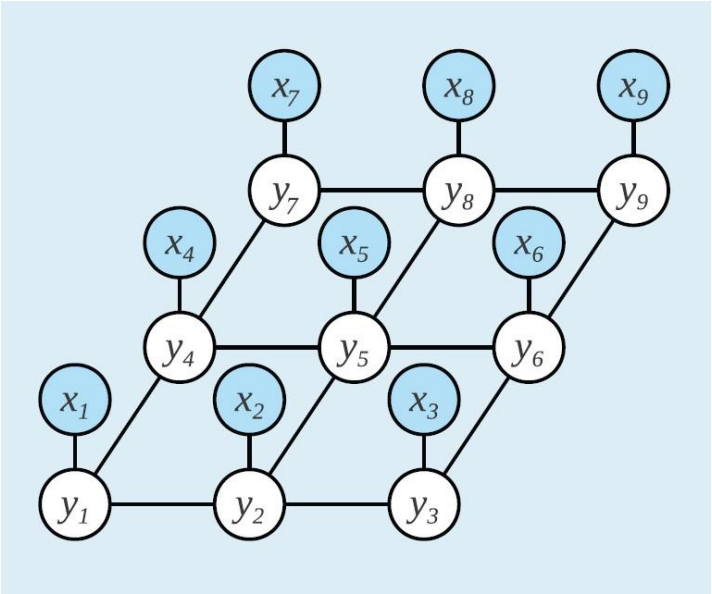
Probabilistic Graphical Models in Bioinformatics

Lecture 12: Variable elimination; Gibbs sampling for motif finding



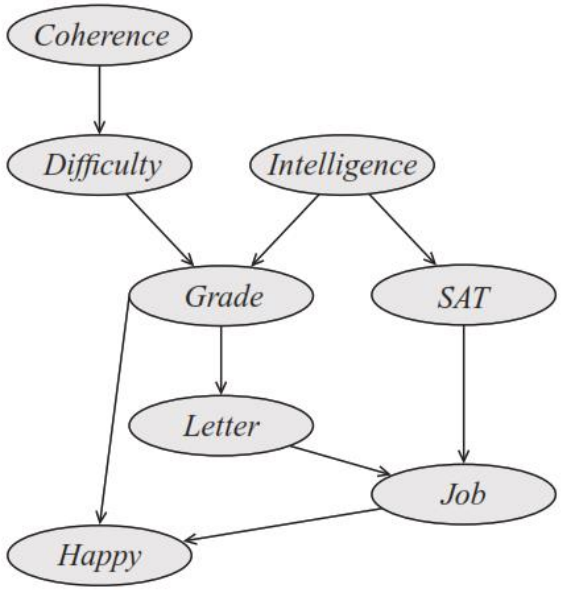
Hesam Montazeri
Department of Bioinformatics, IBB, University of Tehran

Khordad 1, 1398
(May 22, 2019)

Variable elimination

Two runs of variable eliminations for the query $P(J)$

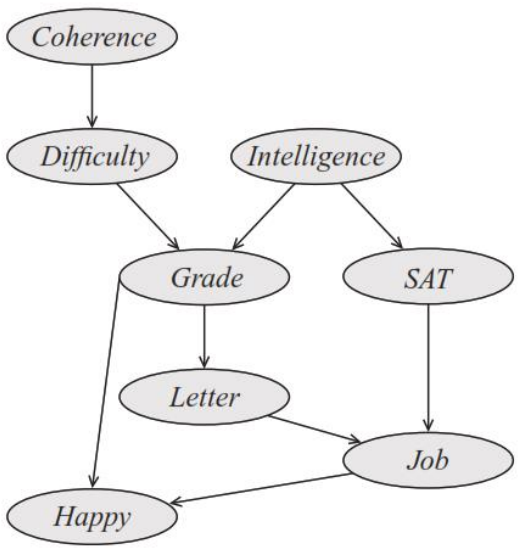
Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1(D)$
2	D	$\phi_G(G,I,D)$, $\tau_1(D)$	G, I, D	$\tau_2(G,I)$
3	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_2(G,I)$	G, S, I	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	H, G, J	$\tau_4(G,J)$
5	G	$\tau_4(G,J)$, $\tau_3(G,S)$, $\phi_L(L,G)$	G, J, L, S	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	J, L	$\tau_7(J)$



Step	Variable eliminated	Factors used	Variables involved	New factor
1	G	$\phi_G(G,I,D)$, $\phi_L(L,G)$, $\phi_H(H,G,J)$	G, I, D, L, J, H	$\tau_1(I,D,L,J,H)$
2	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_1(I,D,L,S,J,H)$	S, I, D, L, J, H	$\tau_2(D,L,S,J,H)$
3	S	$\phi_J(J,L,S)$, $\tau_2(D,L,S,J,H)$	D, L, S, J, H	$\tau_3(D,L,J,H)$
4	L	$\tau_3(D,L,J,H)$	D, L, J, H	$\tau_4(D,J,H)$
5	H	$\tau_4(D,J,H)$	D, J, H	$\tau_5(D,J)$
6	C	$\phi_C(C)$, $\phi_D(D,C)$	D, J, C	$\tau_6(D)$
7	D	$\tau_5(D,J)$, $\tau_6(D)$	D, J	$\tau_7(J)$

This elimination ordering introduces factors with much larger scope.

How to deal with evidence?



- **Goal:** to compute $P(J \mid i^1, h^0)$. We can use

$$P(J \mid i^1, h^0) = \frac{P(J, i^1, h^0)}{P(i^1, h^0)}$$

- we first compute unnormalized distribution $P(J, i^1, h^0)$. By renormalization (to the probability of the evidence) we obtain the conditional probability.

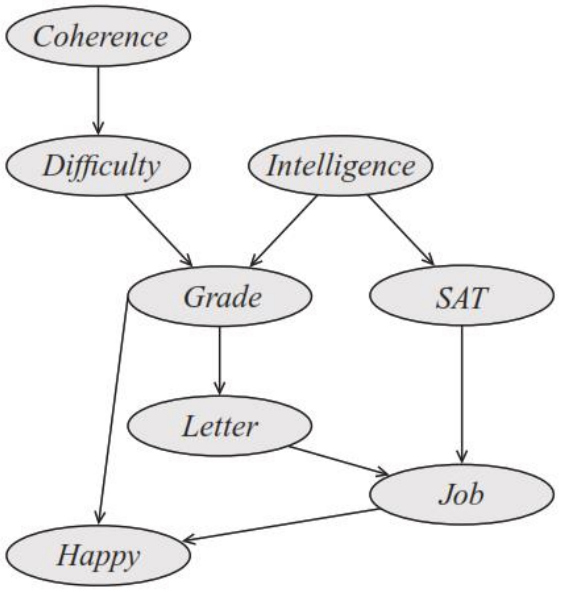
Algorithm 9.2 Using Sum-Product-VE **for computing conditional probabilities**

Procedure Cond-Prob-VE (
  $\mathcal{K}$, // A network over $\mathcal{X}$
  $\boldsymbol{Y}$, // Set of query variables
  $\boldsymbol{E} = \boldsymbol{e}$ // Evidence
)

$\Phi \leftarrow$ Factors parameterizing $\mathcal{K}$
Replace each $\phi \in \Phi$ by $\phi[\boldsymbol{E} = \boldsymbol{e}]$
Select an elimination ordering $\prec$
$\boldsymbol{Z} \leftarrow = \mathcal{X} - \boldsymbol{Y} - \boldsymbol{E}$
$\phi^* \leftarrow$ Sum-Product-VE$(\Phi, \prec, \boldsymbol{Z})$
$\alpha \leftarrow \sum_{\boldsymbol{y} \in Val(\boldsymbol{Y})} \phi^*(\boldsymbol{y})$
return α, ϕ^*

Computing $P(J, i^1, h^0)$:

Step	Variable eliminated	Factors used	Variables involved	New factor
1'	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1'(D)$
2'	D	$\phi_G[I=i^1](G,D)$, $\phi_I[I=i^1]()$, $\tau_1'(D)$	G, D	$\tau_2'(G)$
5'	G	$\tau_2'(G)$, $\phi_L(L,G)$, $\phi_H[H=h^0](G,J)$	G, L, J	$\tau_5'(L,J)$
6'	S	$\phi_S[I=i^1](S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6'(J,L)$
7'	L	$\tau_6'(J,L)$, $\tau_5'(J,L)$	J, L	$\tau_7'(J)$



Compare with computing $P(J)$:

Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1(D)$
2	D	$\phi_G(G,I,D)$, $\tau_1(D)$	G, I, D	$\tau_2(G,I)$
3	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_2(G,I)$	G, S, I	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	H, G, J	$\tau_4(G,J)$
5	G	$\tau_4(G,J)$, $\tau_3(G,S)$, $\phi_L(L,G)$	G, J, L, S	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	J, L	$\tau_7(J)$

- Steps 3 & 4 disappear since I and H do not need to be eliminated
- By not eliminating I, we avoid the step that correlates G and S.
- $\phi_I(I=i^1)() = P(i^1)$ is simply a number and can be multiplied to any factor.

Graph-theoretic analysis

- The VE algorithm does not care whether the graph that generated factors is directed, undirected or partly directed.

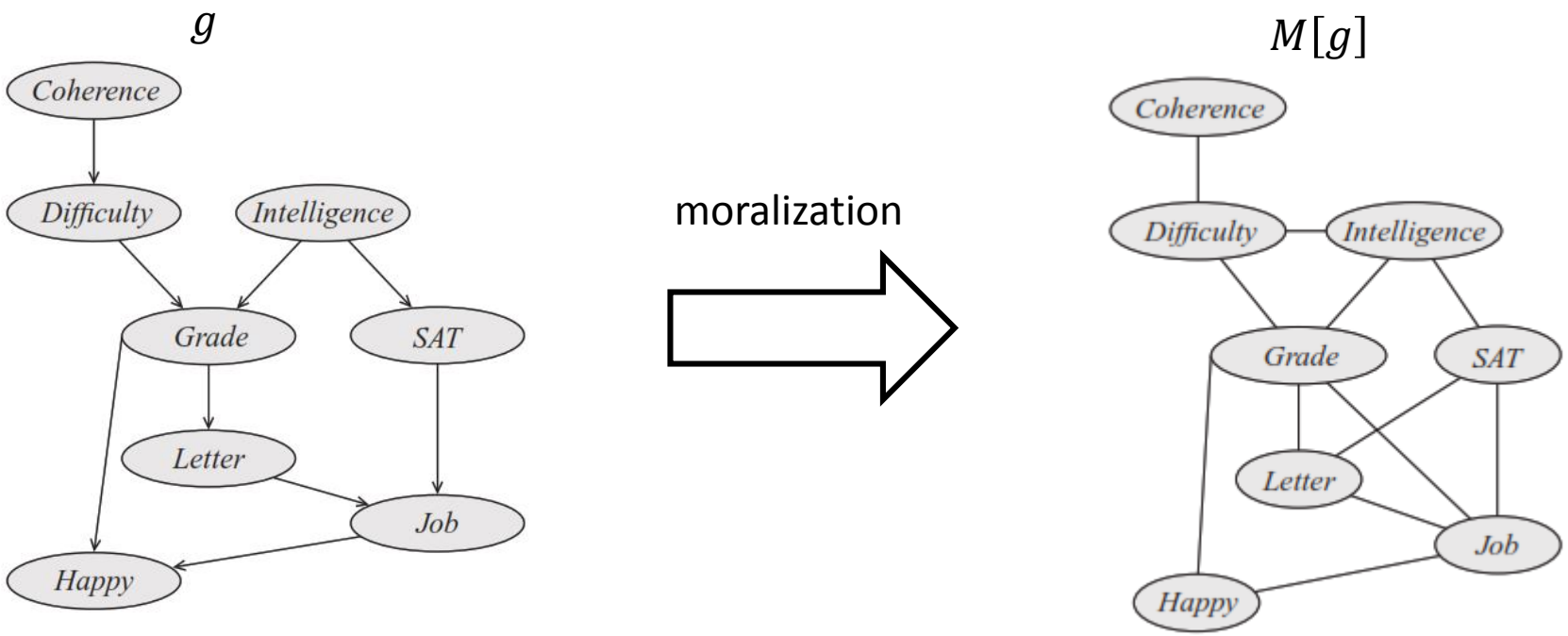
- The algorithm's input is a set of factors Φ, and the only relevant aspect of the computation is the scope of the factors.

- Hence, it can be viewed the algorithm is operating on an undirected graph.

- For a Bayesian network g, in the case without evidence, the undirected graph is precisely the moralized graph of g.

From Bayesian networks to Markov networks

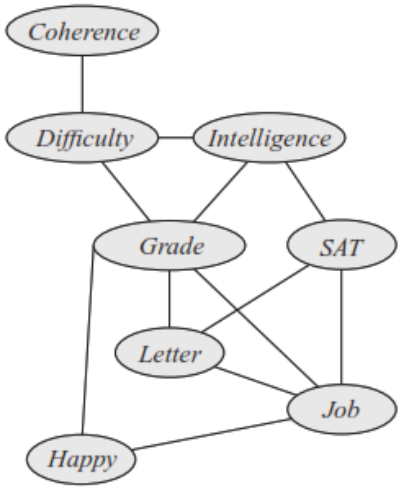
- The moral graph $M[g]$ of a Bayesian network structure g is the undirected graph that contains an edge between X and Y if
  a) There is a directed edge between them (in either direction)
  b) X and Y are both parents of the same node (the name morality originated from marrying the parents of a node).



Proposition: the moralized graph $M[g]$ is a minimal I-map for Bayesian network g.

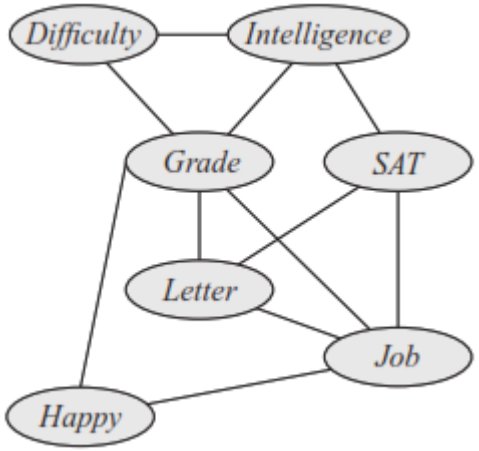
Elimination as graph transformation

Original graph

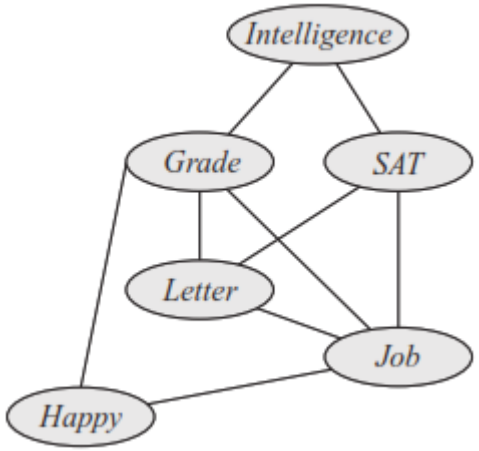


Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1(D)$
2	D	$\phi_G(G,I,D)$, $\tau_1(D)$	G, I, D	$\tau_2(G,I)$
3	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_2(G,I)$	G, S, I	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	H, G, J	$\tau_4(G,J)$
5	G	$\tau_4(G,J)$, $\tau_3(G,S)$, $\phi_L(L,G)$	G, J, L, S	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	J, L	$\tau_7(J)$

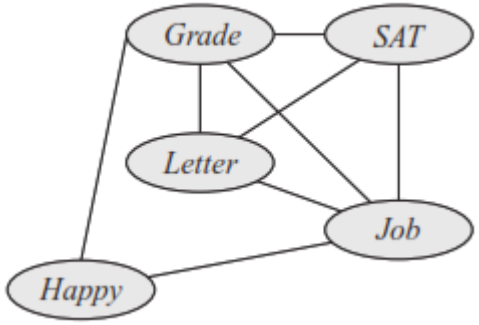
Eliminating C



Eliminating D



Eliminating I



Finding elimination orderings

- Finding an optimal eliminating orderings is *NP-complete.*
- Alternative approach: greedy search using heuristic cost function
- Possible cost functions:
  - Min-neighbors, min-fill, min-weight, etc.

Algorithm 9.4 Greedy search for constructing an elimination ordering

```
Procedure Greedy-Ordering (
    H    // An undirected graph over X ,
    s    // An evaluation metric
)
  Initialize all nodes in X as unmarked
  for k = 1 ... |X|
    Select an unmarked variable X ∈ X that minimizes s(H, X)
    π(X) ← k
    Introduce edges in H between all neighbors of X
    Mark X
  return π
```

Project 3:
Gibbs Sampling for motif finding

From Compeau, P. and Pevzner, P., 2015. Bioinformatics algorithms: an active learning approach, chapter 2.

How to score motifs?

- **Idea:** the goal is to select k-mers resulting in the most *conserved* motif matrix.
- We indicate the most frequent nucleotide in each column of the motif matrix by upper case letters.
- One simple definition for $Score(Motifs)$ is the number of lower case letters in the motif matrix *Motifs.*

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

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

Scoring motifs

COUNT(*Motifs*)	A:	2	2	0	0	0	0	9	1	1	1	3	0
	C:	1	6	0	0	0	0	0	4	1	2	4	6
	G:	0	0	10	10	9	9	1	0	0	0	0	0
	T:	7	2	0	0	1	1	0	5	8	7	3	4
PROFILE(*Motifs*)	A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
	C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
	G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
	T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4
CONSENSUS(*Motifs*)		T	C	G	G	G	G	A	T	T	T	C	C

An improved method of scoring motif matrices: the sum of the entropies of its columns.

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

For example, the entropy of the second column

$$-(0.2\log_2 0.2 + 0.6\log_2 0.6 + 0.0\log_2 0.0 + 0.2\log_2 0.2) \approx 1.371$$

Laplace's rule of succession

- Consider the following *Profile:*

$$Profile \quad \begin{array}{lcccccccccccc} \mathbf{A}: & .2 & .2 & .0 & .0 & .0 & .0 & .9 & .1 & .1 & .1 & .3 & .0 \\ \mathbf{C}: & .1 & .6 & .0 & .0 & .0 & .0 & .0 & .4 & .1 & .2 & .4 & .6 \\ \mathbf{G}: & .0 & .0 & 1 & 1 & .9 & .9 & .1 & .0 & .0 & .0 & .0 & .0 \\ \mathbf{T}: & .7 & .2 & .0 & .0 & .1 & .1 & .0 & .5 & .8 & .7 & .3 & .4 \end{array}$$

- The probability of some DNA strings might be zero.

$$\Pr(\mathbf{TCGTGGATTTCC}|Profile) = .7 \cdot .6 \cdot 1 \cdot .0 \cdot .9 \cdot .9 \cdot .9 \cdot .5 \cdot .8 \cdot .7 \cdot .4 \cdot .6 = 0$$

due to fourth symbol

- **Solution:** adding psuedocounts

Laplace's rule of succession

$$\textit{Motifs} \quad \begin{matrix} T & A & A & C \\ G & T & C & T \\ A & C & T & A \\ A & G & G & T \end{matrix}$$

$$\text{COUNT}(\textit{Motifs}) \quad \begin{matrix} A: & 2 & 1 & 1 & 1 \\ C: & 0 & 1 & 1 & 1 \\ G: & 1 & 1 & 1 & 0 \\ T: & 1 & 1 & 1 & 2 \end{matrix} \qquad \text{PROFILE}(\textit{Motifs}) \quad \begin{matrix} 2/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 2/4 \end{matrix}$$

Laplace's Rule of Succession adds 1 to each element of COUNT(Motifs)

$$\text{COUNT}(\textit{Motifs}) \quad \begin{matrix} A: & 2+1 & 1+1 & 1+1 & 1+1 \\ C: & 0+1 & 1+1 & 1+1 & 1+1 \\ G: & 1+1 & 1+1 & 1+1 & 0+1 \\ T: & 1+1 & 1+1 & 1+1 & 2+1 \end{matrix} \qquad \text{PROFILE}(\textit{Motifs}) \quad \begin{matrix} 3/8 & 2/8 & 2/8 & 2/8 \\ 1/8 & 2/8 & 2/8 & 2/8 \\ 2/8 & 2/8 & 2/8 & 1/8 \\ 2/8 & 2/8 & 2/8 & 3/8 \end{matrix}$$

Gibbs Sampling

- Gibbs Sampler is a randomized algorithm that search for the minimum scoring motif in the space of all motifs

```
GIBBSSAMPLER(Dna, k, t, N)
    randomly select k-mers Motifs = (Motif_1, ..., Motif_t) in each string from Dna
    BestMotifs ← Motifs
    for j ← 1 to N
        i ← RANDOM(t)
        Profile ← profile matrix formed from all strings in Motifs except for Motif_i
        Motif_i ← Profile-randomly generated k-mer in the i-th sequence
        if SCORE(Motifs) < SCORE(BestMotifs)
            BestMotifs ← Motifs
    return BestMotifs
```

```
ttaccttaac        ttaccttaac
gatatctgtc        gatatctgtc
acggcgttcg   →    acggcgttcg
ccctaaagag        ccctaaagag
cgtcagaggt        cgtcagaggt
```

- **Example:** Suppose five DNA strings with implanted motif **ACGT** are given:

```
ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT
```

- At the initial step, the algorithm has chosen the following 4-mers (in red) and has selected the third string for removal

	ttACCTtaac		ttACCTtaac
	gATGTctgtc		gATGTctgtc
Dna	ccgGCGTtag	$\longrightarrow$	----------
	cactaACGAg		cactaACGAg
	cgtcagAGGT		cgtcagAGGT

Motifs

t	a	a	c
G	T	c	t
a	c	t	a
A	G	G	T

PROFILE(*Motifs*)

A:	3/8	2/8	2/8	2/8
C:	1/8	2/8	2/8	2/8
G:	2/8	2/8	2/8	1/8
T:	2/8	2/8	2/8	3/8

Motifs

t	a	a	c
G	T	c	t
a	c	t	a
A	G	G	T

PROFILE(*Motifs*)

A:	3/8	2/8	2/8	2/8
C:	1/8	2/8	2/8	2/8
G:	2/8	2/8	2/8	1/8
T:	2/8	2/8	2/8	3/8

The 4-mer probabilities in the deleted string

ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
$4/8^4$	$8/8^4$	$8/8^4$	$24/8^4$	$12/8^4$	$16/8^4$	$8/8^4$

$$\text{RANDOM}\left(\frac{4/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}, \frac{24/8^4}{80/8^4}, \frac{12/8^4}{80/8^4}, \frac{16/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}\right)$$

Assume the fourth 4-mer is chosen

It results in :

ttACCT**taac**
gAT**GTct**gtc
ccg**GCGT**tag
c**acta**ACGAg
cgtcag**AGGT**

```
ttACCTtaac        ----------
gATGTctgtc        gATGTctgtc
ccgGCGTtag   ⟶   ccgGCGTtag
cactaACGAg        cactaACGAg
cgtcagAGGT        cgtcagAGGT
```

```
ttACCTtaac        ttACCTtaac
gATGTctgtc        gATGTctgtc
ccgGCGTtag   ⟶   ccgGCGTtag
cactaACGAg        ----------
cgtcagAGGT        cgtcagAGGT
```

```
ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT
```

We can see it is slowly converging to the implanted motif!