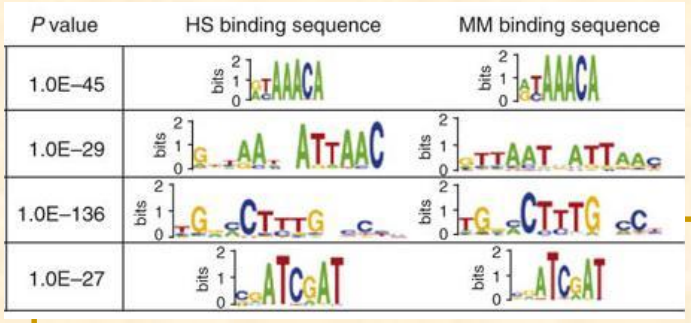
Discovering binding site motifs

Yaron Orenstein

Simons Institute Bootcamp

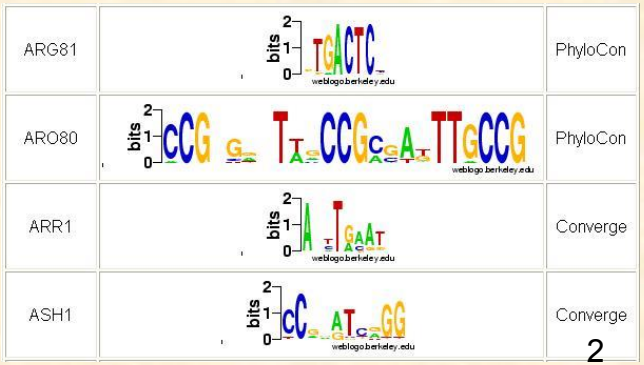
Gene regulation mini-course

January 21st 2016



Part I: de novo motif discovery using EM

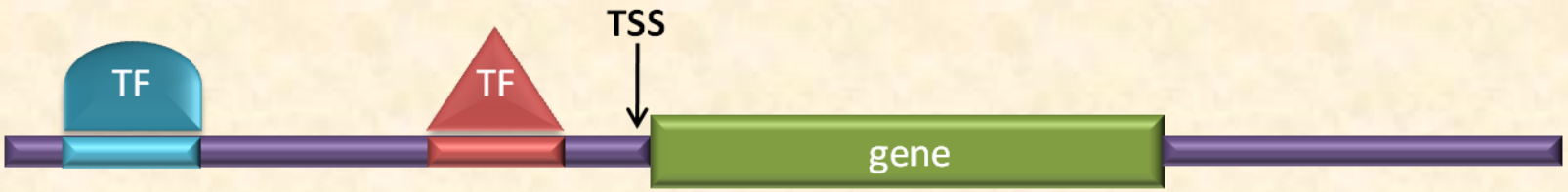
Slides sources:
Computational genomics, TAU
Prof. Roded Sharan, Prof. Ron Shamir



Gene expression regulation

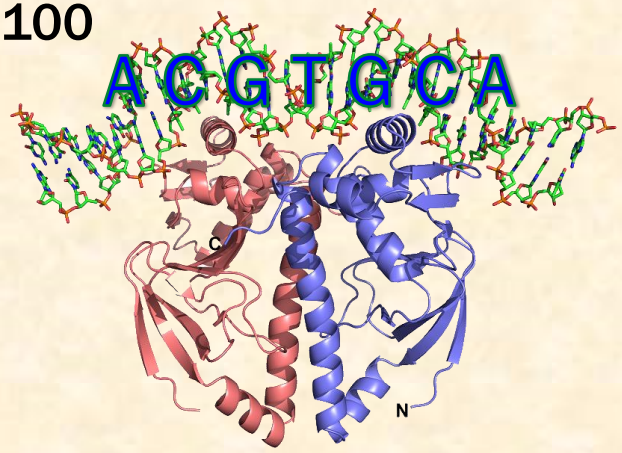
- Transcription is regulated mainly by transcription factors (**TFs**) - proteins that bind to DNA subsequences, called binding sites (**BSs**).

- TFBSs are located mainly in the gene's promoter – the DNA sequence upstream the gene's transcription start site.



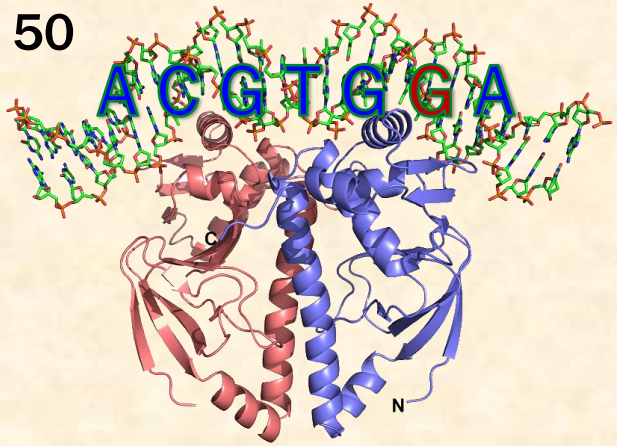
Binding preferences

100

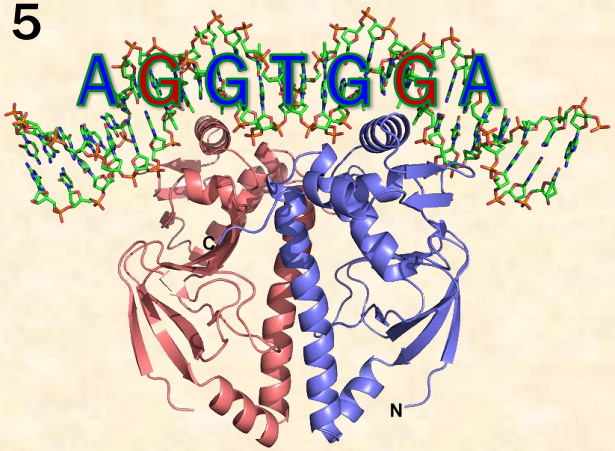


80

50



5



TFBS models

The BSs of a particular TF share a common pattern, or **motif**, which is often modeled using:

- **Consensus string**

  GGAATT

  with up to d mismatches

d=1
ATCGGAATTCTGCAG
GGCAATTCGGGAATG
AGGTATTCTCAGATTA

- **Degenerate string**

  GGWATB (W={A,T}, B={C,G,T})

ATCGGAATTCTGCAG
GGCAATTCGGGAATG
AGGTATTCTCAGATTA

TFBS models

- **PWM = Position weight matrix**

	1	2	3	4	5	6
A	0.1	0.8	0	0.7	0.2	0
C	0	0.1	0.5	0.1	0.4	0.6
G	0	0	0.5	0.1	0.4	0.1
T	0.9	0.1	0	0.1	0	0.3

➢Cutoff = 0.009

AGCTACACCCATTTAT 0.06
AGTAGAGCCTTCGTG 0.06
CGATTCTACAATATGA 0.01

- **K-mer model**

- other variants:

  DNA shape features

  di-nucleotides

```
ATGATTAATTGC     1.00
GCAATTAATCAT     0.96
GTAATTAATCAT     0.94
ATGATTAATCAT     0.91
GTCATTAATCAT     0.91
ATGATTAATGAC     0.91
AATGATTAATGA     0.88
ATGATTAATGGC     0.88
AATGATTAATGG     0.86
AATGATTAATTG     0.86
...
...
```

Motif profile

- Given the binding sites, the motif profile is easily constructed.

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

```
                A  3 0 1 0 3 1 1 0
Profile         C  2 4 0 0 1 4 0 0
                G  0 1 4 0 0 0 3 1
                T  0 0 0 5 1 0 1 4
```

```
Consensus       A C G T A C G T
```

- Line up the patterns by their start indexes

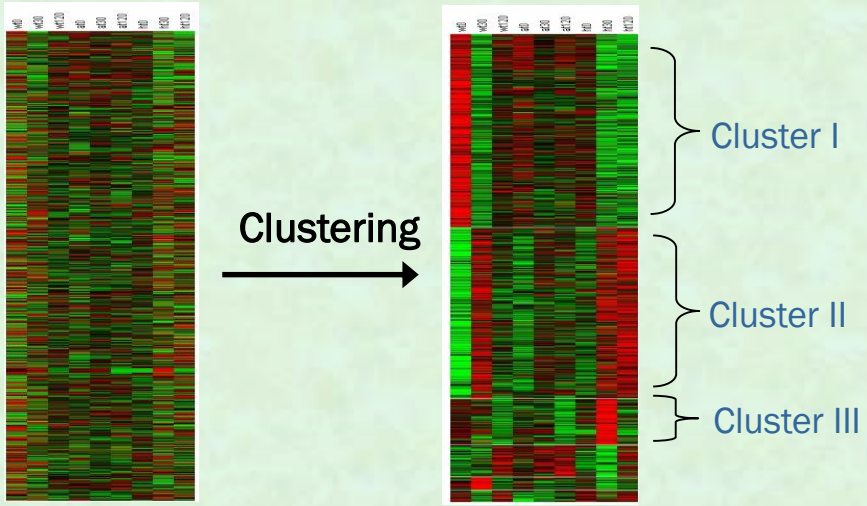
$$\mathbf{s} = (s_1, s_2, \ldots, s_t)$$

- Construct matrix profile with frequencies of each nucleotide in columns

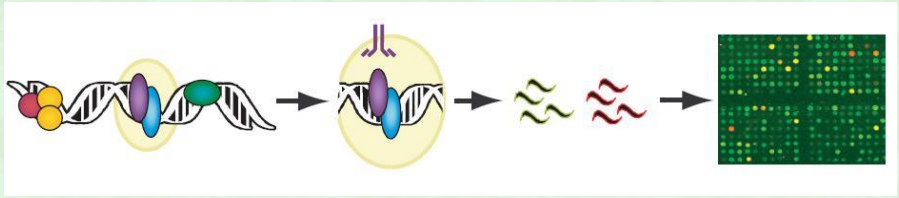
Motif discovery

Co-regulated gene set

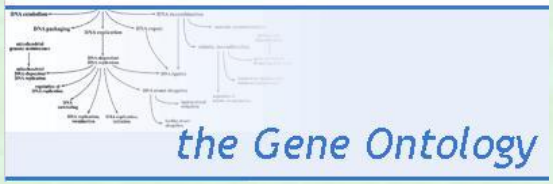
Gene expression microarrays



Location analysis (ChIP-chip, ...)



Functional group (e.g., GO term)



Promoter/3'UTR sequences

Motif discovery

Motif discovery: Goals and challenges

- **Goal:** Given a set of co-regulated genes, find a recurrent motif in their promoter regions.

- **Challenges:**
  - **BSs** are short and degenerate (**non-specific**)
  - **Promoters** are long + complex (**hard to model**)
  - **Search space** is huge (**motif and sequence**)
  - **Data** is noisy

An example: Implanting Motif AAAAAAAGGGGGGG

atgaccgggatactgatAAAAAAAAGGGGGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataAAAAAAAAGGGGGGGa

tgagtatccctgggatgacttAAAAAAAAGGGGGGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgAAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAAGGGGGGGcttatag

gtcaatcatgttcttgtgaatggatttAAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtAAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAAGGGGGGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAAGGGGGGGa

Where is the Implanted Motif? (*)

atgaccgggatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataaaaaaaaggggggga

tgagtatccctgggatgacttaaaaaaaaggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgaaaaaaaaggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggcttatag

gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtaaaaaaaaggggggcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttaaaaaaaaggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga

Implanting Motif **AAAAAAGGGGGGG** with Four Mutations

atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa

tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag

gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGCcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa

Where is the Motif???

atgaccgggatactgatagaagaaaggttgggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacaataaaacggcggga

tgagtatccctgggatgacttaaaataatggagtggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgcaaaaaaagggattgtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatataataaaggaagggcttatag

gtcaatcatgttcttgtgaatggatttaacaataagggctgggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtataaacaaggagggccaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttaaaaaatagggagccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttactaaaaaggagcgga

MEME

Multiple EM for Motif Elicitation

[Bailey, Elkan ISMB '94]

Goal: Given a set of sequences, find a motif (PWM) that maximizes the expected likelihood of the data

Technique: EM (Expectation Maximization)

(based on [Lawrence, Reilly '90])

EM reminder: the goal

- Input: incomplete data originating from a probability distribution with some unknown parameters
- Want to find the parameter values that maximize the likelihood
- EM – approach that helps when maximum likelihood solution cannot be directly computed.
- Seeks a local maximum by iteratively solving two easier subproblems

EM-reminder: the setting

Input: data X coming from a probabilistic model with hidden information y

Goal: Learn the model's parameters θ so that the likelihood is maximized.

EM-reminder: the algorithm

Main component:

$$Q(\theta \mid \theta^t) = \sum_y P(y \mid x, \theta^t) \cdot \log P(x, y \mid \theta)$$

log P(x,y|θ) is called the complete log likelihood function

➔ Q is the expectation of the complete log likelihood over the distribution of y given the current parameters θ^t

The algorithm:

repeat

- **E-step: Calculate the Q function**
- **M-step: Maximize $Q(\theta|\theta^t)$ with respect to θ**
- **Stopping criterion: improvement in log likelihood ≤ ε**

Note: local optimum guaranteed to be reached, not global.
Starting point matters! Try many..

MEME: The Mixture Model

Data: $X = (X_1,\ldots,X_n)$:

all (overlapping) *l*-mers in the input sequences

Assume X_i's were generated by a two-component mixture model - $\theta = (\theta_1, \theta_2)$:

Model #1: θ_1 = motif model:

$f_{i,b}$ = prob. of base *b* at pos *i* in motif, $1 \le i \le l$

Model #2: θ_2 = background (BG) model:

$f_{0,b}$ = prob. of base *b*

Mixing parameter: $\lambda = (\lambda_1, \lambda_2)$

λ_j = prob. that model #*j* is used $(\lambda_1+\lambda_2=1)$

Assume independence between *l*-mers

Log Likelihood

Missing data: $Z = (Z_1, \ldots, Z_n)$:

$Z_i = (Z_{i1}, Z_{i2})$; $Z_{ij} = 1$ if X_i from model #j ; 0 o/w

Complete Likelihood of model given data:

$$L(\theta, \lambda \mid X, Z) = p(X, Z \mid \theta, \lambda)$$

$$= \Pi_{i=1\ldots n}\, p(X_i, Z_i \mid \theta, \lambda)$$

$$p(X_i, Z_i \mid \theta, \lambda) = p(X_i \mid Z_i, \theta, \lambda)\, p(Z_i \mid \theta, \lambda) =$$

$$= \lambda_1\, p(X_i \mid \theta_1) \text{ if } Z_{i1}=1;\ \lambda_2\, p(X_i \mid \theta_2) \text{ if } Z_{i2}=1$$

$$\rightarrow \log L = \Sigma_{i=1\ldots n} \Sigma_{j=1,2} Z_{ij} \log(\lambda_j\, p(X_i \mid \theta_j))$$

MEME: Algorithm

Goal: Maximize $E[\log L]$

Outline of EM algorithm:

- Choose starting θ^0, λ^0
- Repeat until convergence of θ:
  - E-step: Re-estimate Z from θ^t, λ^t, X
  - M-step: Re-estimate θ^{t+1}, λ^{t+1} from X, Z
- Repeat all of the above for various θ, λ …

E-step

Compute expectation of *log L* over Z:

$$E_Z[\log L] = \Sigma_{i=1\ldots n} \Sigma_{j=1,2} Z'_{ij} \log (\lambda_j \, p(X_i | \theta_j))$$

where:

$$Z'_{ij} = p(Z_{ij}=1 | \, \theta^t, \lambda^t, X_i) =$$

$$= p(Z_{ij}=1, X_i | \, \theta^t, \lambda^t) \, / \, p(X_i | \, \theta^t, \lambda^t) =$$

$$= p(Z_{ij}=1, X_i | \, \theta^t, \lambda^t) \, / \, \Sigma_{k=1,2} \, p(Z_{ik}=1, X_i | \, \theta^t, \lambda^t) =$$

$$= \lambda^t_j \, p(X_i | \theta^t_j) \, / \, \Sigma_{k=1,2} \lambda^t_k \, p(X_i | \theta^t_k)$$

M-step

Find θ,λ that maximize $E[\log L]=Q(\theta,\lambda \mid \theta^t, \lambda^t)$:

$$E[\log L] = \Sigma_{i=1\ldots n} \Sigma_{j=1,2} Z'_{ij} \log (\lambda_j p(X_i \mid \theta_j))$$

Finding λ:

Suffices to maximize $L_1 = \Sigma_{i=1\ldots n} \Sigma_{j=1,2} Z'_{ij} \log \lambda_j$

$\lambda_1+\lambda_2=1 \rightarrow L_1 = \Sigma_{i=1\ldots n} (Z'_{i1} \log \lambda_1 + Z'_{i2} \log (1-\lambda_1))$

$dL_1/d\lambda_1 = \Sigma_{i=1\ldots n} (Z'_{i1} / \lambda_1 - Z'_{i2} / (1-\lambda_1))$

MEME: Algorithm

M-step (cont.):

$dL_1/d\lambda_1 = \Sigma_{i=1...n}(Z'_{i1} / \lambda_1 - Z'_{i2} / (1-\lambda_1)) = 0$

$\rightarrow \lambda_1 \Sigma_{i=1...n} Z'_{i2} = (1-\lambda_1) \Sigma_{i=1...n} Z'_{i1}$

$\rightarrow \lambda_1 (\Sigma_{i=1...n}(Z'_{i1}+Z'_{i2})) = \Sigma_{i=1...n} Z'_{i1}$

$\rightarrow \lambda_1 = (\Sigma_{i=1...n} Z'_{i1}) / n$

$\lambda_2 = 1 - \lambda_1 = (\Sigma_{i=1...n} Z'_{i2}) / n$

Finding θ: $\theta_{jk} = C_{jk} / \Sigma_{k\epsilon\{A,C,G,T\}} C_{jk}$

$C_{jk} = \Sigma_{i=1...n} Z'_{i1} I(k, X_{ij})$

Expected #times that letter *k* is produced at column *j*

Multiple passes

- To find multiple motifs: After finding one, erase (mask) its positions from all sequences and rerun the alg.
- By doing several passes, a drop in the log likelihood score may indicate which are true motifs

MEME
Multiple Em for Motif Elicitation

Version 4.3.0

Use this form to submit DNA or protein sequences to MEME. MEME will analyze your sequences for similarities among them and produce a description (**motif**) for each pattern it discovers.

Data Submission Form

Required

Your **e-mail address:**

Re-enter **e-mail address:**

How do you think the occurrences of a single motif are **distributed** among the sequences?
- **One per sequence**
- **Zero or one** per sequence
- **Any number** of repetitions

Please enter the **sequences** which you believe share one or more motifs. The sequences may contain no more than **60000 characters** total in any of a large number of **formats**.

MEME will find the optimum **width** of each motif within the limits you specify here:

6 **Minimum** width (>= 2)

50 **Maximum** width (<= 300)

Enter the **name of a file** containing the sequences here:

Browse... **Clear**

3 Maximum **number of motifs** to find

or

the **actual sequences** here (**Sample Protein Input Sequences**):

Optional

Description of your sequences:

MEME will find the optimum **number of sites** for each motif within the limits you specify here:

Minimum sites (>= 2)

Maximum sites (<= 300)

Enter the name of a file containing a **background Markov model**:

Browse... **Clear**

DNA-ONLY OPTIONS
(Ignored for protein searches)

Shuffle sequence letters

Search given **strand** only

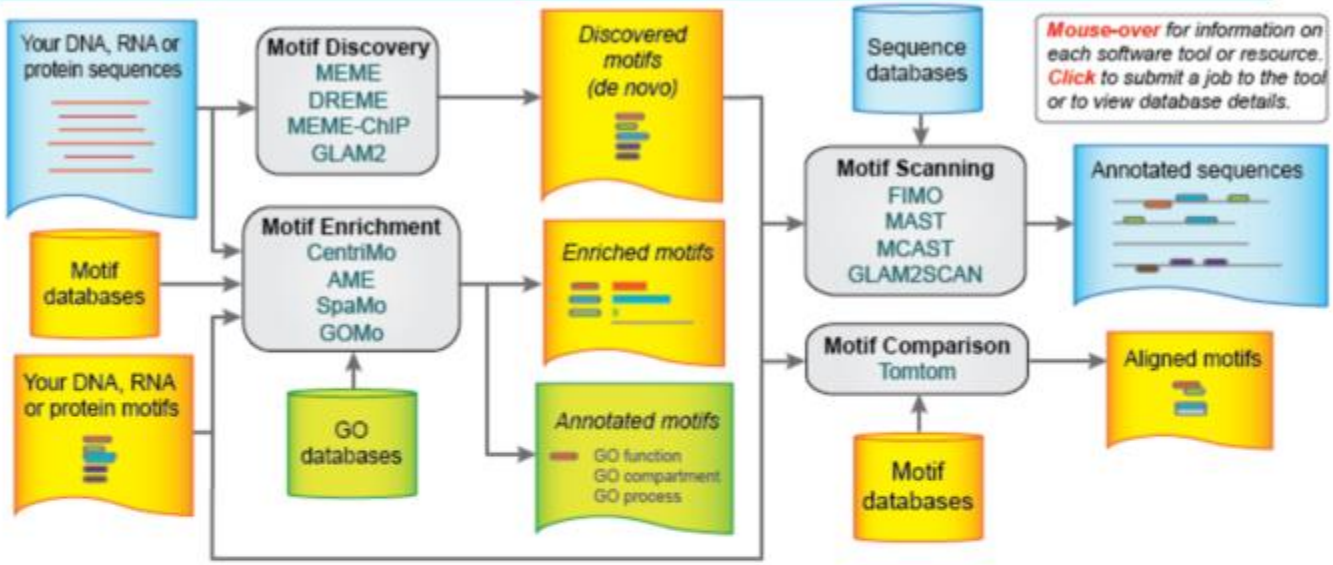
Look for **palindromes** only

Start search Clear Input

Version 4.3.0 **Please send comments and questions to: meme@nbcr.net** **Powered by Opal**

The MEME Suite

Motif-based sequence analysis tools



MEME — Multiple Em for Motif Elicitation

CentriMo — Local Motif Enrichment Analysis

FIMO — Find Individual Motif Occurrences

DREME — Discriminative Regular Expression Motif Elicitation

AME — Analysis of Motif Enrichment

MAST — Motif Alignment & Search Tool

MEME-ChIP — Motif Analysis of Large Nucleotide Datasets

SpaMo — Spacing Motif Analysis Tool

MCAST — Motif Cluster Alignment and Search Tool

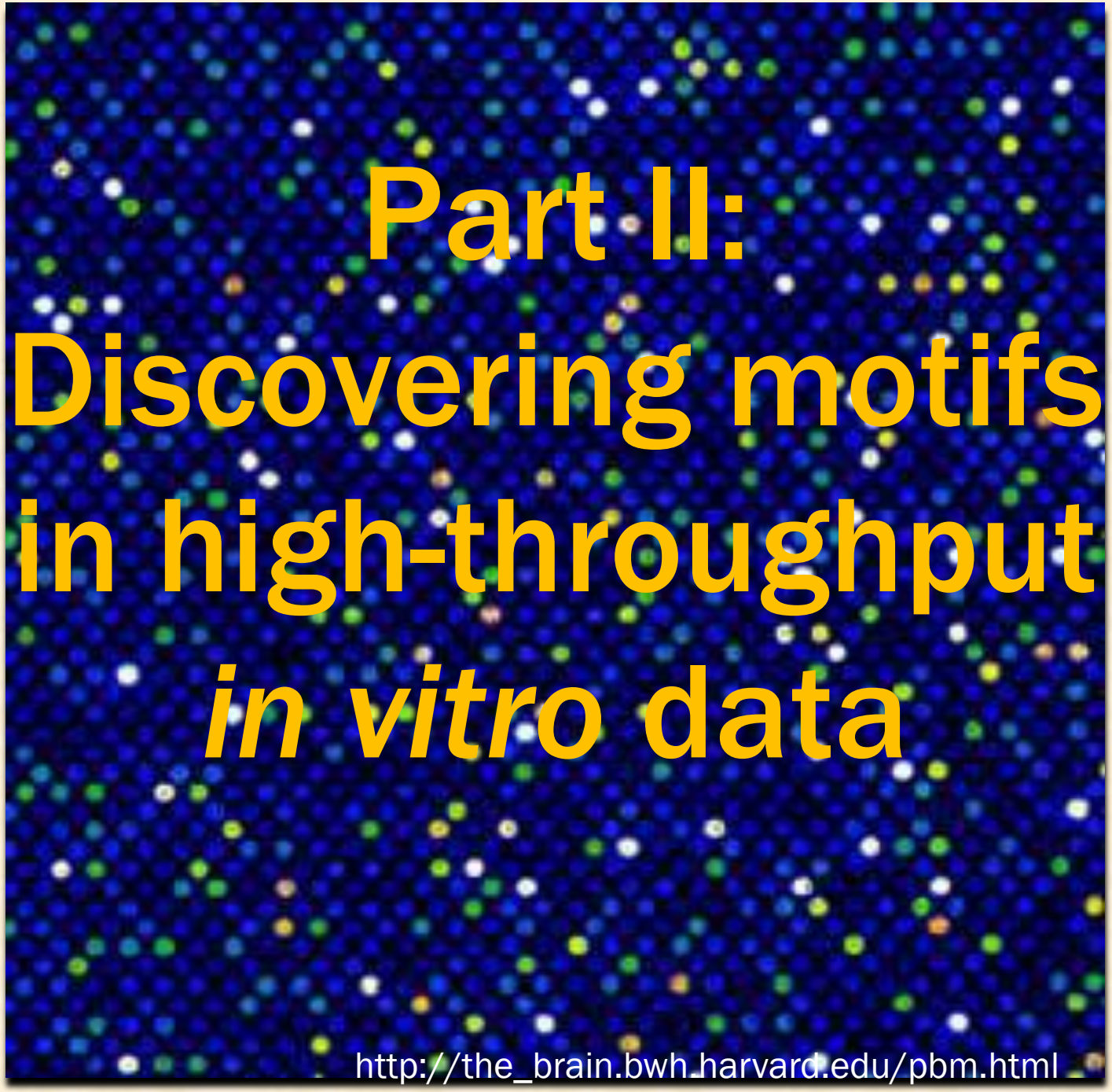
GLAM2 — Gapped Local Alignment of Motifs

GOMo — Gene Ontology for Motifs

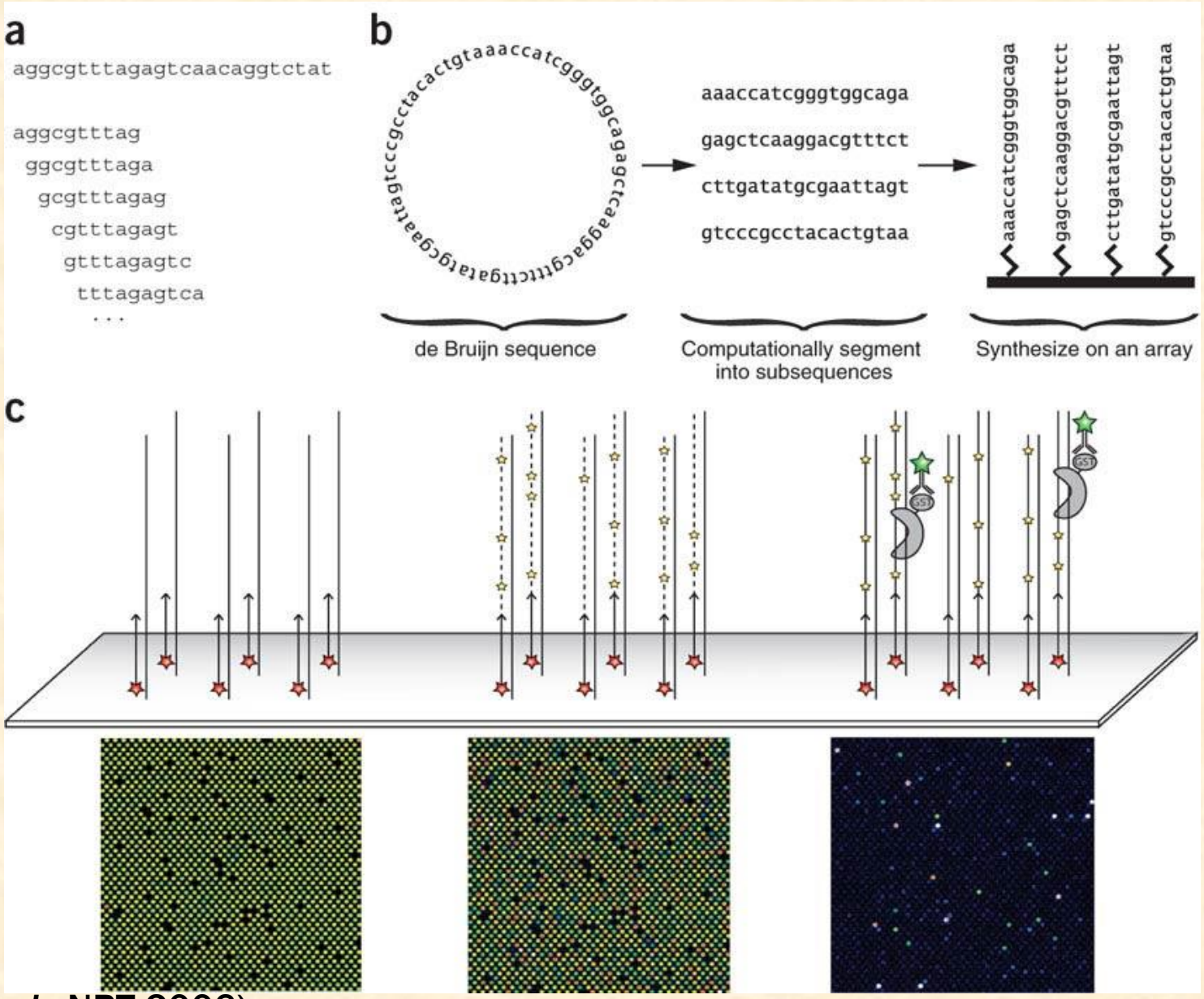
GLAM2Scan — Scanning with Gapped Motifs

Tomtom — Motif Comparison Tool

GT-Scan — Identifying Unique Genomic Targets


Part II:
Discovering motifs
in high-throughput
in vitro data
http://the_brain.bwh.harvard.edu/pbm.html

Protein binding microarrays



(Berger *et al.*, NBT 2006)

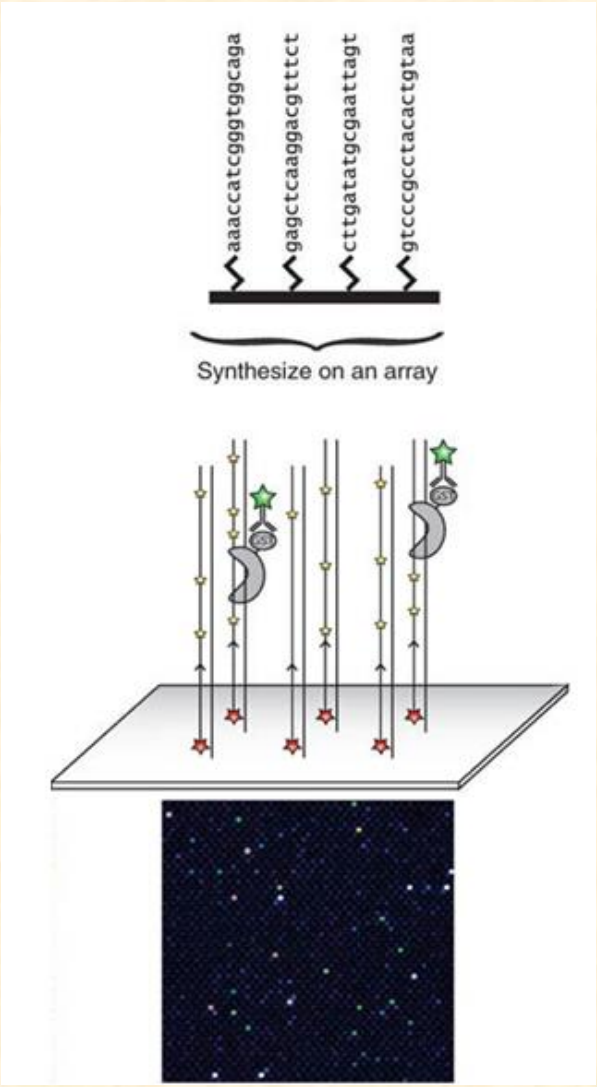
The data

- Around 41,000 probes, each 36bp-long.
- Plus, corresponding binding intensities.

936656.772613291	AAATTTGATACAACTGCGTCTTGAGCTCTAATTCAC
908113.295168776	TTTTTGATTTAGCCGGGTCAATCCTCCAATATCCTA
897768.671357285	TTTTTGATAAGAATAATTACATAAGAGTGAAAACAT
889610.449301355	ATATATATATTGGTAAAGAGCAATTAATCTTCGCTG
788786.81899799	GGTTTGATACACTCAGTTGTTATAAGAGCGTAGGCA
758147.585093369	GAAGCTTAATTAAATCCTGCACTACGCCTGTTTTCA
748029.092176931	TAGTTTGATTCGACTGCAATTTGTTTCCCCTGCCTT
740301.561188748	GTTTTGATAGGGGTATGCAGAGTCTCTTTCTCCGTT
...	
...	
979.99239669736	GAAGGAGCAGCCCCTGAGGGCGGCTCGCATAACTGG
973.610994073221	TCTCTAGGAGTCAGGGCTGTCCTCGCATACCGTGCC
964.88923536786	GAGCGTCAGCGCGCGGTCACTTCATGTATGGTCGCC
577.174517674586	AGGGATTCTTGGCTGCGGCCTGTATCGCTGTTGGAA
552.401117184654	GGCGTGGCTTGTTACCTCATGCAGAAGACCGGTAAG
-218.900301178986	CGTGTACTTCAAACGCGGAGTTGGCATATGCGGTCC
-288.772718876298	GGAGGTCAACCACGGTCAGTCCATGTGAAGTCCATA
-2830.13880248726	ACCGGTCGCAGTTGGCCTTCCGCTGATAAGCATTGC

(**UniPROBE database, Robasky and Bulyk, *NAR* 11**)

Computational challenge



(Berger *et al.*, NBT 2006)

experiment

AGCTCGAGTAG	208
GTTGATGCATT	134
GCTAGCCAGGA	10
CGGCTAGCGCC	45
GCATCGCTGCG	65
CGATCGCTCGA	544
AGCTCGAGGCG	414
GTTGATGCAGC	13
GCTAGCCAGGC	51
CGGCTAGCGAG	62
GCATCGCTGTT	102
CGATCGCTCGA	54

Computational analysis

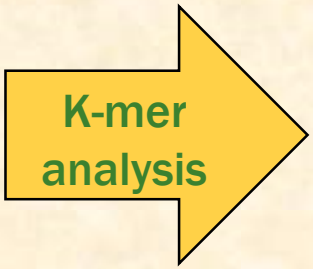
	1	2	3	4	5
A	0.1	0.8	0	0	0
C	0.9	0.1	0.9	0.5	0.1
G	0	0	0.1	0.4	0
T	0	0.1	0	0.1	0.9

RAP algorithm

Input: a list of sequences and scores.

Output: a binding model.

AGCTCGAGTAG	10
GTTGATGCATT	5
GCTAGCCAGGA	20
CGGCTAGCGCC	25
GCATCGCTGCG	35
CGATCGCTCGA	50
AGCTCGAGGCG	70
GTTGATGCAGC	30
GCTAGCCAGGC	45
CGGCTAGCGAG	60
GCATCGCTGTT	100
CGATCGCTCGA	40



Ranked 4-mer list

TCGA	42.5
GCTC	42.5
CCAG	32.5
AGCT	40
…	

(Orenstein, Mick and Shamir, *JCB* 2013)

RAP

RANK ALIGN PWM

TAATGCGA 150

ATAATCGGG 132

AAATGCGAC 98

GATAATCGC 95

TAATGTGC 85

AAATGCGCC 76

TAATGTAC 74

ATATGTACC 71

Weight

				0.95			
					0.95		
			0.9				
		0.85					

PWM

BEEML-PBM

(Binding Energy Estimation by Maximum Likelihood for PBMs)

- Follows a biomechanical model of TF binding to sequence S_i:

$$TF + S_i \underset{k_{off}}{\overset{k_{on}}{\leftrightarrow}} TF \bullet S_i$$

- K_{on} and K_{off} = on- and off-rates.

$$K_d = \frac{K_{off}}{K_{on}} = \frac{[S_i][TF]}{[TF \bullet S_i]}$$

(Zhao and Stormo, *NBT* 2011)

BEEML-PBM model

- Probability of binding in equilibrium is:

$$P(S_i) = \frac{[TF \bullet S_i]}{[S_i] + [TF \bullet S_i]} = \quad = \quad = \quad =$$

- E_i = free energy difference between S_i and S_{ref}.

$$\mu = \frac{K_d(S_{ref})}{[TF]}$$

BEEML-PBM model

- Energy contribution by PWM model ε:

$$E_i = \prod_{k=1}^{l} \varepsilon(S_i(k), k)$$

- l = motif length.
- $S_i(k)$ = letter k of S_i
- $\varepsilon(b, k)$ = energy contribution of b in position k.

BEEML-PBM model

- Binding probability to double-stranded S_i :

$$F(S_i) = P(S_i) + (1 - P(S_i))P(S_i')$$

- S_i' = reverse complement of S_i.

- Binding probability to sequence *T*:

$$B(T) = \sum_{i=1}^{|T|-l+1} F(T_{i:i+l-1})$$

- $T_{i:i+l-1}$ = subseq of *T* of length *l* starting at *i*.

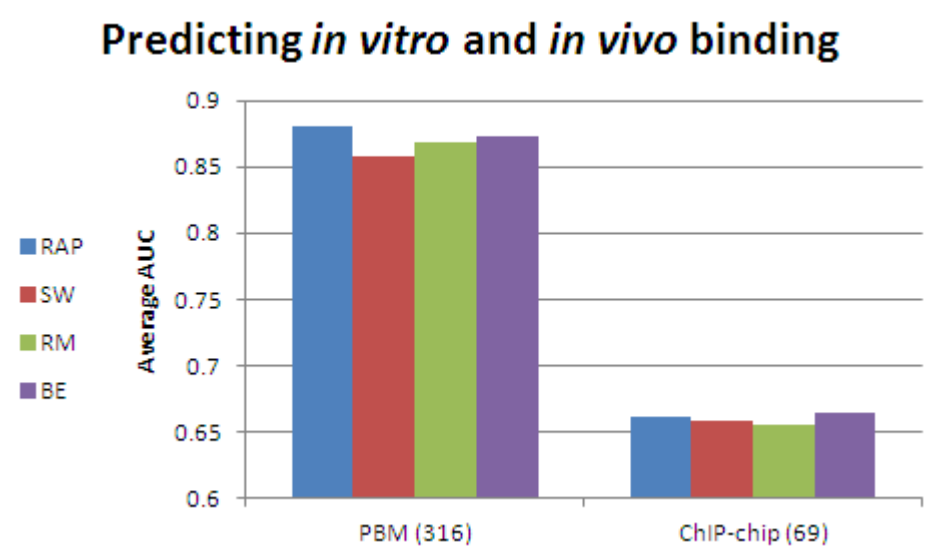
BEEML-PBM model parameters

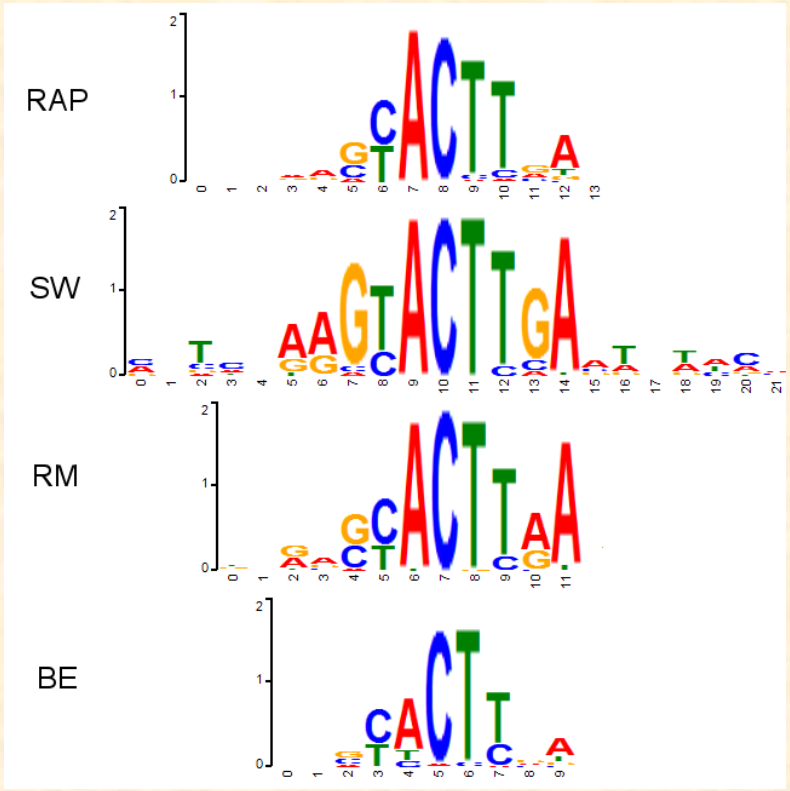
- ε = PWM, $\mu = \frac{K_d(S_{ref})}{[TF]}$
- Objective function:

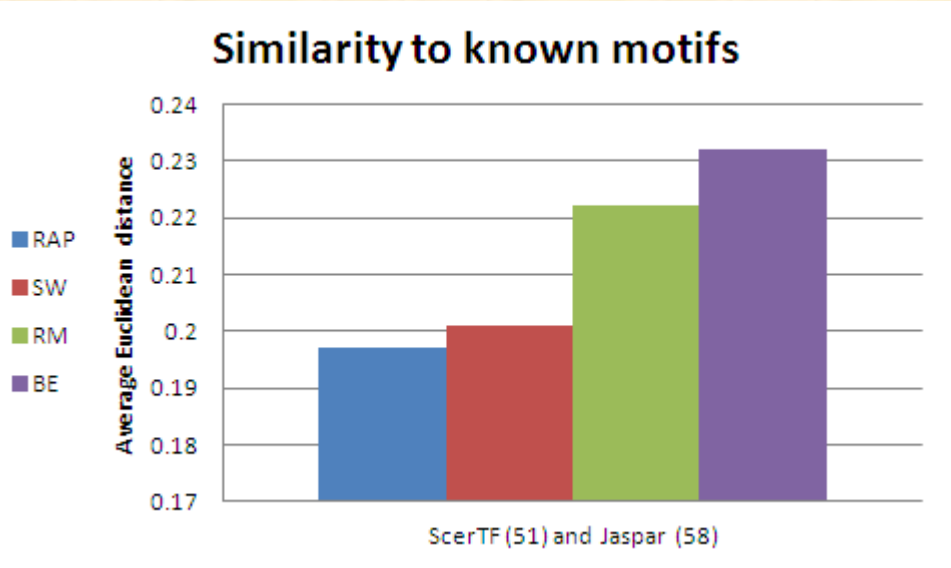
$$O(\varepsilon,\mu) = \sum_i (Y_i - a - cB(T_i))^2 + \lambda \sum_{b=A}^{T} \sum_{k=1}^{l} \varepsilon(b,k)^2$$

- Y_i =observed binding intensity.
- T_i = sequence of probe *i*.
- Minimization by Levenberg-Marquardt algorithm.

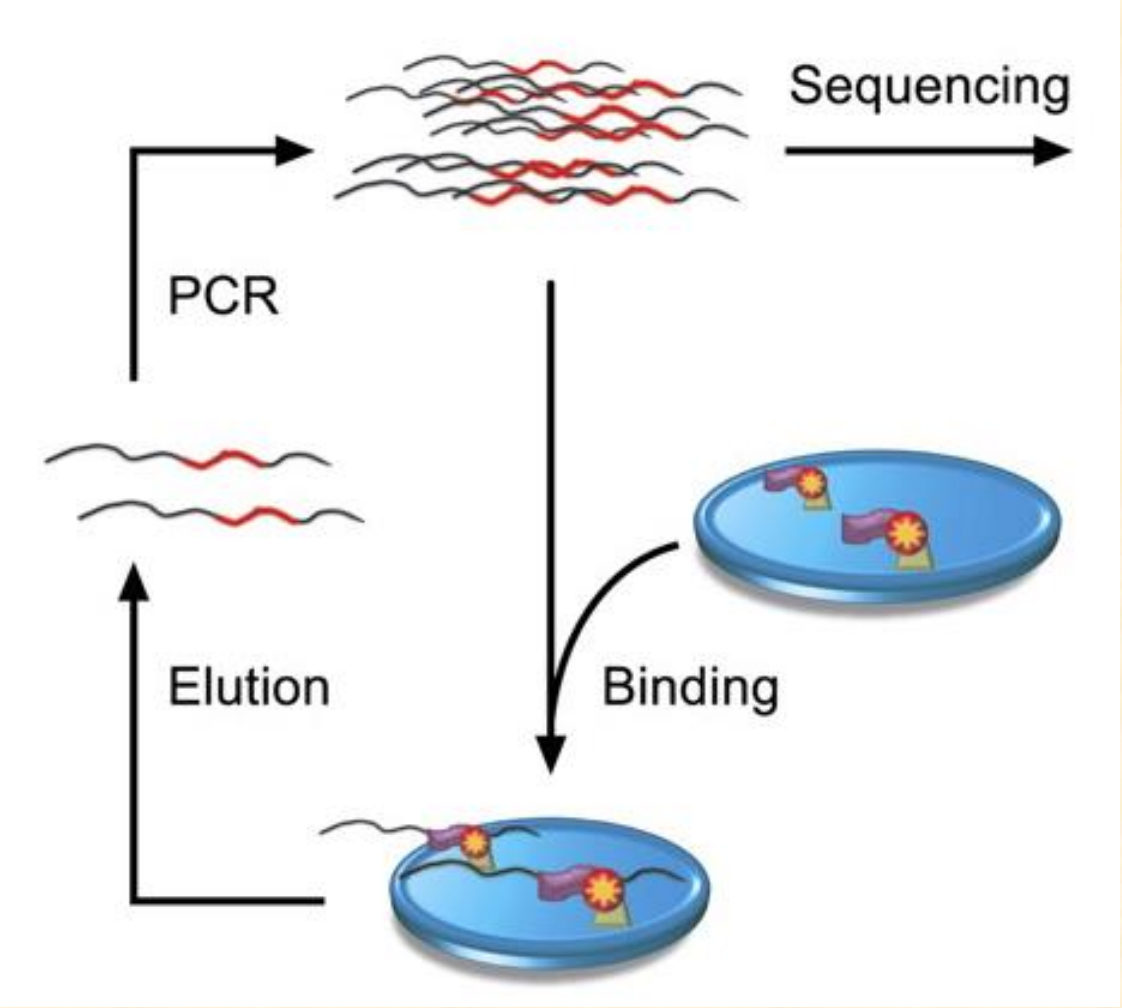
Results


Predicting *in vitro* and *in vivo* binding
RAP
SW
RM
BE
Average AUC
0.9
0.85
0.8
0.75
0.7
0.65
0.6
PBM (316)
ChIP-chip (69)

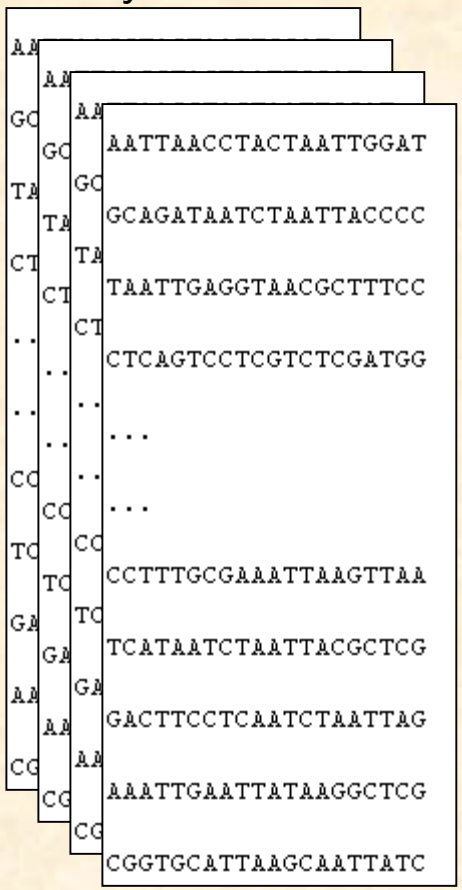

RAP
SW
RM
BE


Similarity to known motifs
RAP
SW
RM
BE
Average Euclidean distance
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
ScerTF (51) and Jaspar (58)

The next step: High-throughput SELEX



cycles 0-3



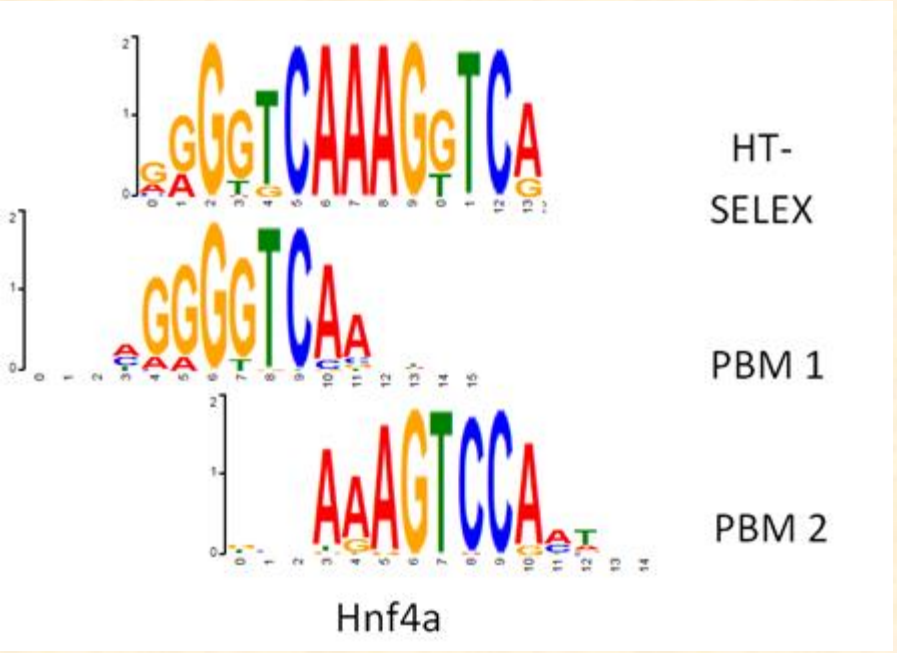
(Jolma *et al.*, *Genome Research* 10)

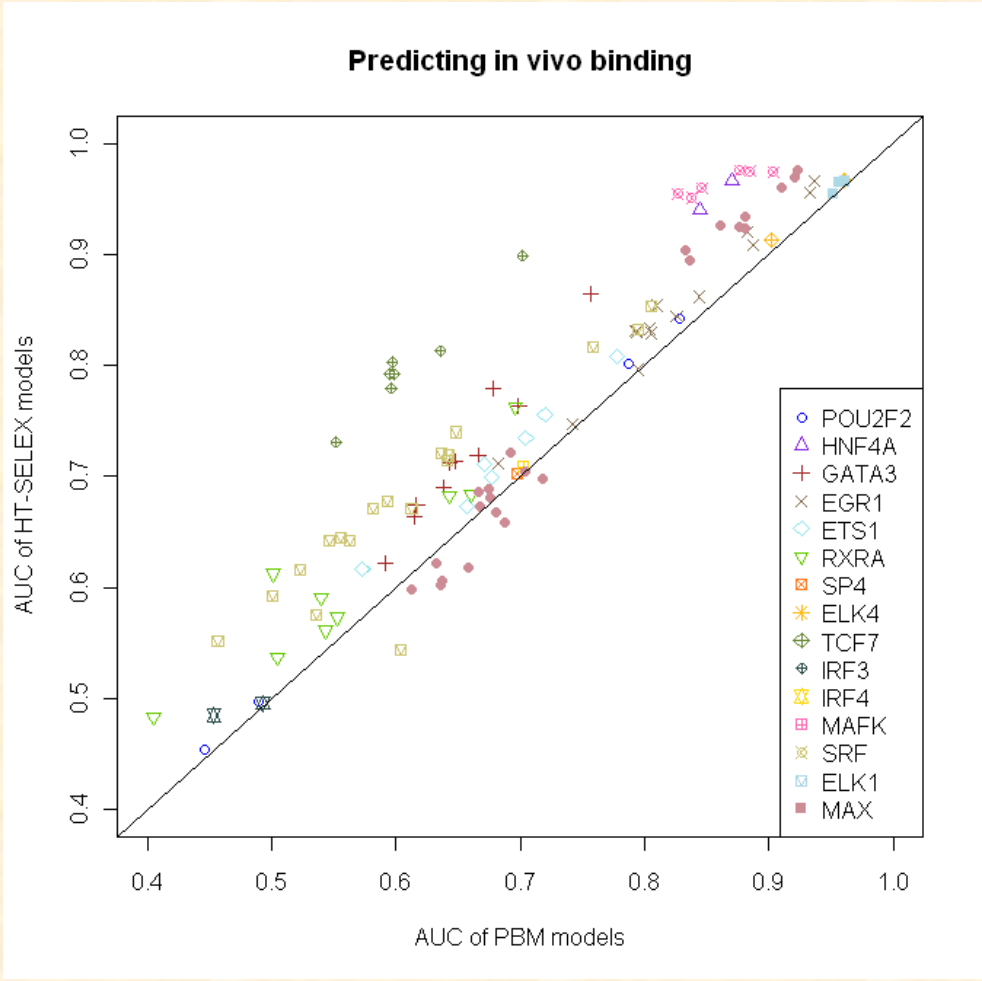
What happens through the cycles…

Cycle 0	Cycle 1	Cycle 2	Cycle 3
GTCAGTGTCAGTGACGTG	GTCAGTGTCAGTGACGTG	GTCAGTGTCAGTGACGTG	TTCACTCTCGAGGCTAGC
GCGTGACGTGAAGTCGTA	GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA
GTGCAACGTTGTTTGAGG	GTGCAACGTTGTTTGAGG	GTGCAACGTTGTTTGAGG	GTGCAACGTTGTTTGAGG
GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA
AAGGTGACCGTACGAGTC	AAGGTGACCGTACGAGTC	AAGGTGACCGTACGAGTC	AAGGTGACCGTACGAGTC
GTCCATTCACTGGTGAGT	GTCCATTCACTGGTGAGT	GTCCATTCACTGGTGAGT	GTCCATTCACTGGTGAGT
GTGCAGCGTGAACGTTGG	GTGCAGCGTGAACGTTGG	TTCACTCTCGAGGCTAGC	TTCACTCTCGAGGCTAGC
GTGATTTGGAACACCCAG	GTGATTTGGAACACCCAG	GTGATTTGGAACACCCAG	GTGATTTGGAACACCCAG
CCCCACCCACCGGCCTGC	CCCCACCCACCGGCCTGC	CCCCACCCACCGGCCTGC	TTCACTCTCGAGGCTAGC
ACAGTCAGCCCTAGCACG	TTTGACTCTGCTACGCAT	TTTGACTCTGCTACGCAT	TTTGACTCTGCTACGCAT
CACATACGCTGACTCGTA	CACATACGCTGACTCGTA	GCGATTAACTCTGTGAAA	GCGATTAACTCTGTGAAA
TTTGACTCTGCTACGCAT	TTTGACTCTGCTACGCAT	TTTGACTCTGCTACGCAT	TTTGACTCTGCTACGCAT
CGATCGATCAGGCTAGCT	CGATCGATCAGGCTAGCT	CGATCGATCAGGCTAGCT	GCGATTAACTCTGTGAAA
…	…	…	…
…	…	…	…
…	…	…	…
TTCACTCTCGAGGCTAGC	TTCACTCTCGAGGCTAGC	TTCACTCTCGAGGCTAGC	TTCACTCTCGAGGCTAGC

Comparison of HT-SELEX to PBM

(Orenstein and Shamir 2014)



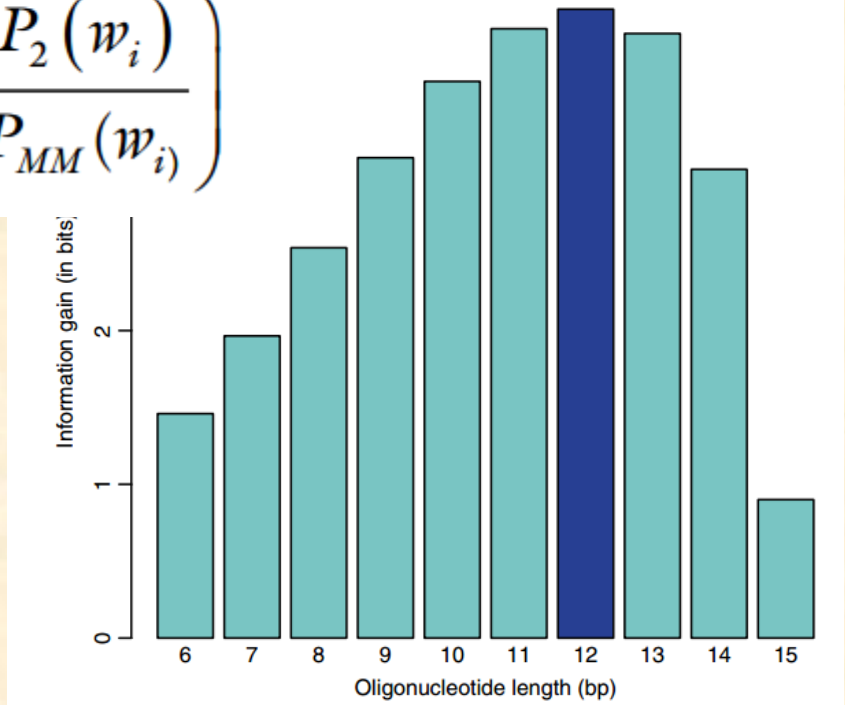


SELEX-seq
(Riley et al. 2014)

- Find k by KL-divergence score:

$$D_{CGKL}(K) = \sum_{i \in S_{100}(K)} \left(P_2(w_i) \log \frac{P_2(w_i)}{P_{MM}(w_i)} \right)$$
$$+ \left[1 - \sum_{i \in S_{100}(K)} P_2(w_i) \right] \log \left(\frac{1 - \Sigma_{i \in S_{100}(K)} P_2(w_i)}{1 - \Sigma_{i \in S_{100}(K)} P_{MM}(w_i)} \right)$$

- $S_{100}(K)$: count ≥ 100

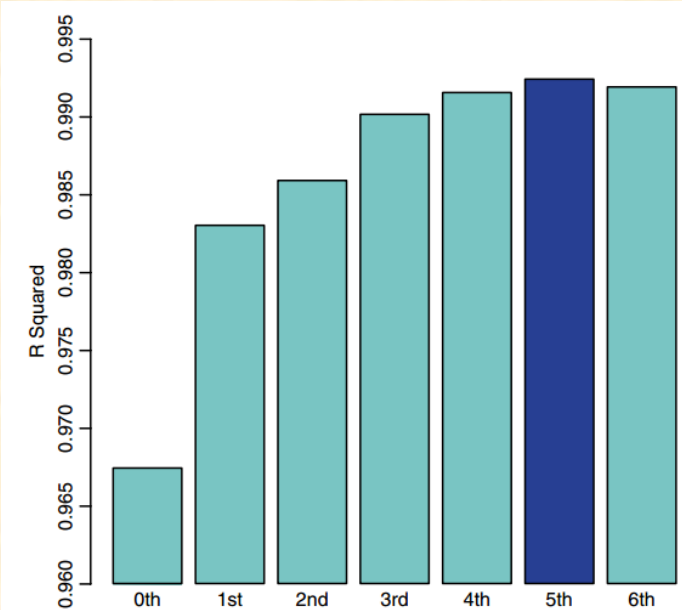


SELEX-seq

- Scoring k-mers:

$$score_i(w) = \sqrt[i]{freq_i(w)/freq_0(w)}$$

- Estimated frequency of cycle 0 by 5-th order Markov model.



Open challenges

- HT-SELEX
  - Choice of cycle.
  - Utilization of several cycles together.
  - Technological biases.

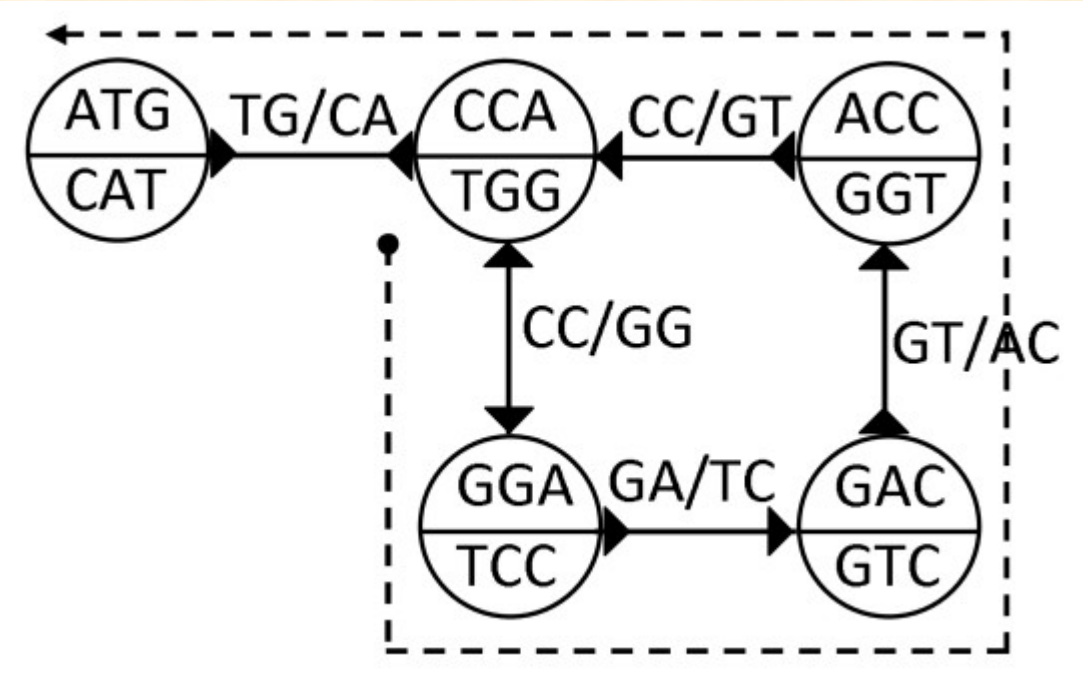
- *In vivo* binding prediction:
  - Improve prediction using *in vitro* models.

Summary

- PWM model for protein-DNA binding.
- MEME: find motifs in genomic sequences.
- RAP, BEEML-PBM: infer motifs from PBM data.
- New computational challenges with HT-SELEX.


THANKYOU
1 2 3 4 5 6 7 8

Bi-directed de Bruijn graphs for assembly and design



(Kundeti *et al.*, BMC Bioinformatics 2010)

De Bruijn sequence of order k

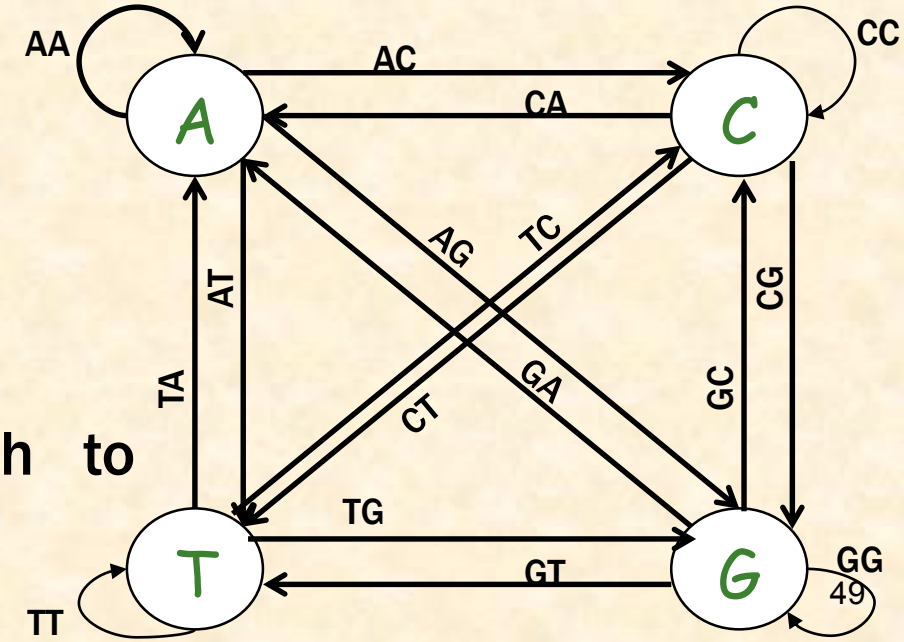
- A cyclic sequence s.t. every possible k-mer appears in it exactly once.
- The most compact sequence to cover all *k*-mers.
- Length: $|\sum|^k$
- DNA alphabet = {A, C, G, T}
- dB sequence of order k=2:
  ACGTAGAATTGGCCTC

De Bruijn graph of order k

- Vertex = k-mer $(x_1,\ldots,x_k)$ $\quad |V| = |\Sigma|^k$
- Directed edge for overlap of (k-1):

  $(x_1,x_2,\ldots,x_k) \rightarrow (x_2,\ldots,x_k,x_{k+1})$ $\quad |E| = |\Sigma|^{k+1}$
- $\forall v\ d^+(v) = d^-(v) = |\Sigma|$

  outdegree indegree



The graph is strongly connected and each vertex is balanced

➔ **Euler tour in a de Bruijn graph to generate de Bruijn sequence.**

Assembly by de Bruijn graph

- Construct de Bruijn graph according to k-mer reads.
- [Edge weights by read scores.]
- Find an Euler tour in the graph.
- Is it always possible?

Chinese Postman Problem

- Find the shortest path that traverses each edge at least once.

- Solved by minimum-cost flow:
  - $V^- = \{v \mid d^-(v) > d^+(v)\}$, $V^+ = \{\ldots\}$
  - $c(s,v) = d^-(v) - d^+(v)$, $c(v,t) = d^+(v) - d^-(v)$, $c(u,v) = \infty$
  - $a(u,v) = w(u,v)$, $a(s,v) = a(v,t) = 0$

- Add f(u,v) copies of edge (u,v).

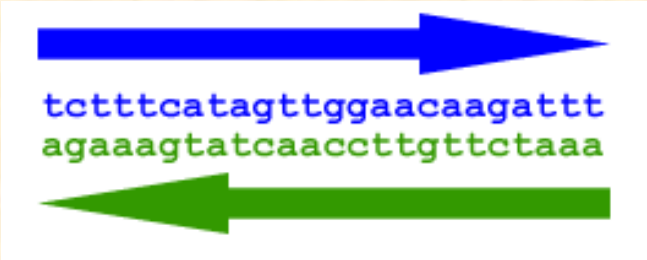
Running time of CPP

- Successive shortest path:
  - Continue till no augmenting path exists:
    - Send flow through the cheapest path.
    - Update residual network (negative costs).
  - #iterations $\leq$ maxF = $\frac{1}{2} \sum_v |d^+(v)-d^-(v)|$
  - Finding cheapest augmenting path:
    - Bellman-Ford: O(|V||E|)
    - Dijkstra (Johnson's): O(|V|log|V|)
    - Integral and bounded costs: O(|V|+|E|)

Room for improvement

- Reverse Complementary of double-stranded DNA:

  A ⇔ T C ⇔ G



➔ In a double stranded sequence, when a k-mer is present, so is its RC.

Goal: find the shortest sequence

to cover all the k-mers.

Outline

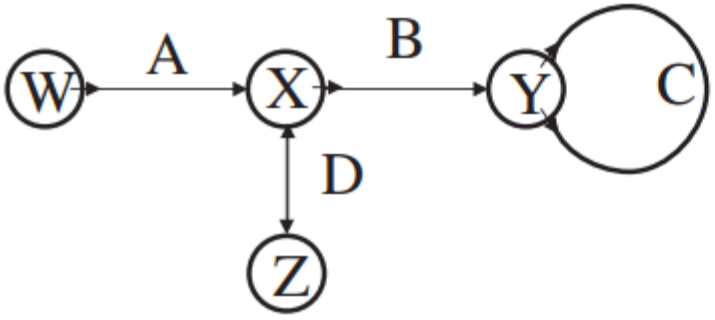
- Bi-directed de Bruijn graphs.
- Solution to assembly: LP and bipartite.
- Solution to design: RCdB.
- Summary and open problems.

Bi-directed graphs

- Each edge endpoint has an orientation.
- Incidence matrix: VxE -> {-2,-1,0,1,2}
- $d^-(v) = -\sum_{\{e \in E \mid I(v,e)<0\}} I(v,e)$, $d^+(v) = +\sum \ldots$
- $bal(v) = d^+(v) - d^-(v) = \sum_e I(v,e)$

A is negative-incident to X

B is positive-incident to X

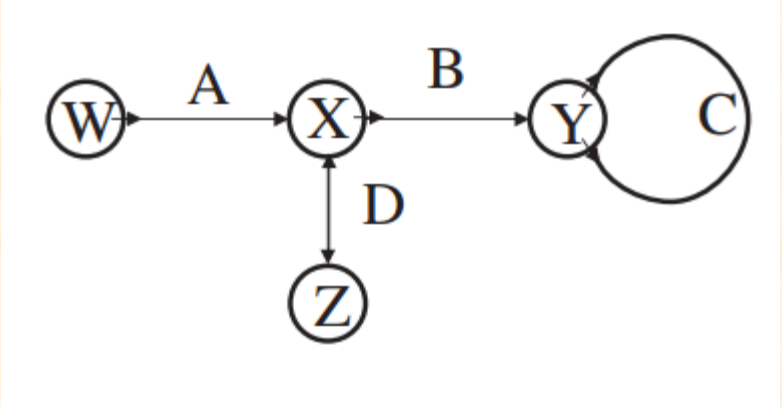


	A	B	C	D
W	1			
X	−1	1		−1
Y		−1	2	
Z				−1

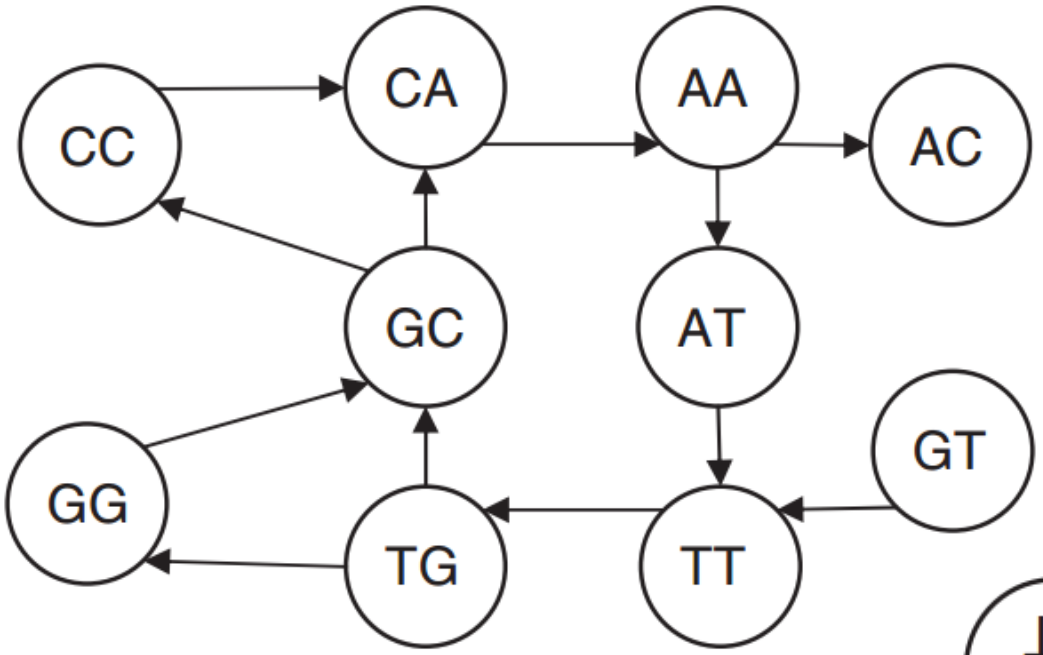
(Medvedev and Brudno, JCB 2009)

Walks in bi-directed graphs

- $v_1, e_1, \ldots, v_{k-1}, e_{k-1}, v_k$
  - e_i incident to x_i and x_{i+1}
  - e_{i-1} and e_i have opposite orientation at x_i
- W, A, X, B, Y, C, Y, B, X, D, Z.
- W, A, X, D, Z.

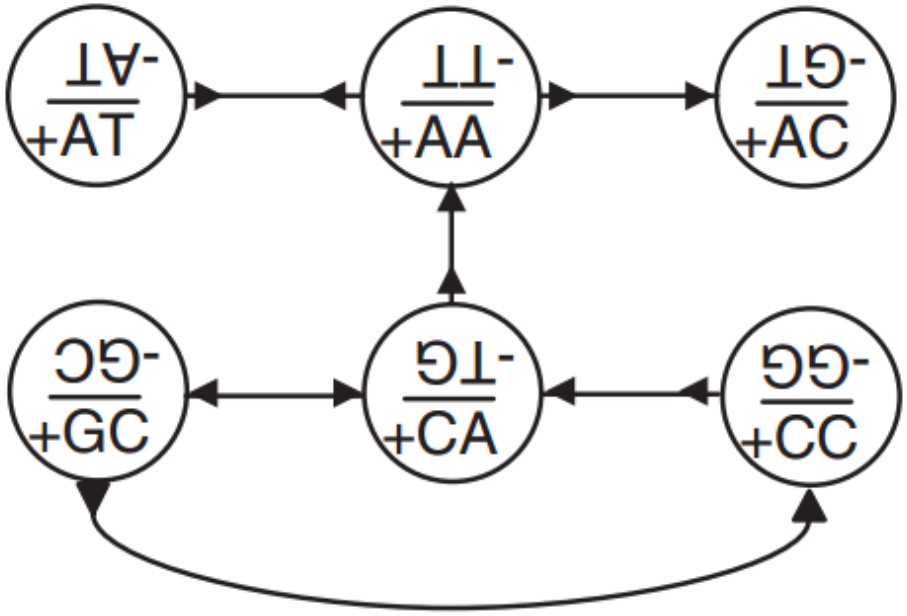


Bi-directed de Bruijn graphs



Bi-directed overlap:

- e pos to x, neg to y, p(x) overlaps p(y)
- e pos to x, pos to y, p(x) overlaps n(y)
- e neg to x, neg to y, n(x) overlaps p(y)
- e neg to x, pos to y, n(x) overlaps n(y)



Linear Programming to solve Chinese postman problem

Find f: E -> N (#copies to add of each edge)

$$\text{minimize} \sum w(e)f(e)$$

$$\text{subject to} \sum_{e} I^G(x,e)f(e) = -bal^G(x) \text{ for each vertex } x$$

$$f(e) \geq 0 \qquad \text{for each edge } e$$

Solving the LP

- LP solutions to binet matrices are half-integral (Appa and Kotnyek, 2006).
- A reduction to undirected graphs (totally unimodular matrices) has integral solutions (Hochbaum, 2004).
- The solution is a 2-approximation.

Shortest path in bidirected

- Modified Dijkstra:
  - updates neighbors consistent with a walk.
  - $O((|V|+|E|)\log|V|)$
- Shortest paths between unbalanced:
  - $V^- = \{v \mid d^-(v) > d^+(v)\}$, $V^+ = \{\ldots\}$
  - Run $P = \min\{|V^+|, |V^-|\}$ times.
- Now, how can we solve assembly problem?

(**Kundeti, Rahasekaran and Dinh, Discrete Math. Algorithm. Appl. 2012**)

Solution by bipartite matching

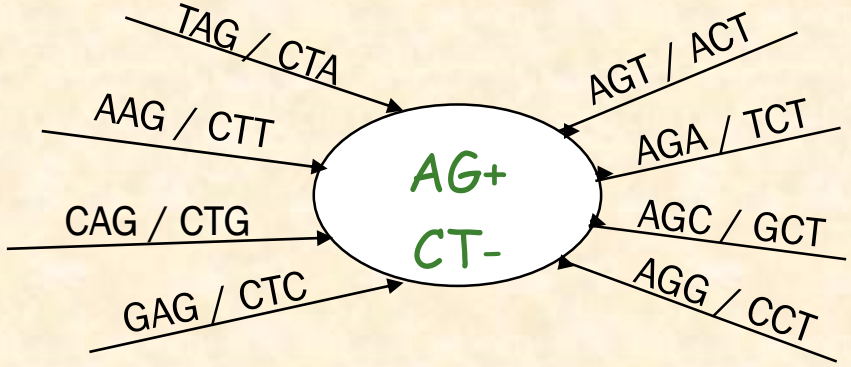
- Construct bipartite graph:
  - $U^- = U_{v \epsilon V} - \{v^{(1)},\ldots,v^{(d-(v) > d+(v))}\}$, $U^+ = \{\ldots\}$
  - $w(u^{(i)},v^{(j)})$ = shortest_path(u,v)
- For each edge in the matching:
  - Add edges of the shortest path.
- Find Euler tour in original graph.

Running time

- Modified Dijkstra:
  - $O((|V|+|E|)\log|V|)$
  - Run $P = \min\{|V^+|, |V^-|\}$ times.
- Bipartite matching:
  - Hungarian method
  - $O(|V|^3)=O((\Sigma_v|d^+(v)-d^-(v)|)^3)$

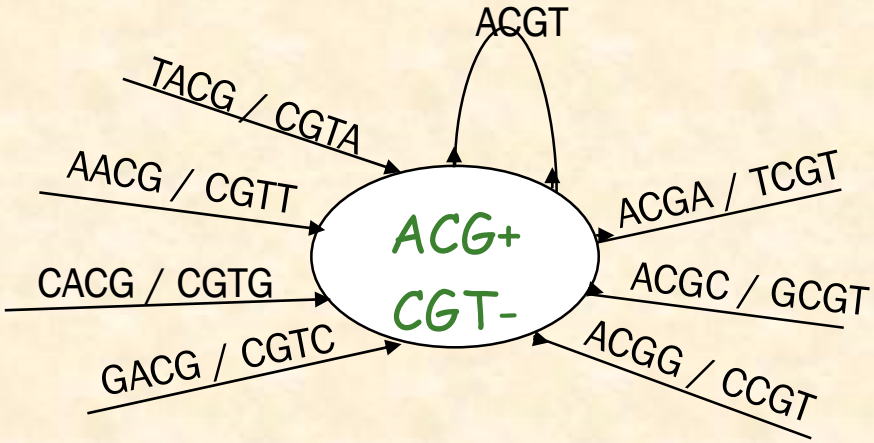
Reverse complement de Bruijn sequence

- For odd k, the graph is balanced.



- For even k, the graph is unbalanced.

e pos to x, pos to y,
p(x) overlaps n(y)



(**Orenstein and Shamir, Bioinformatics 2013**)

Solutions

- Can be solved by linear programming.
- Or by maximum weighted matching:
  - Shortest paths = label overlaps.
  - #vertices = $4^{k/2}$
  - Run time = $O(4^{3k/2})$
- Integer weights => improved run-time.

Computational results

Reverse complement de Bruijn sequence

K	2	4	6	8	10	12	14
De Bruijn	16	256	4,096	65,536	1,048,576	16,777,216	268,435,456
Optimal	10	142	2,140	33,262	526,816	8,400,772	134,274,844
Lower bound	10	136	2,080	32,896	524,800	8,390,656	134,225,920
Saving factor	1.6	1.8	1.91	1.97	1.99	1.997	1.999

Open questions

1. A sequence with improved coverage of gapped k-mers.
2. What is the number of sequences?

$$\frac{(k!)^{k^{n-1}}}{k^n}$$

3. Probe design that minimizes the number of probes.
4. Linear time algorithm.
5. Closed formula for length.

References

1. Maximum likelihood of genome assembly.
   (Medvedev and Brudno, *Journal of Computational Biology* 2009)

2. An efficient algorithm for Chinese postman walk on bi-directed de Bruijn graphs.
   (Kundeti, Rajasekaran and Dinh, *Discrete Math. Algorithm. Appl.* 2012)

3. Design of shortest double-stranded DNA sequence covering all k-mers.
   (Orenstein and Shamir, *Bioinformatics* 2013)