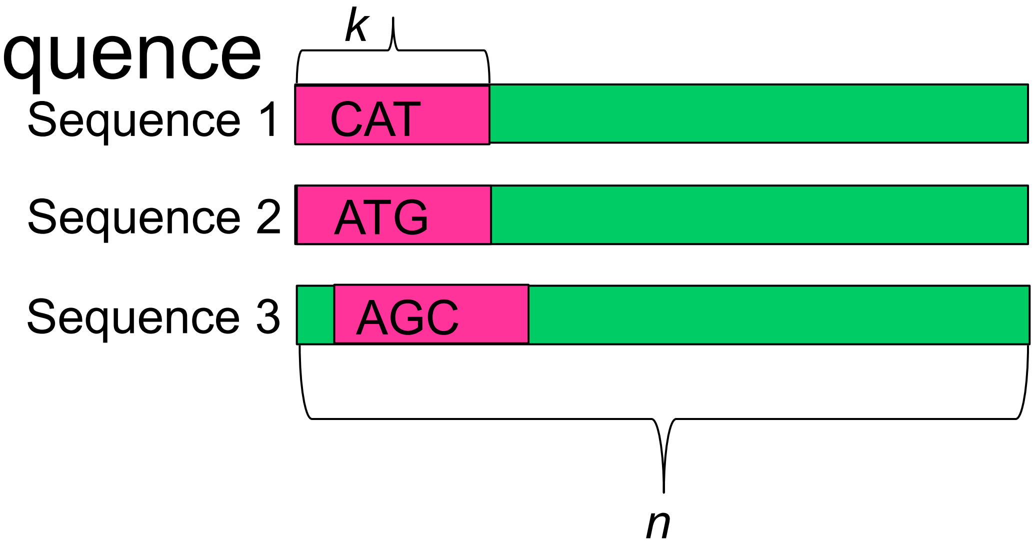
Suppose we have
t sequences
n nucleotides per sequence
k is length of motif



Brute force approaches to motif discovery

- Idea 1: Try every possible combination of position in each sequence

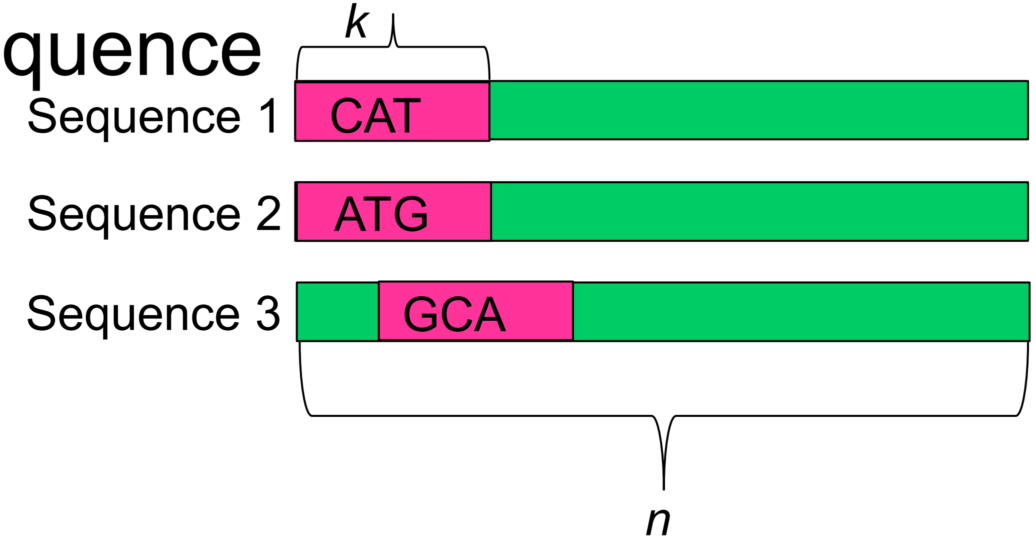
Suppose we have
t sequences
n nucleotides per sequence
k is length of motif



Brute force approaches to motif discovery

- Idea 1: Try every possible combination of position in each sequence

Suppose we have
t sequences
n nucleotides per sequence
k is length of motif

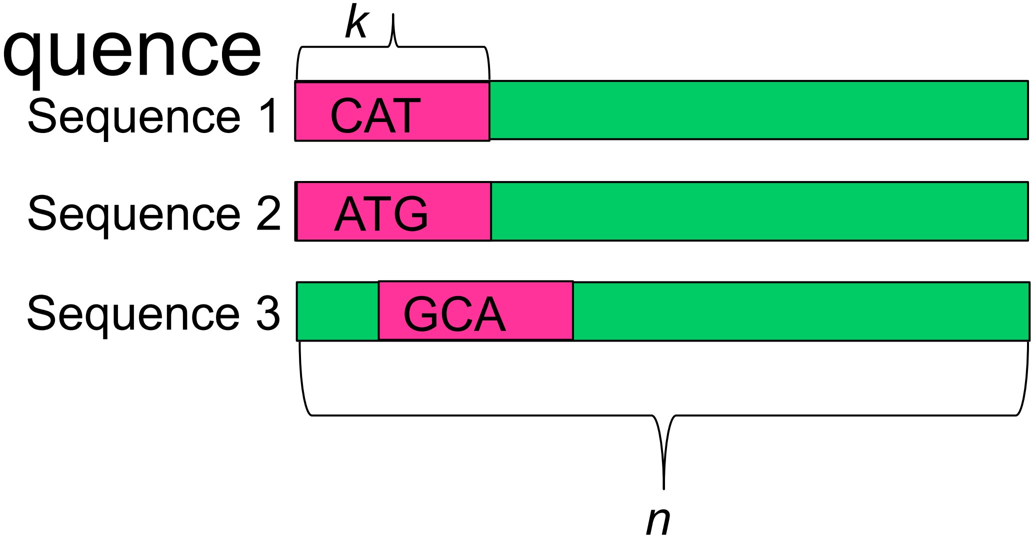


Question: How long would this take to solve assuming *k* is much smaller than *n*?

Brute force approaches to motif discovery

- Idea 1: Try every possible combination of position in each sequence

Suppose we have
t sequences
n nucleotides per sequence
k is length of motif



Question: How long would this take to solve assuming k is much smaller than n?

$O((n-k+1)^t)$

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a k-mer sequence
t sequences
n nucleotides per sequence
k is length of motif

Question: What complexity be of trying and evaluating all k-mers

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

  Suppose our motif representation is a k-mer sequence
  t sequences
  n nucleotides per sequence
  k is length of motif

Question: What complexity be of trying and evaluating all k-mers

4^k possible k-mers. *n* x *k* x *t* time to evaluate each k-mer
Would require $O(4^k * n * k * t)$ time

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a k-mer sequence
t sequences
n nucleotides per sequence
k is length of motif

Question: What complexity be of trying and evaluating all k-mers

4^k possible k-mers. n x k x t time to evaluate each k-mer
Would require $O(4^k * n * k * t)$ time

Note compares favorably to previous approach for large *n*. Why?

$O((n-k+1)^t)$

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a k-mer sequence
t sequences
n nucleotides per sequence
k is length of motif

Question: What complexity be of trying and evaluating all k-mers

4^k possible k-mers. $n \times k \times t$ time to evaluate each k-mer
Would require $O(4^k * n * k * t)$ time

Note compares favorably to previous approach for large *n*. Why?

$O((n-k+1)^t)$

Can independently score each sequence

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

  Suppose our motif representation is a PWM
  t sequences
  n nucleotides per sequence
  k is length of motif

Question: How can we try to (approximately) optimize this if applying brute force on the PWM representation?

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a PWM
t sequences
n nucleotides per sequence
k is length of motif

Question: How can we try to (approximately) optimize this if applying brute force on the PWM representation?

Discretize entries into *d* possible values for each entry of PWM. Could cover space of PWMs with $d=t+1$

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a PWM
t sequences
n nucleotides per sequence
k is length of motif

Question: How can we try to (approximately) optimize this if applying brute force on the PWM representation?

Discretize entries into *d* possible values for each entry of PWM. Could cover space of PWMs with $d=t+1$
Question: What would the complexity of this be?

Brute force approaches to motif discovery

- Idea 2: Brute force search over the motif representation

Suppose our motif representation is a PWM
t sequences
n nucleotides per sequence
k is length of motif

Question: How can we try to (approximately) optimize this if applying brute force on the PWM representation?

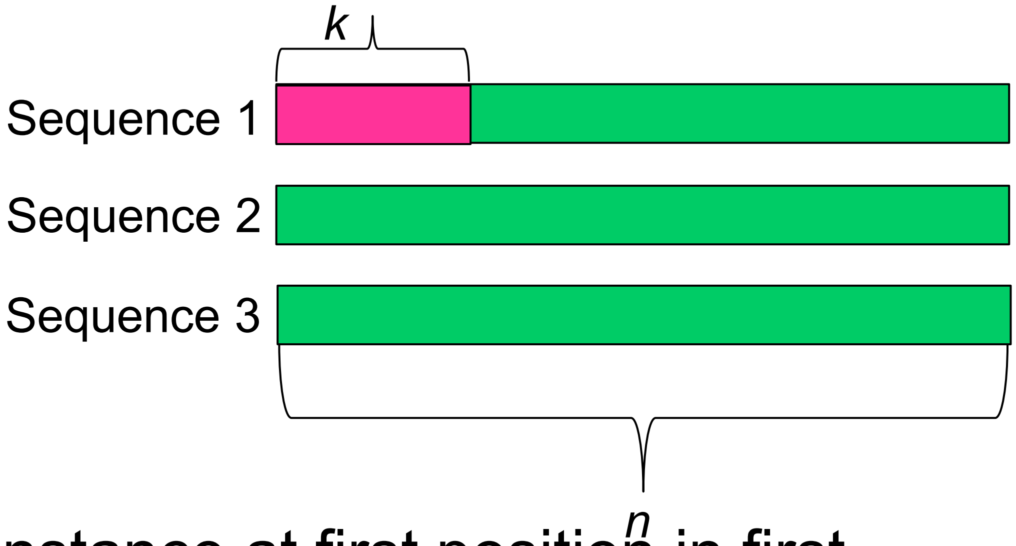
Discretize entries into *d* possible values for each entry of PWM. Could cover space of PWMs with $d=t+1$
Question: What would the complexity of this be?

$$O(d^{3k} * n* k * t)$$

Optimization problem

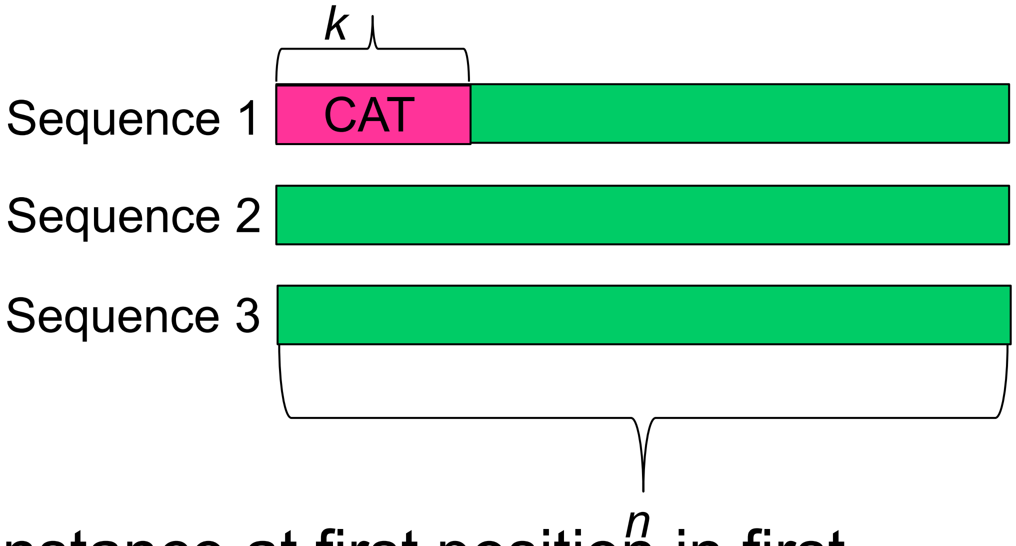
- We need to find a motif instance from each sequence and corresponding motif
- What are some brute force strategies we could use for this problem?
- What are some other strategies?

Greedy approach to motif discovery



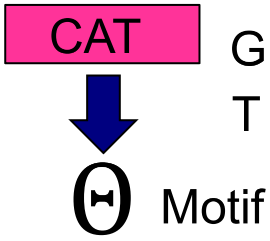
- Start by placing a motif instance at first position in first sequence

Greedy approach to motif discovery

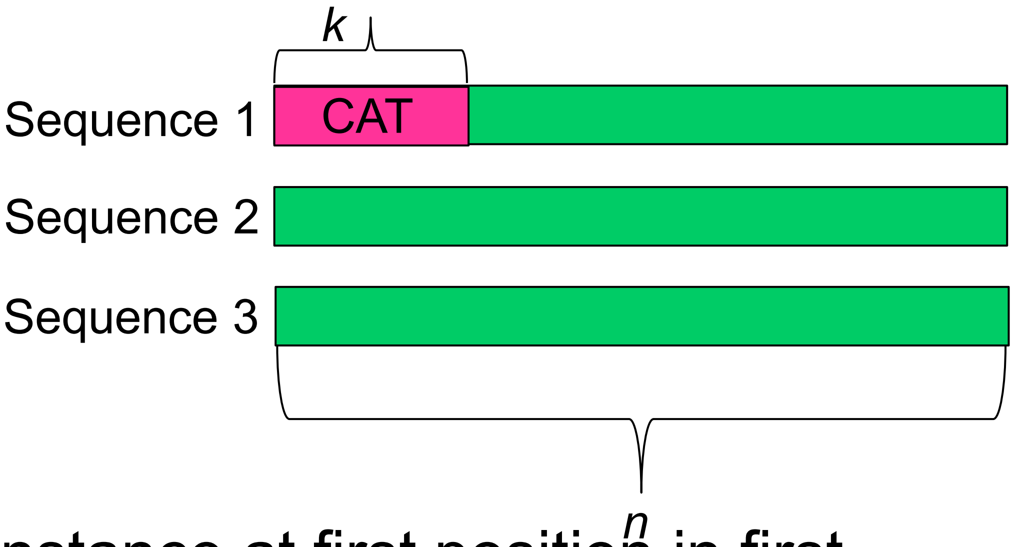


- Start by placing a motif instance at first position in first sequence

Greedy approach to motif discovery



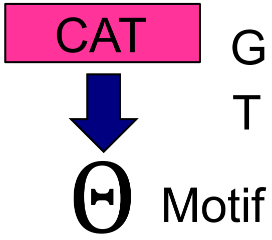
	1	2	3
A	1/5	2/5	1/5
C	2/5	1/5	1/5
G	1/5	1/5	1/5
T	1/5	1/5	2/5

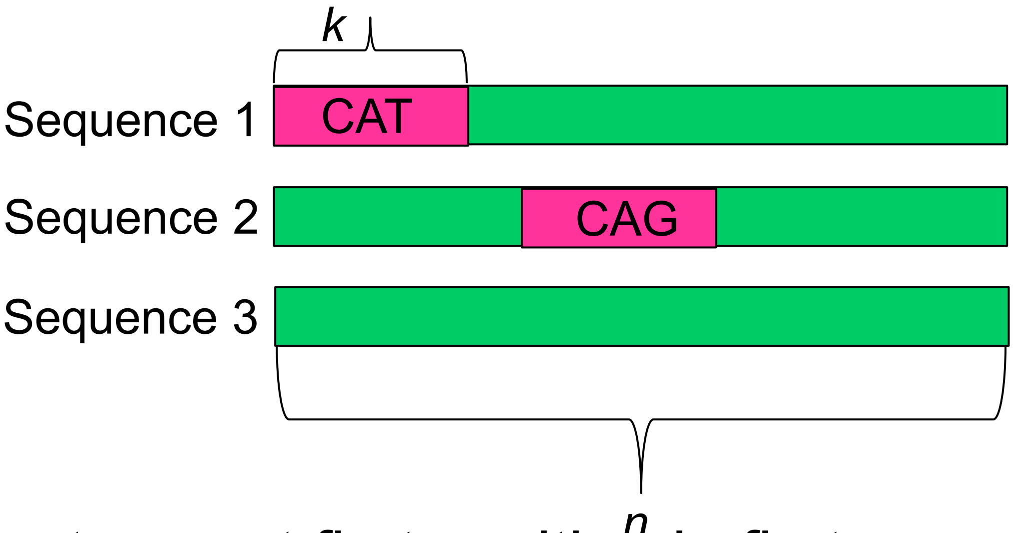


- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)

Greedy approach to motif discovery

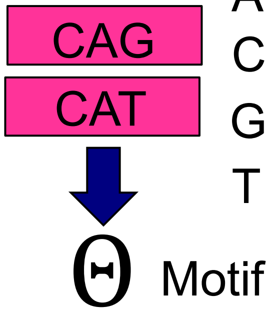
	1	2	3
A	1/5	2/5	1/5
C	2/5	1/5	1/5
G	1/5	1/5	1/5
T	1/5	1/5	2/5



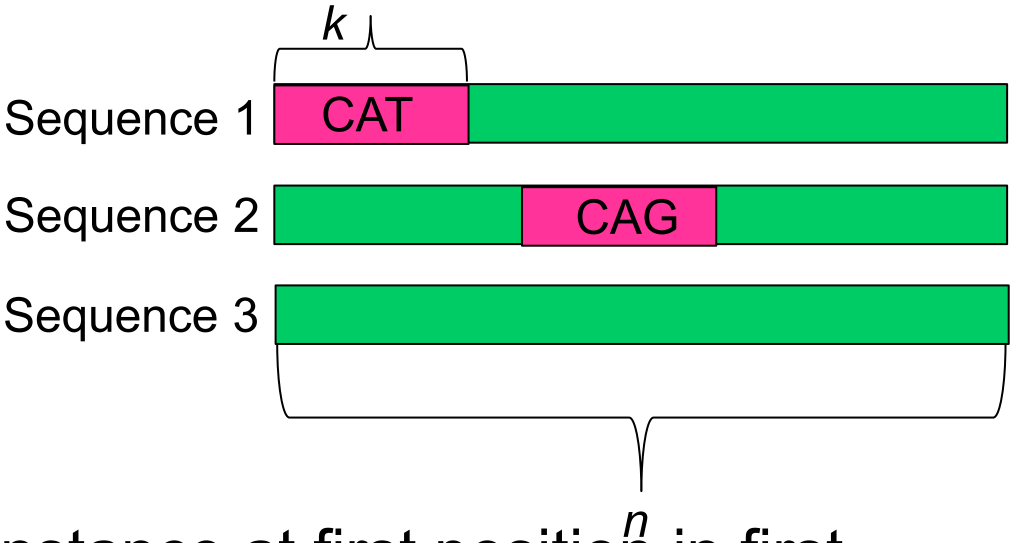


- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence

Greedy approach to motif discovery

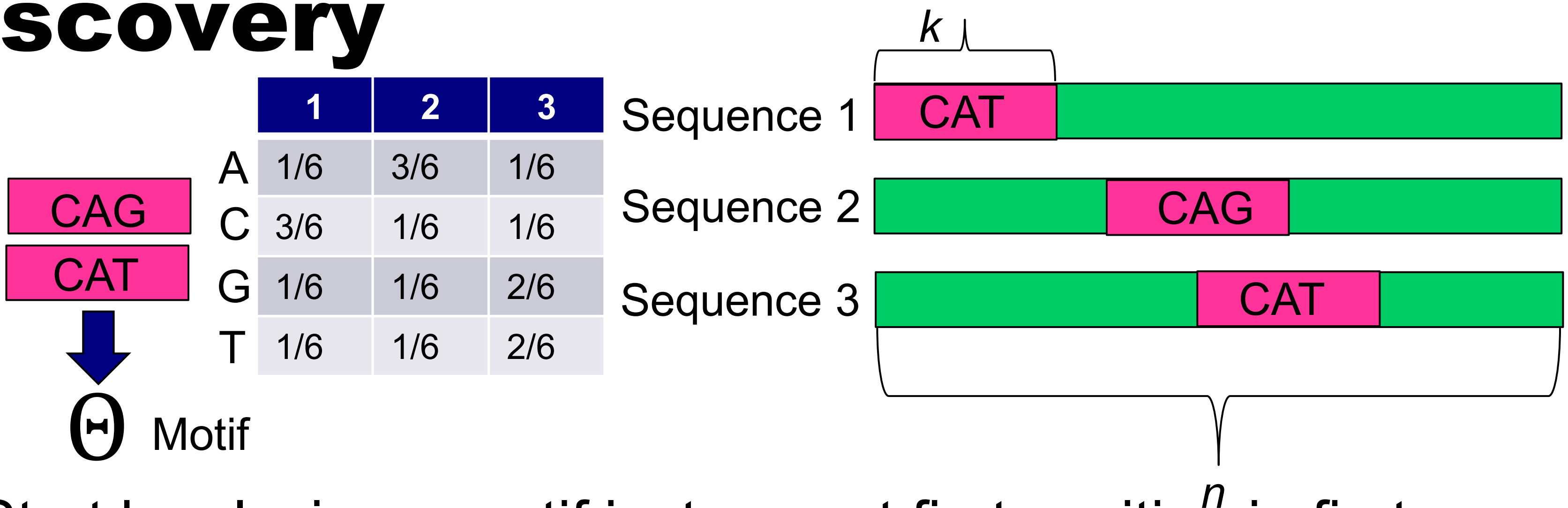


	1	2	3
A	1/6	3/6	1/6
C	3/6	1/6	1/6
G	1/6	1/6	2/6
T	1/6	1/6	2/6



- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif

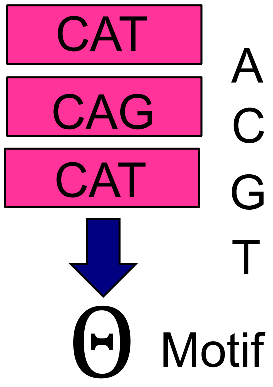
Greedy approach to motif discovery



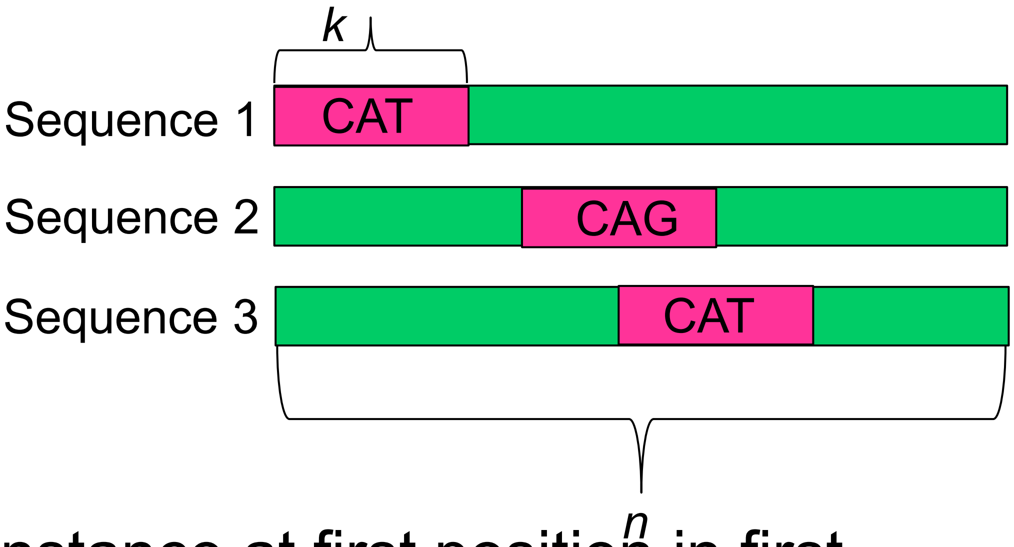
	1	2	3
A	1/6	3/6	1/6
C	3/6	1/6	1/6
G	1/6	1/6	2/6
T	1/6	1/6	2/6

- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence

Greedy approach to motif discovery

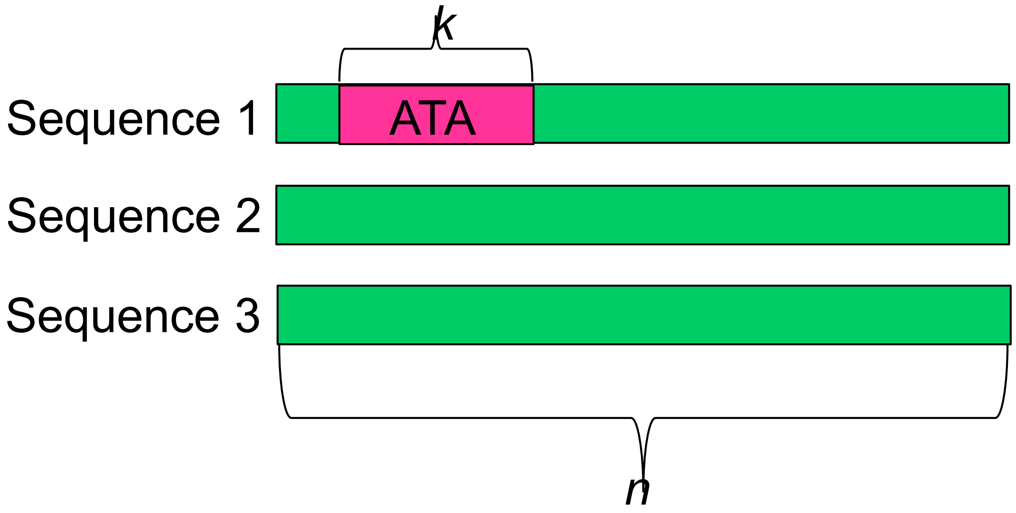


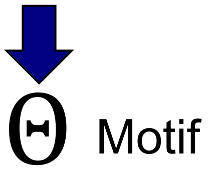
	1	2	3
A	1/7	4/7	1/7
C	4/7	1/7	1/7
G	1/7	1/7	2/7
T	1/7	1/7	3/7



- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence

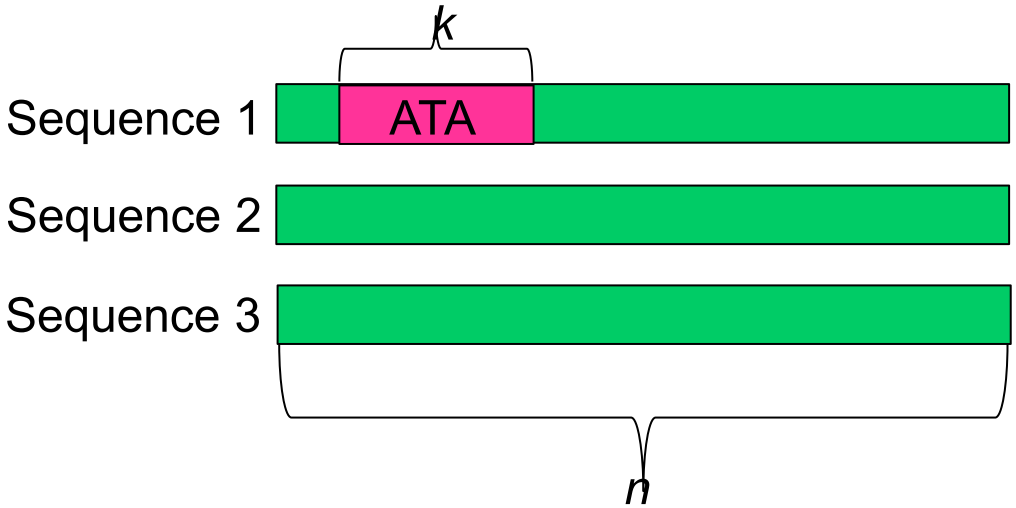
Greedy approach to motif discovery

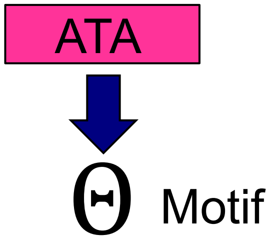




- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence
- Repeat for next start position of sequence 1

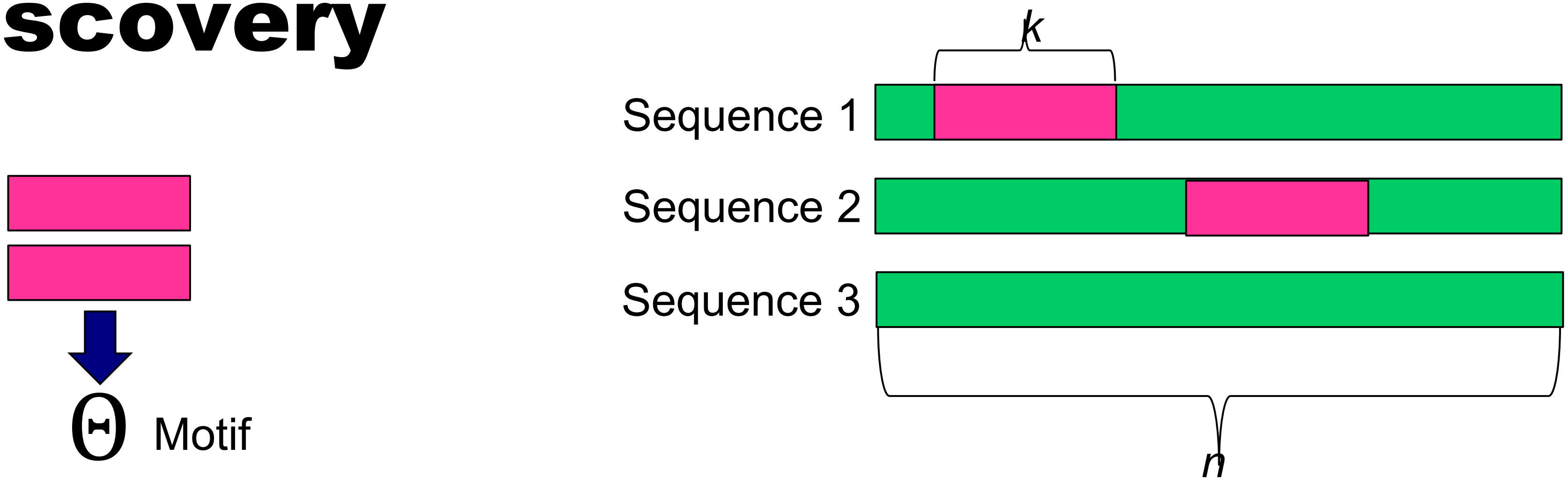
Greedy approach to motif discovery





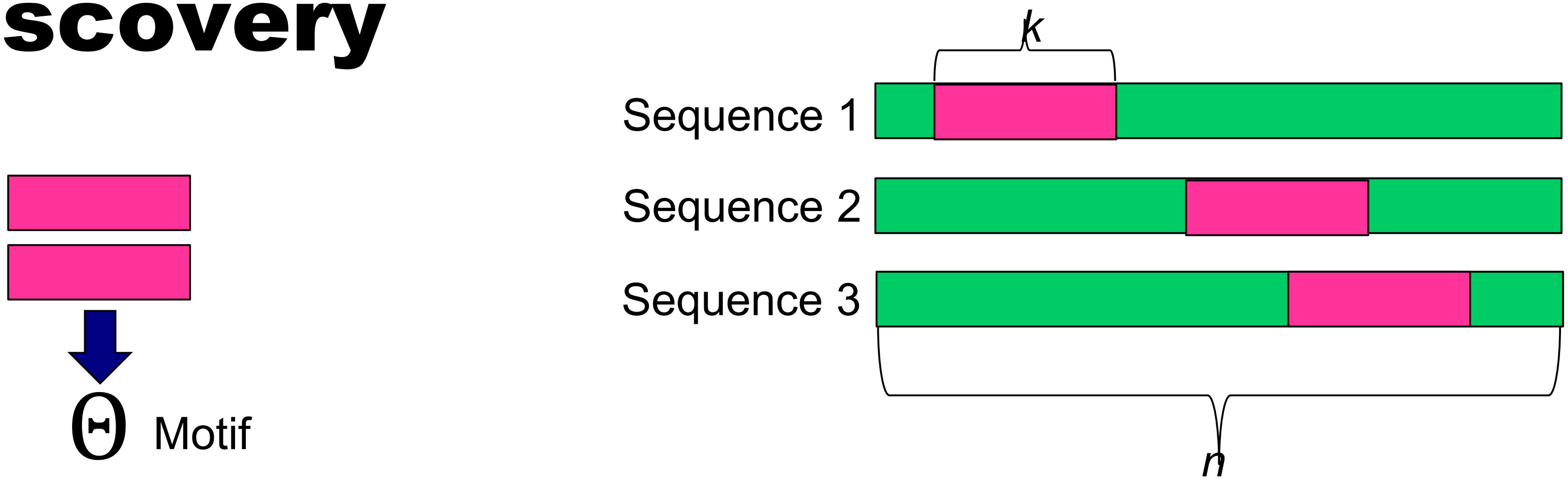
- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence
- Repeat for next start position of sequence 1

Greedy approach to motif discovery



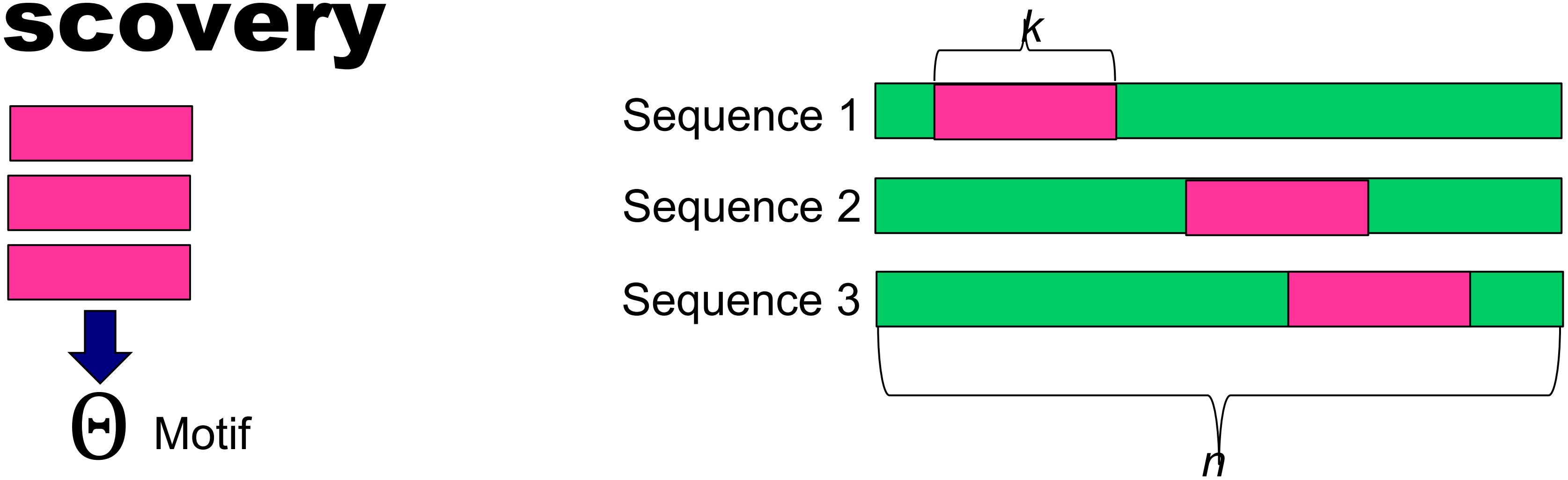
- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence
- Repeat for next start position of sequence 1

Greedy approach to motif discovery



- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence
- Repeat for next start position of sequence 1

Greedy approach to motif discovery

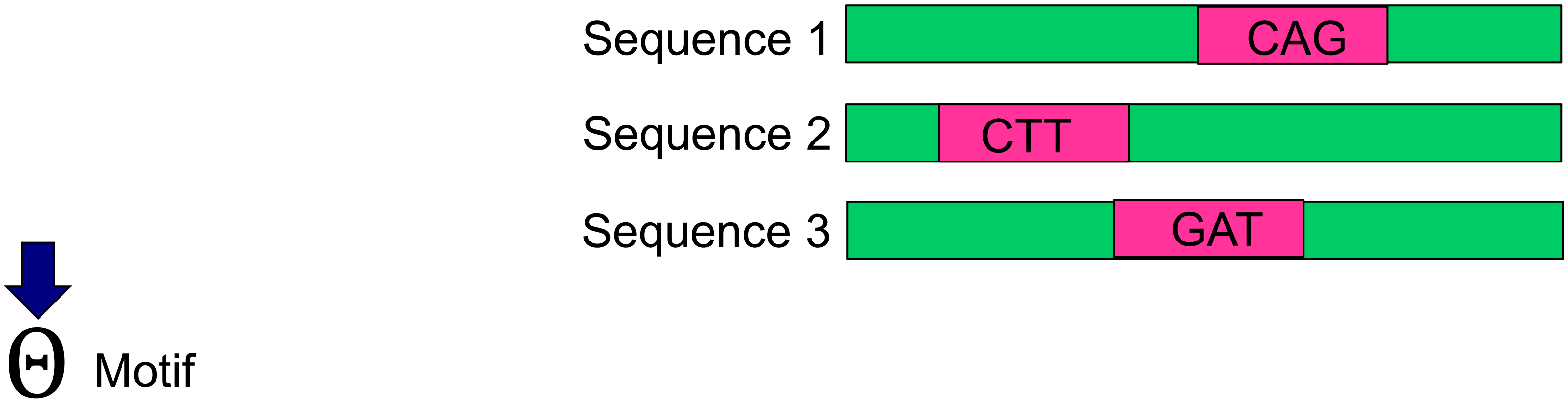


- Start by placing a motif instance at first position in first sequence
- Build motif based off of it (with pseudocounts)
- Identify highest scoring motif instance in second sequence
- Update motif
- Repeat for next sequence
- Repeat for next start position of sequence 1

Optimization problem

- We need to find a motif instance from each sequence and corresponding motif
- What are some brute force strategies we could use for this problem?
- What are some limitations/other strategies?

Random Initialization + Iterative Batch Greedy Updates



Sequence 1 CAG

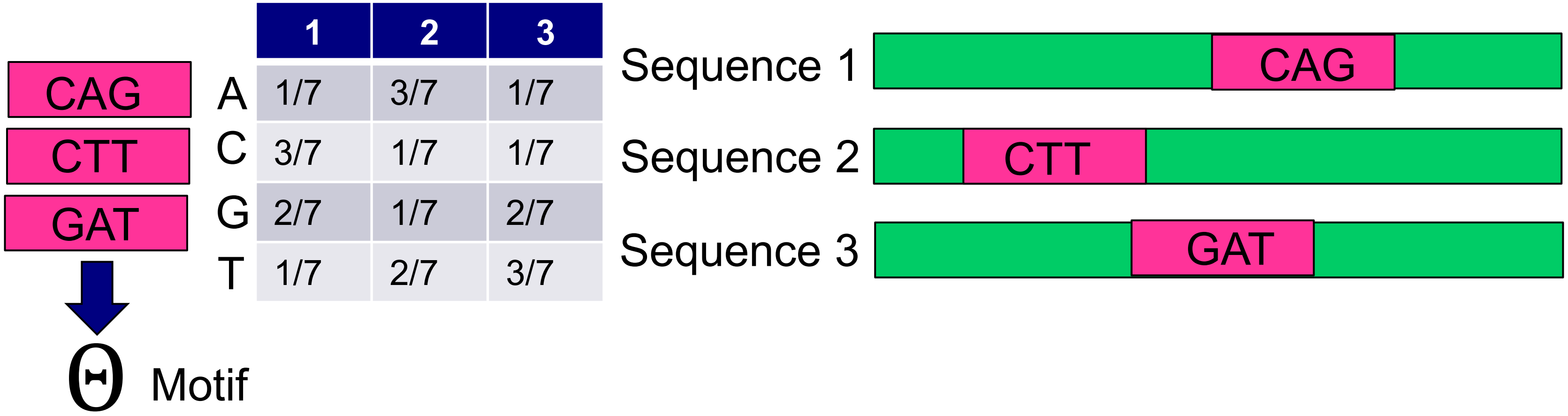
Sequence 2 CTT

Sequence 3 GAT

Θ Motif

- Start by placing a motif instance randomly for each sequence

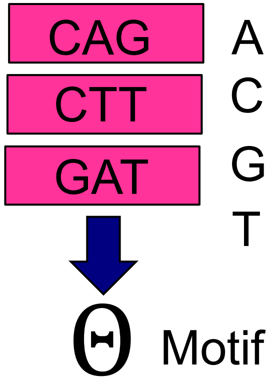
Random Initialization + Iterative Batch Greedy Updates



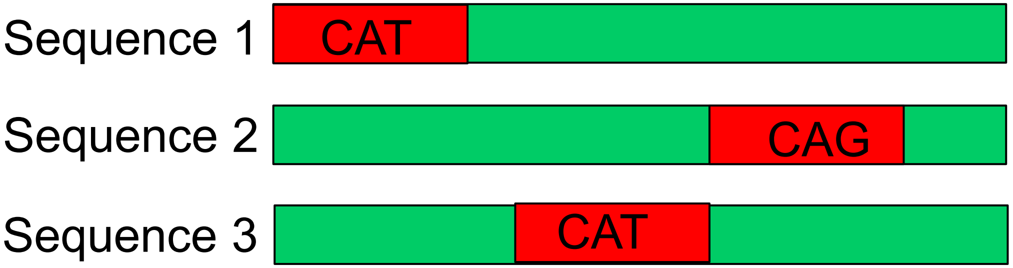
	1	2	3
A	1/7	3/7	1/7
C	3/7	1/7	1/7
G	2/7	1/7	2/7
T	1/7	2/7	3/7

- Start by placing a motif instance randomly for each sequence
- Create a motif matrix

Random Initialization + Iterative Batch Greedy Updates

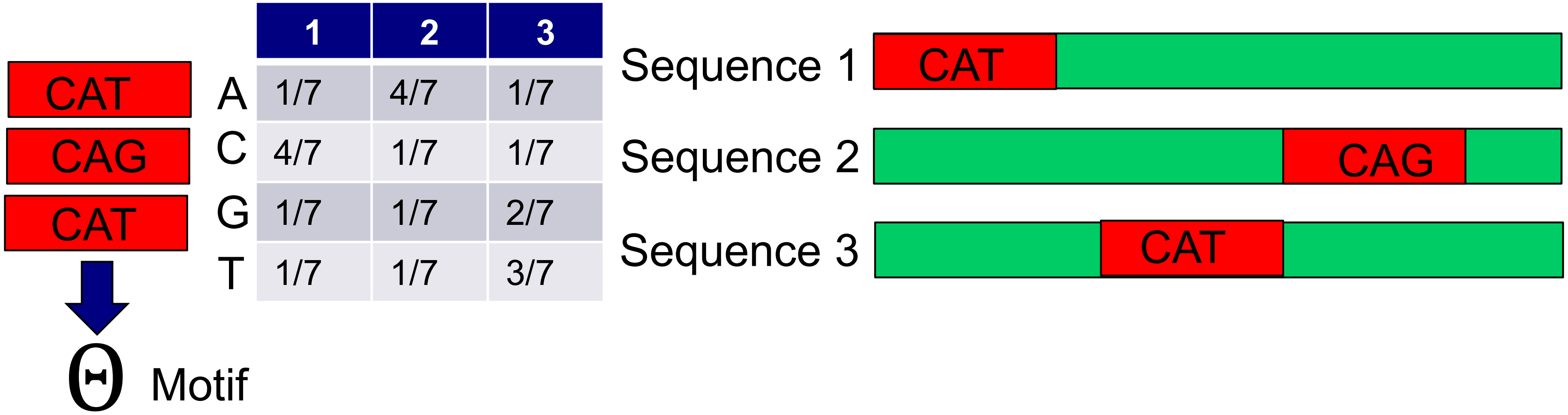


	1	2	3
A	1/7	3/7	1/7
C	3/7	1/7	1/7
G	2/7	1/7	2/7
T	1/7	2/7	3/7



- Start by placing a motif instance randomly for each sequence
- Create a motif matrix
- Update motif instances to be highest score based on current motif

Random Initialization + Iterative Batch Greedy Updates



	1	2	3
A	1/7	4/7	1/7
C	4/7	1/7	1/7
G	1/7	1/7	2/7
T	1/7	1/7	3/7

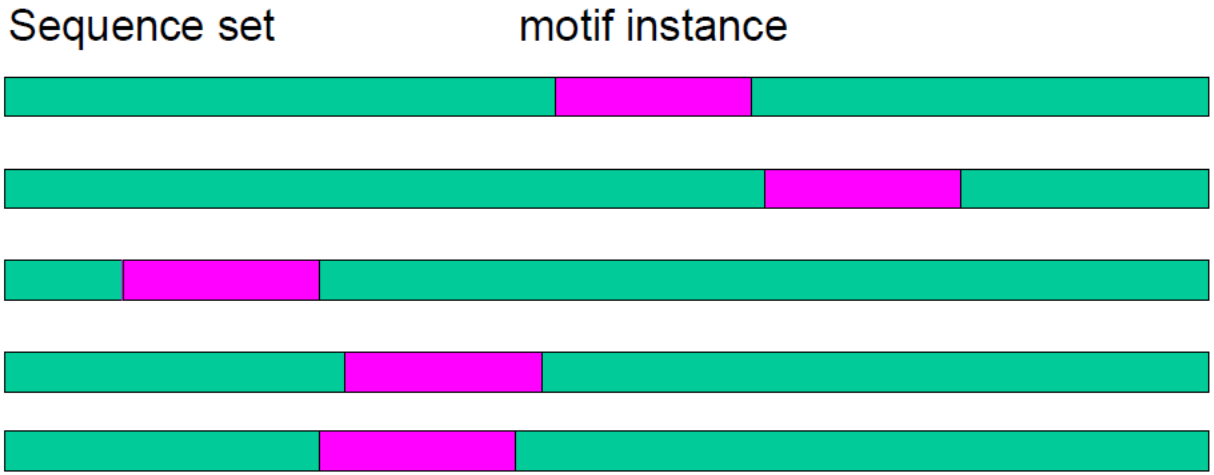
- Start by placing a motif instance randomly for each sequence
- Create a motif matrix
- Update motif instances to be highest score based on current motif
- Update motif based on current motif instances
- Iterate until convergence
- Repeat for multiple different initializations

Optimization problem

- We need to find a motif instance from each sequence and corresponding motif
- What are some brute force strategies we could use for this problem?
- What are some limitations/other strategies?

Gibbs Sampling Algorithm I

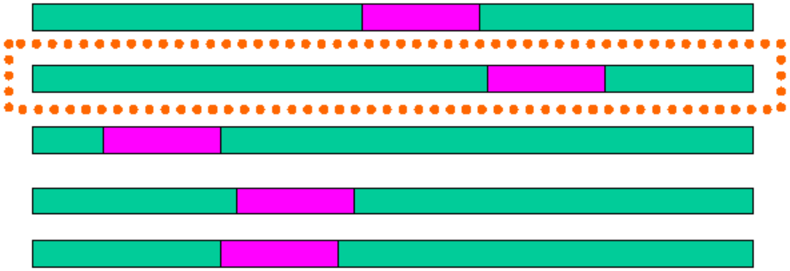
1. Select a random position in each sequence



Example modified from Chris Burges

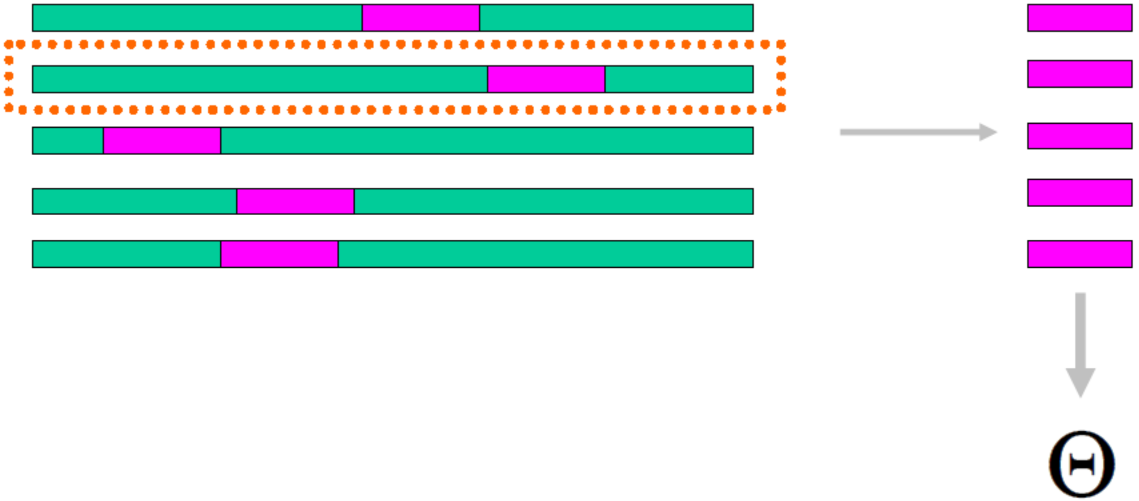
Gibbs Sampling Algorithm II

2. Select a sequence at random



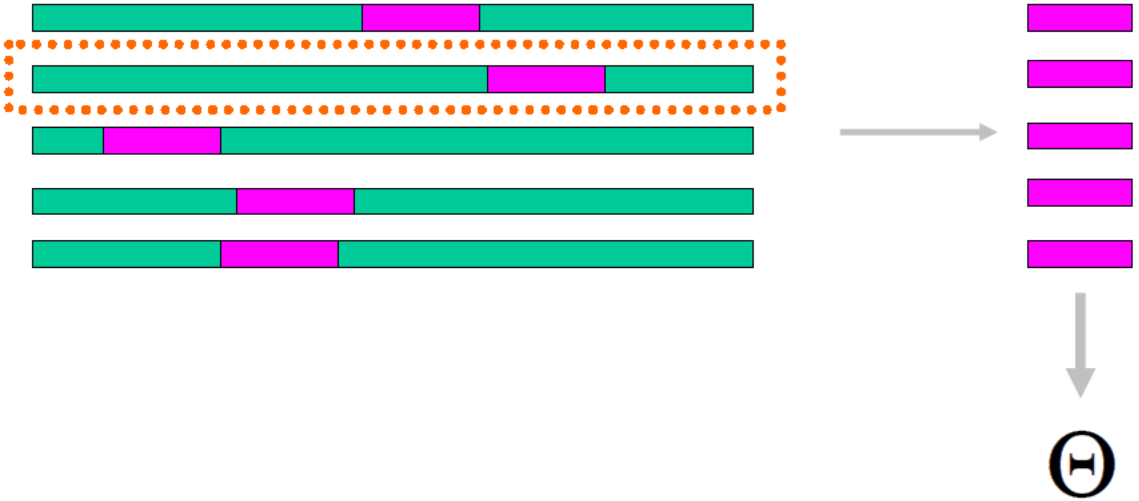
Gibbs Sampling Algorithm III

3. Select a sequence at random



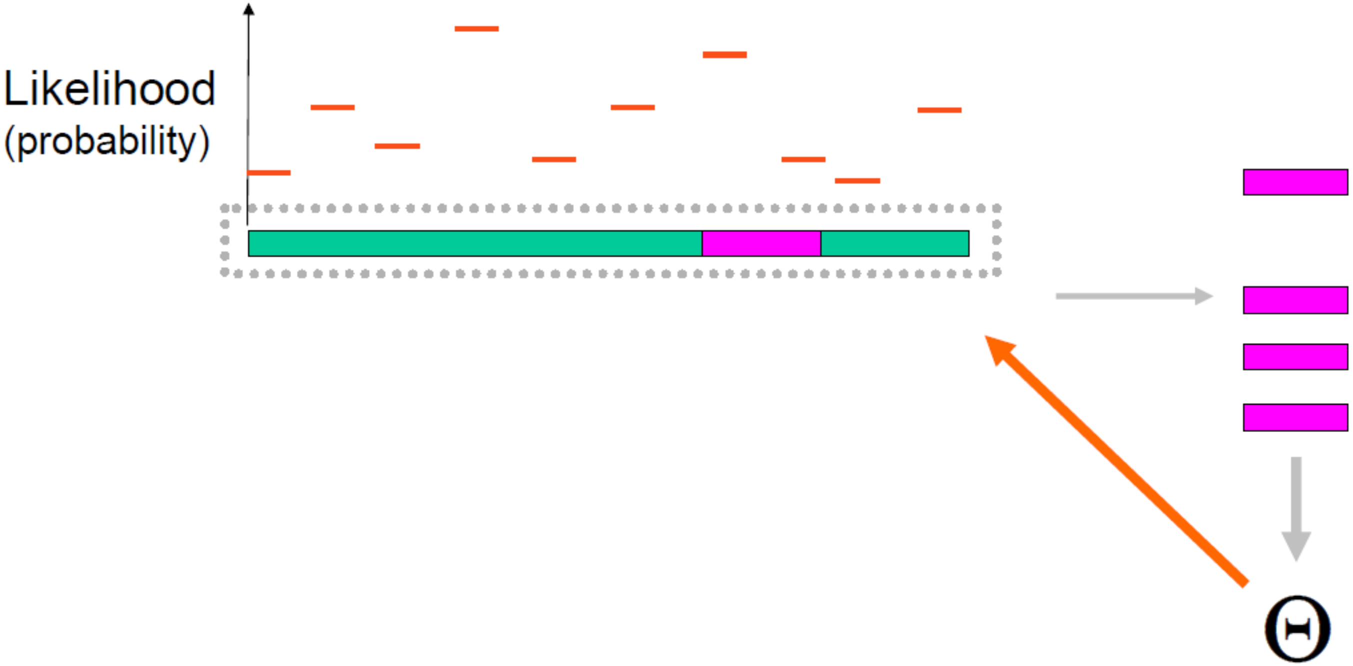
Gibbs Sampling Algorithm III

3. Select a sequence at random



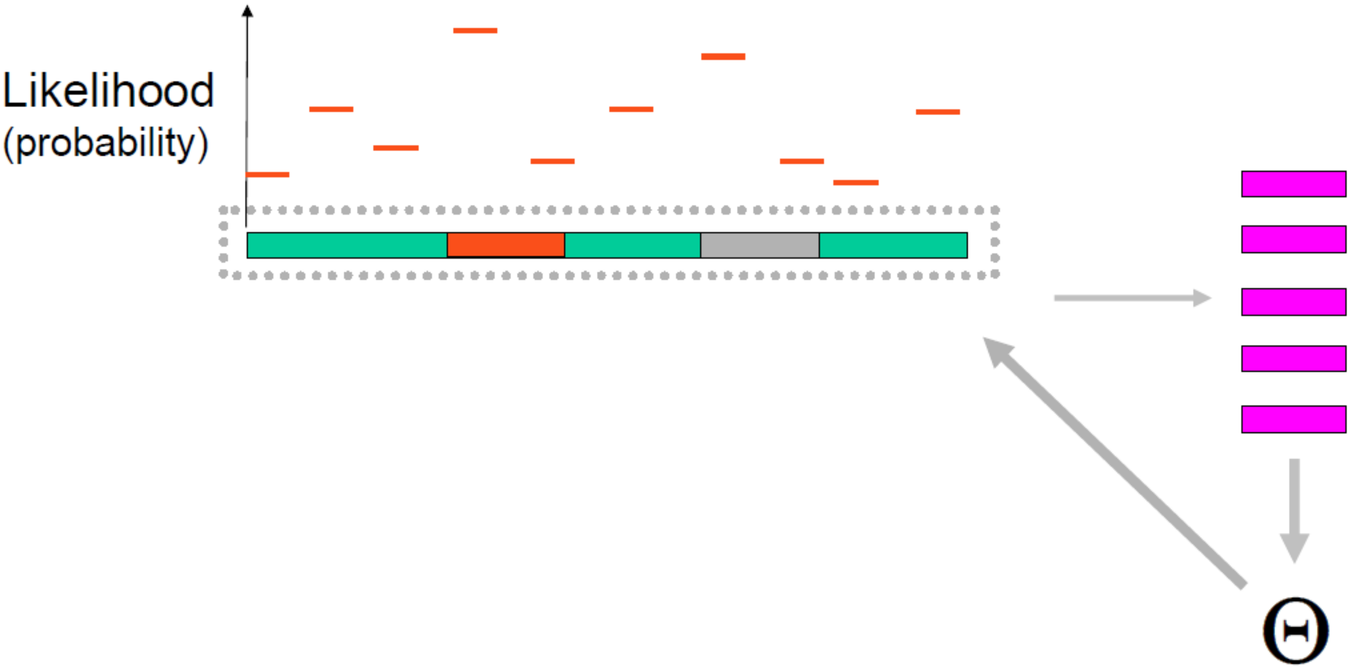
Gibbs Sampling Algorithm IV

4. Score possible sites in seq using weight matrix



Gibbs Sampling Algorithm V

5. Sample a new site proportional to likelihood



Gibbs Sampling Algorithm VI

6. Iterate until convergence (no change in sites or minimal change in Θ)