Finding Motifs Using Random Projections

by J. Buhler and M. Tompa

A Presentation by

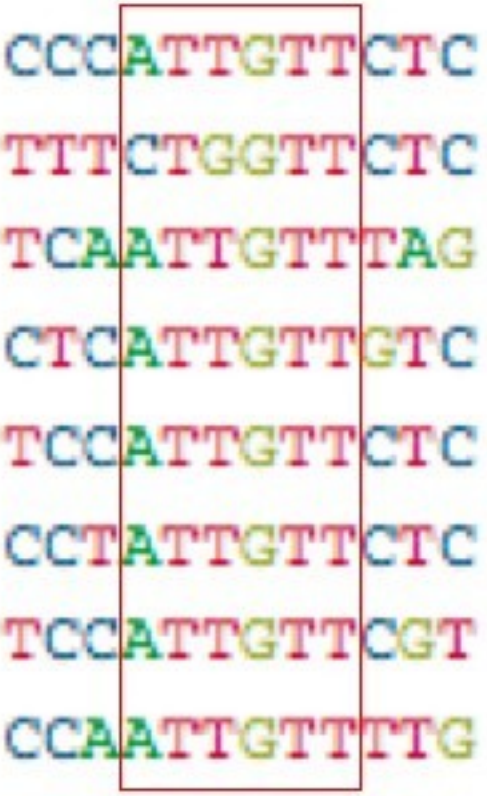
Guénola Drillon
Anisah Ghoorah
Lin Han
Frank Dondelinger

Overview

- DNA motifs
- The problem
- Current approaches
- New algorithm - PROJECTION
- PROJECTION's results
- Conclusions

DNA Motifs

- DNA
  - 4 nucleotides: A, T, C and G
- DNA Motifs
  - Short, recurring patterns in DNA
  - Strongly conserved
  - Have biological function
    - Gene regulation, gene interaction
  - Indicate sequence-specific binding sites for proteins



D'haeselleer (2006)

Motif Representations

- Consensus pattern
  - Using IUPAC code
- Frequency matrix
- Logo

Nucleic acid codes

code	description
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
U	Uracil
R	Purine (A or G)
Y	Pyrimidine (C, T, or U)
M	C or A
K	T, U, or G
W	T, U, or A
S	C or G
B	C, T, U, or G (not A)
D	A, T, U, or G (not C)
H	A, T, U, or C (not G)
V	A, C, or G (not T, not U)
N	Any base (A, C, G, T, or U)

D'haeseleer (2006)

Motif Finding Problem

GAATTCATACCAGATCACCGGATTCCCGACTCCAAATGTGTCCCCCTCACAC

TCCCCCGAAAACCGACTTCTGCTCTTAGACCACTCTACCCTATTCCCCACACT

CACCGGAGCCAAAGCCGCGGCCCTTCCGTTCCGATTACCGAAAAGACCCCA

CCCGTAGGTGGCAAGCTAGCTTAAGTAACGCCACTTCGATTAACGAGGAAA

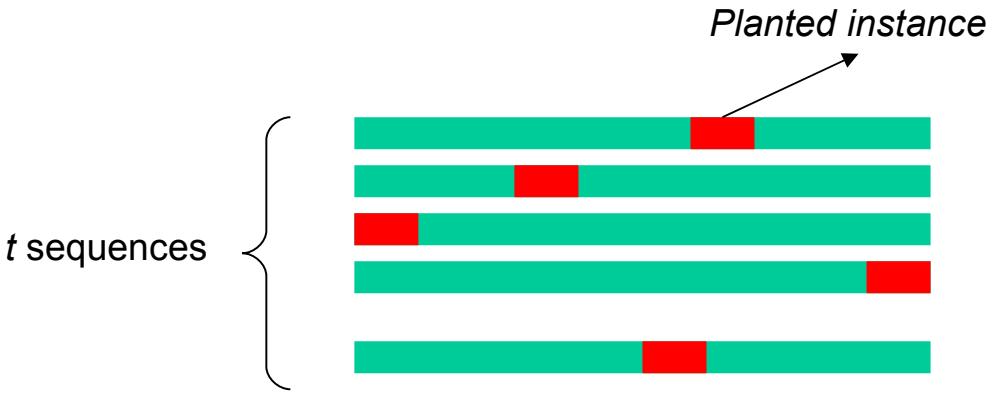
AATACATAACTGACCTATTATCGAGTTCAGATCAAGGTCAGGAACAAAGAA

ACACCGATTACCGTAACCGTAAGATARTGGTATCGATACGTAGACAGTTTA

- Planted (*l*,*d*)-Motif
  - Planted (11,2)-Motif: `CCGATTACCGA`
- *l*-mers
  - All possible subsequences of length *l* in each sequence

Problem Definition

- Given *t* sequences, each of length *n*, find a motif *M* of length *l*, where each planted instance differs from *M* in *d* positions
  - Planted (*l*,*d*)-Motif
- No prior knowledge of motif M



Why Motif Finding?

- Comparative genomics
  - Study similar genes in different species using microarrays
  - Identification of transcription factor binding sites
  - Genetic regulatory network
- Genomes are large and complex
- Simple search won't work!
- Need more efficient search algorithm

Current approaches (1) - Local Search

- Gibbs Sampling - Lawrence et al (1993)
  - Obtain an initial motif model
  - Use an iterative approach based on probability to find correct motif
- MEME - Bailey & Elkan (1995)
  - Obtain an initial motif
  - Use EM approach to find correct motif
- CONSENSUS - Hertz & Stormo (1999)
  - Obtain an initial motif
  - Use an iterative approach to build up motifs by adding more and more pattern instances.

Problem with Local Search

- Depends on initial conditions
- Local optima issues
  - Returns best solution in neighbourhood
  - Not necessarily the best planted motif

Current approaches (2)

- Enumeration
  - Exhaustive enumeration of all possible motifs M
  - Cover the entire search space
  - No risk of getting stuck in local optimum
- Problem
  - Too rigid for most real-world binding sites
  - Run in time exponential to motif length

Current approaches (3)

- WINNOWER – Pevzner & Sze (2000)
  - Graph-theoretic approach which represents a motif as a large clique
- SP-STAR – Pevzner & Sze (2000)
  - Heuristic local improvement technique using a scoring function
- Both solve the planted (15,4)-motif problem
- Problem
  - Fail to find the planted (14,4), (16,5), (18,6) motif problems

New Approach

PROJECTION Algorithm

- Random Projections (global search)
- Motif Refinement (local search)

Random Projection

Hash h(x)

- Choose *k* of the *l* positions at random
- Consider x as an *l*-mer, then h(x) is the *k*-mer resulting from selecting *k* residues of x
- A projection from *l*-dimensional space onto a *k*-dimensional subspace
- Example:

l = 15 Projection *k* = 7

Projection = (2, 4, 5, 7, 11, 12, 13)

Random Projection

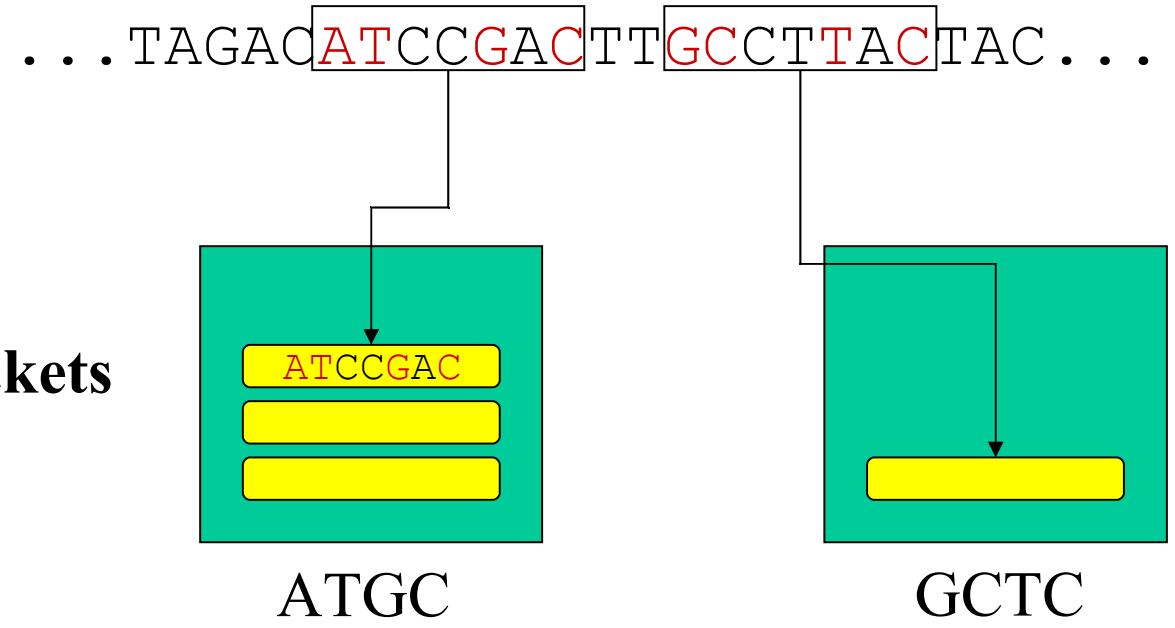
- 4^k buckets in total
- M: motif; h(M): the planted bucket
- If $k<l-d$, a number of planted motifs in planted bucket
- If k not too small, less than one l-mer in random bucket
- Highly enriched l-mers in planted bucket enable recovering the motif.

Random Projection

- *s* is the threshold for potential planted bucket
- Choose buckets that contain at least *s* *l*-mers

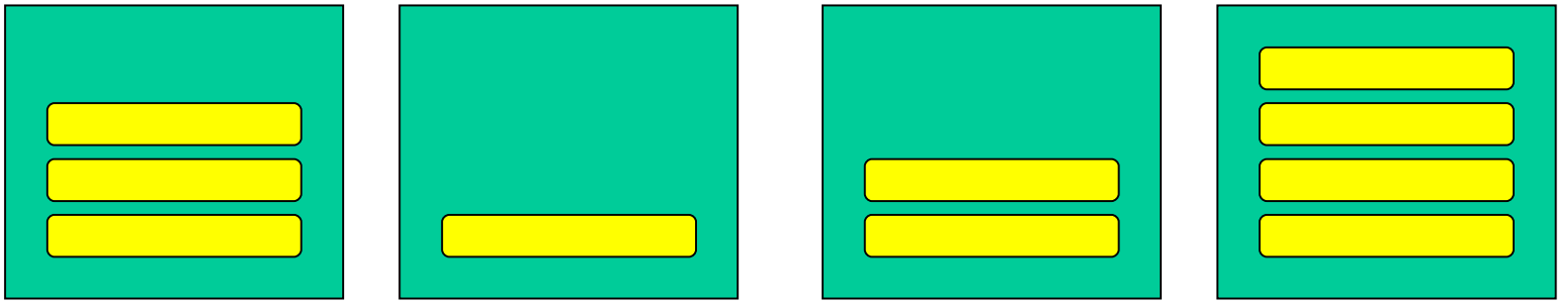
Example of Hashing and Buckets

$l = 7$, $k = 4$ with projection position (1,2,5,7)



Example of Hashing and Buckets

- s=3
- Choose buckets which contain more than *3 l*-mers



Three Important Parameters

Projection size k

- $k<l-d$ and k not too small to keep planted bucket highly enriched
- Larger k to ensure we have less than one l-mer in each bucket

Three Important Parameters

Bucket threshold *s*

- Varies according to the data we use.
- Case of (20, 2) and (16, 5)
- Larger number of sequences

Three Important Parameters

The number *m* of independent trials to run:

$$m = \frac{\log(1-Q)}{\log(B)}$$

- Q: probability that *s* or more motif instances in planted bucket in at least one of *m* trials
- B: probability that fewer than *s* planted instances in planted bucket in a number of independent Bernoulli trials

Motif Refinement

We have our buckets:

Now what?

For each large enough bucket h:

- Use h as a starting point W_h
- Apply EM to refine W_h to W_h^*
- Get consensus motif C using model W_h^*

At the end, return best C found

Starting Point W_h

W_h is a model for the motif

4 x *l* matrix

→ $W_h(i, j)$ = probability of base i in position j

Approximation that works in practice

Starting Point W_h: Example

In bucket h:

AGT
AAA
AGC

$$W_h = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 1/3 & 1/3 \\ 0 & 0 & 1/3 \\ 0 & 2/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix} \end{array}$$

Positions

Bases

To avoid too many zeroes, add background probability b_i using Laplace smoothing

Refinement: Finding W_h^*

Use EM to refine initial model W_h

Let *S* be the dataset, *P* the background distribution

Find W_h^* that (locally) maximises:

$$\frac{Pr(S|W_h^*, P)}{Pr(S|P)}$$

Could take a long time!

Better: Run only a few iterations of EM

Refinement: Find the Motif

From W_h^*, want to determine motif:

For each input sequence:
Determine likeliest *l*-mer w.r.t. W_h^*

Likelihood of *l*-mer x determined by:

$$\frac{Pr(x \mid W_h^*)}{Pr(x \mid P)}$$

Get set T of *t* most likely *l*-mers

Refinement: Find the Motif

Example: Most likely 2-mer in AGT

(Assume *P* same for all bases in all positions)

$$W = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 0.88 & 0.20 \\ 0.01 & 0.30 \\ 0.10 & 0.01 \\ 0.01 & 0.49 \end{pmatrix} \end{array}$$

Two 2-mers: AG and GT

`Likelihood of AG: 0.88x0.01 = 0.0088`

`Likelihood of GT: 0.10x0.49 = 0.0490`

Add GT to set T

Refinement: Find the Motif

Once set *T* complete: Find consensus C_h

Then calculate s(T): number of *l*-mers in T that are further than *d* away from C_h

Return the consensus with the smallest value s(T) over all buckets and all runs

Ideally, find C_h such that s(T) = 0

Refinement: Find the Motif

Example: $l = 3$, $d = 1$, $T = \{AGT, AAA, AGC\}$

Consensus: AG? -> Many schemes possible

Let's say consensus AGT

`dist(AGT, AGT) = 0`

`dist(AAA, AGT) = 2` **`2 > d`**

`dist(AGC, AGT) = 1`

s(T) = 1

Refinement: A Heuristic

For the simulated data, we can do better than minimising s(*T*) over all buckets and all runs.

sc(T) = number of *l*-mers in T that are **at most** *d* away from C_h

Let T' contain the *l*-mers that are closest to C_h

If sc(T') > sc(T), replace T with T' and repeat

Usually converges quickly. If final score sc(T) = *t*, return the motif, otherwise maximise score over all buckets and all runs

Refinement: A Heuristic

Example: $l = 3$, $d = 1$, $T = \{AGT, AAA, AGC\}$,

$C_h = AGT$

$$sc(T) = 2$$

If: $S = \{AGTC, AAAT, AGCT\}$

then: $T' = \{AGT, \mathbf{AAT}, AGC\}$

$sc(T') = 3$

$\rightarrow$ return consensus of T' (which happens to be C_h)

PROJECTION Algorithm Recap

PROJECTION algorithm:

- Do random projections
  - Hash *l*-mers to buckets using *k* random positions
  - Use full buckets as starting points
- Do motif refinement
  - Get model W_h from bucket h
  - Refine to optimal model W_h^* (using e.g. EM)
  - Return best consensus motif

Experimental Results

- Experiments on Simulated Data
- Limitations on Solvable (l,d)-Motif Problems
- Transcription Factor Binding Sites
- Ribosomes Binding Sites

Experiments on Simulated Data

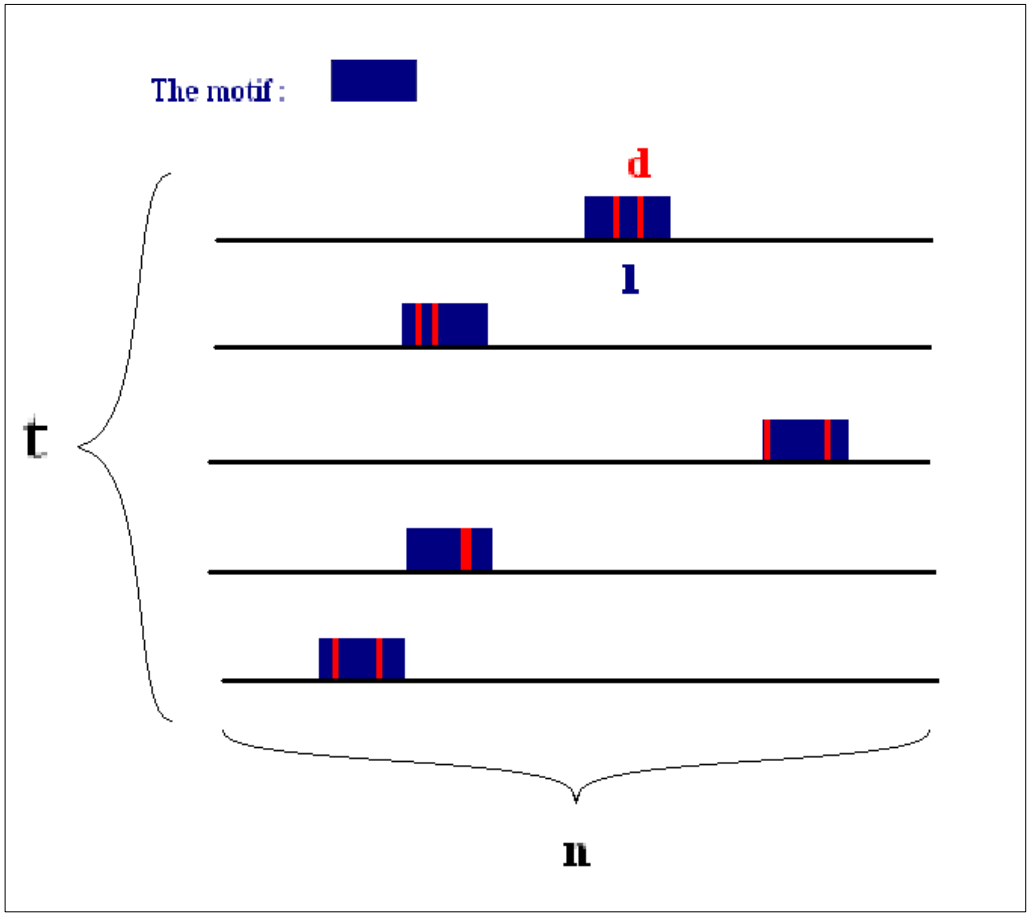
Simulated Data:

1 – a motif M is chosen randomly

2 – *t* independent planted instances are produced by randomly selecting d positions in M

3 – their position in the input sequence is selected randomly

4 – *n-l* residues of each sequence are chosen randomly



Experiments on Simulated Data

Performance coefficient :

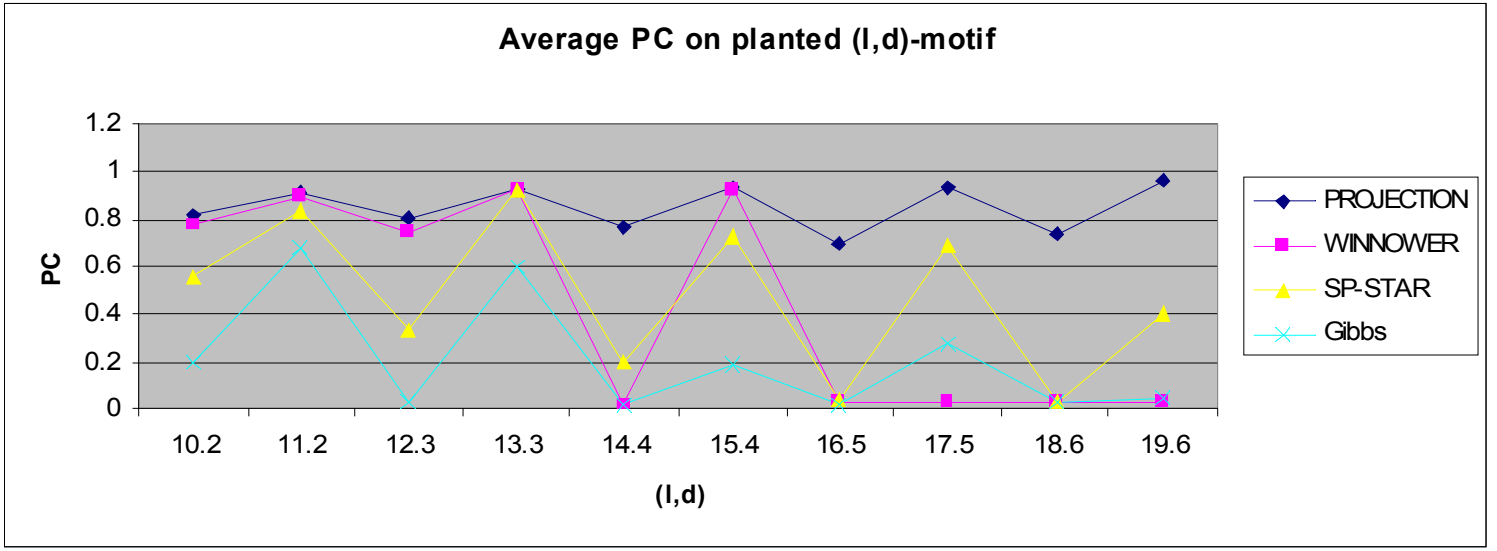
GGACCTCAATGCAGGATACACCGATCGGTA
GGAGTACGGCAAGTCCCCATGTGAGGACCT
AGGCTGGACCAGGACCTGACTCTACACCTA
TGGACCTGCAGGATACAGCGGGACCTATCG

.....

K = the t*l residue positions in the t planted motif instances
P = corresponding set of residues in the instances predicted by the algorithm

$$PC = \frac{|K \cap P|}{|K \cup P|}$$

Experiments on Simulated Data



Results :

- Average on 20 random instances
- All runs used projection size k = 7

  and bucket threshold s = 4

  but a different number of iterations m
- WINNOWER (k = 2)

Limitations on Solvable (l,d)-Motif Problems

Why is it difficult to find a (l,d)-motif ?

What is the difference between :

`(9,2)` `(11,3)` `(13,4)` `(15,5)` `(17,6)` and

`(10,2)` `(12,3)` `(14,4)` `(16,5)` `(18,6)` ?

between :

(l,d) and (l+1,d) ?

Limitations on Solvable (l,d)-Motif Problems

- The probability that a random sequence will correspond to a motif M with up to *d* substitutions

$$p_d = \sum_{i=0}^{d} \binom{l}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{l-i}$$

- The probability to find a random motif

$$E(l,d) = 4^l \left(1 - \left(1 - p_d\right)^{n-l+1}\right)^t$$

Limitations on Solvable (l,d)-Motif Problems

Statistics of spurious (l,d)-motifs in simulated data :

l	d	$E(l,d)$	$E(l+1,d)$	apc	Correct	Spurious	19/20	m
9	2	1.6	6.1×10^{-8}	0.28	11	5	4	1483
11	3	4.7	3.2×10^{-7}	0.026	1	13	6	2443
13	4	5.2	4.2×10^{-7}	0.062	2	15	3	4178
15	5	2.8	2.3×10^{-7}	0.018	0	7	13	6495
17	6	0.88	7.1×10^{-8}	0.022	0	8	12	9272

Transcription Factor Binding Sites

Context:

- Biological data
- 4 type of genes and a collection of promoter regions
- Known to contain binding sites for transcription factors

Differences:

- Motifs are better conserved
- Less 'subtle' : same *d* positions

Transcription Factor Binding Sites

Sequence	Sample Size	t	Best (20,2) Motif	Reference Motif
preproinsulin	7689	4	GGAAATTGCAGCCTCAGCCC	CCTCAGCCC
DHFR	800	4	CTGCAATTTCGCGCCAAACT	ATTTCNNGCCA
metallothionein	6823	4	CCCTCTGCGCCCGGACCGGT	TGCRCYCGG
c-fos	3695	5	CCATATTAGGACATCTGCGT	CCATATTAGAGACTCT
yeast ECB	5000	5	GTATTTCCCGTTTAGGAAAA	TTTCCCNNTNAGGAAA

- $l = 20$ and $d = 2$
- $k = 7$ and $s = 3$
- Reference motif from databases or experiments

Transcription Factor Binding Sites

Result Analysis

- Noteworthy results: a fairly primitive refinement and 30% to 80% fewer starting point are refined
- Two or more distinct motifs with the same score: the most 5'-shifted
- Only a single high-scoring motif is return (e.g. Preproinsulin)
- A less stringent selection criteria increases the number of identifications (with (14,2): 20 correct sites on 39)
- Need of additional refinement or filtering with biological data

Ribosome Binding Sites

Context:

- Identification of a short site: $l = 6$ ($k = 4$)
- Short DNA sequence: $n = 20$
- Thousands of input sequences: t is big
- The motif occurs in only a fraction of them

Ribosome Binding Sites

Organism	t	s	m	Motif	Occurs	16S rRNA	Best z-score
M. jannaschii	1679	196	14	AGGTGA	606	GGAGGTGATCC	GGTGA
H. influenzae	1716	202	17	AGGAAA	639	TAAGGAGGTGA	AAGGA
T. maritima	1846	216	13	GGAGGT	1198	GAAAGGAGGTG	AGGTG
B. subtilis	4099	480	35	AGGAGG	2742	TAGAAAGGAGG	AGGAG
E. coli	4287	502	35	AAGGAG	1306	TAAGGAGGTGA	AGGAG

Some proof :

- The good fit with the 3' end of the 16S rRNA sequences
- AAGGAGG or a large substring
- Similar binding sites from different algorithms (e.g Z-score)

Comments :

- No need to pick *d* positions randomly
- No need to use the projection algorithm at all (enumerative ones are good enough)

Experimental Results

- For simulated data PROJECTION is quite good
- But still insolvable (l,d)-motif problems
- Biological data less ‘subtle’
- Other algorithms can do ‘as well’

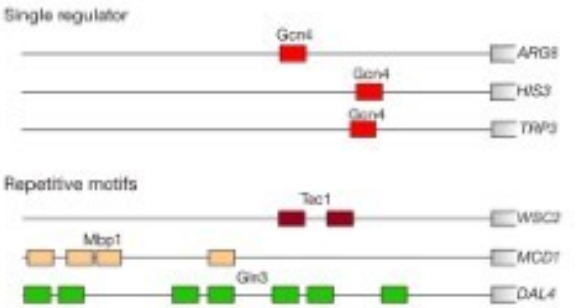
→ So improvement in time, in space and maybe for future more complex biological data

Authors' Conclusions

- Objective
  - To find DNA motifs using Random Projections
- Achievement
  - More efficient than WINNOWER and SP-STAR
- Future work - Consider 'real' motif-finding problem
  - Predicting length of motif
  - Finding multiple motifs
  - Motif instances with insertions and deletions
  - Better biological examples for illustration

Our Conclusions (1)

- Best results for planted (l,d) motif problem
- Assumptions
  - Sequences has same length
  - One motif for each sequence
  - Motif has fixed length
- Not suitable for real biological problems



Harbison et al (2004)

Our Conclusions (2)

- Example of existing program
  - BioProspector - Liu et al (2001)
  - Sequences have varying length
  - >=1 motif per sequence
  - Motifs have varying length
- Most algorithms cover a small subset of known binding sites, with little overlap
- Try combine results from multiple motif finding algorithms!

References

- D'haeseleer P. What are DNA sequence motifs? Nature Biotechnology (2006) Vol. 24 No. 4 Pp. 423-425
- D'haeseleer P. How does DNA sequence motif discovery work? Nature Biotechnology (2006) Vol. 24 No. 8 Pp. 959-961
- Liu X, Brutlag D L, Liu J S. BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Pacific Symposium on Biocomputing (2001) 6:127-138
- Harbison et al. Transcriptional regulatory code of a eukaryotic genome. Nature (2004) Vol. 431 Pp. 100-104

Any Questions?