Monotony of Surprise and Large-Scale Quest for Unusual Words

Stefano Lonardi

University of California, Riverside

joint work with A. Apostolico, M. E. Bock, F. Gong

Detection of unusual words

- **GIVEN**
  - a text *x*
  - a probabilistic *model* of the source which has generated *x*
- **FIND** all the substrings of *x* which are significantly more *frequent/rare* than the model-based expectation

Example

…ATGACAAGTCCTAAAAAGAGCGAAAACACAGGGTTGTTTGATTGTAGAAAATCACAGCG

>*MEK1*

CCACCCTTTTGTGGGGCTTCTATTTCAAGGACCTTCATTATGGAAACAGGGCGAGGTTGT

TTGTTCTTCCTGCATGTTGCGCGCAGTGCGTAAGAAAGCGGGACGTAAGCAGTTTAGCCA

TTCTAAAAGGGGCATTATCAGAATAAGAAGGCCCTATGAGGTATGATTGTAAAGCAAGTG

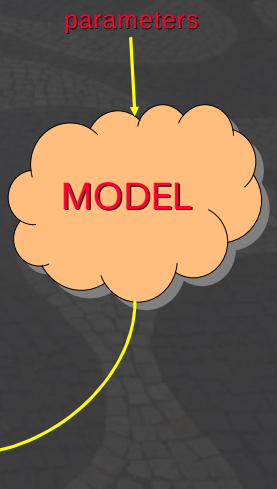
GTGTAAAATTGTGTGCTACCTACCGTATTAGTAGGAACAATTATGCAAGAGGGGTCCTGT

GCAAATAAAAAATATATATCTAGAAAAAGAGTAGGTAGGTCCTTCACAATATTGACTGAT

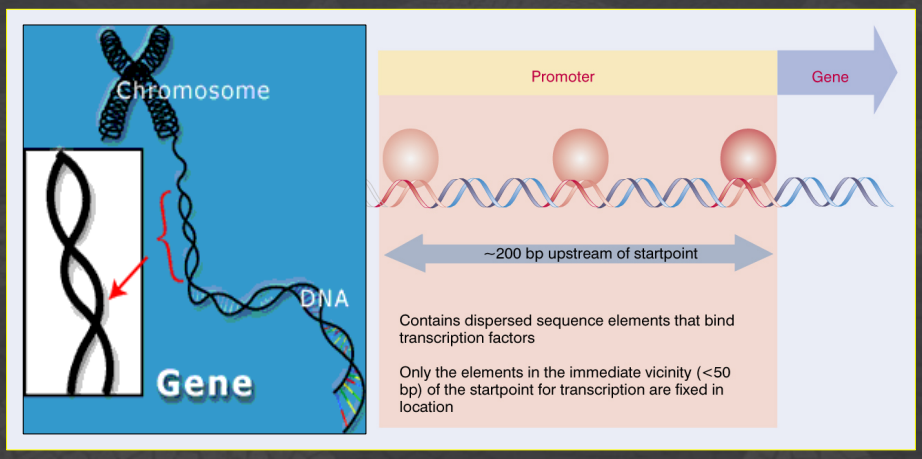
AGCGATCTCCTCACTATTTTTCACTTATATGCAGTATATTTGTCTGCTTATCTTTCATTA

AGTGGAATCATTTGTAGTTTATTCCTACTTTATGGGTATTTTCCAATCATAAAGCATACC

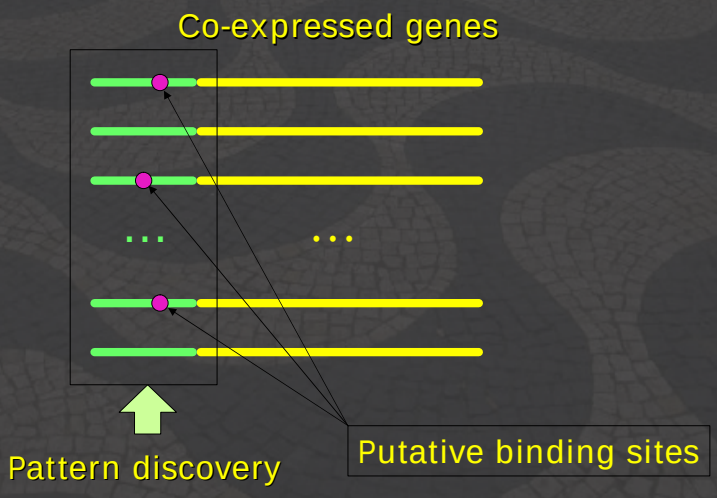
GTGGTAATTTAGCCGGGGAAAAGAAGAATGATGGCGGCTAAATTTCGGCGGC…



Transcription factors binding sites



Transcription factors binding sites



General framework

Which patterns do we count?

What do we expect, under the given model?

What is *unusual*?

How do we count *efficiently*?

How *many* patterns can be unusual?

How do we compute statistical parameters *efficiently*?

Notations

x : sequence, $|x| = n$

y : substring of x, $|y| = m$

$f(y)$: number of occurrences of y in x

Bernoulli model

Let Z_y be a r.v. for the number of occurrences of y,

p_a be the probability of $a \in \Sigma$, and $|y| = m \le (n+1)/2$

- $E(Z_y) = (n-m+1)\prod_{i=1}^{m} p_{y_{[i]}} = (n-m+1)\,\hat{p}$
- $Var(Z_y) = E(Z_y)(1-\hat{p}) - \hat{p}^2(n-m+1)(n-m) + 2\hat{p}B(y)$

where $B(y) = \sum_{d \in P(y)} (n-m+1-d) \prod_{i=m-d+1}^{m} p_{y_{[i]}}$

and $P(y)$ is the set of period lengths of y

Scores

$$z_1(y) = f(y) - E(Z_y)$$

$$z_2(y) = \frac{f(y) - E(Z_y)}{\sqrt{E(Z_y)}}$$

$$z_3(y) = \frac{f(y) - E(Z_y)}{\sqrt{E(Z_y)\ (1 - \hat{p})}}$$

$$z_4(y) = \frac{f(y) - E(Z_y)}{\sqrt{Var(Z_y)}}$$

where Z_y is a r.v. for the number of occurrences of y

What is “unusual” ?

Definition

Let y be a substring of x and $T \in \mathbb{R}^+$

- if $z(y) > T$, then y is over-represented
- if $z(y) < -T$, then y is under-represented
- if $|z(y)| > T$, then y is unusual

Problem definition

Given

- Sequence *x*
- Model *M*
- Type of count (*f*,…)
- Score function *z*
- Threshold *T*

Find

- The set of all unusual words in *x* w.r.t. *(f/…,z,M,T)*

Computational problems

- Counting “events” in strings (occurrences, …)
- Computing expectations, variances, and scores (under the given model)
- Detecting and visualizing unusual words

Combinatorial problem

- A sequence of size *n* could have $O(n^2)$ unusual words
- How to limit the set of unusual words?

Monotony of surprise

Theorem

Let C be a subset of words from text x. If $f(y)$ remains constant for all y in C, then any score of the type

$$z(y) = \frac{f(y) - E(y)}{N(y)}$$

is monotonically increasing with $|y|$ provided that

- $N(y)$ is monotonically decreasing with $|y|$
- $E(y)/N(y)$ is monotonically decreasing with $|y|$

Theorem

Score functions

$$z(y) = f(y) - E(Z_y)$$

$$z(y) = \frac{f(y) - E(Z_y)}{\sqrt{E(Z_y)}}$$

$$z(y) = \frac{f(y) - E(Z_y)}{\sqrt{E(Z_y)(1-\hat{p})}}$$

are monotonically increasing with $|y|$, for all y in class C

Theorem

If $p_{\max} < \min\left\{1/\sqrt[|y|]{4|y|}, \sqrt{2}-1\right\}$, then

$$z(y) = \frac{f(y) - E(Z_y)}{\sqrt{Var(Z_y)}}$$

is monotonically increasing with $|y|$, for all y in class C

Building the partition

abaababaabaababaabaaba

abaababaabaababaabaaba
aa aa

abaababaabaababaabababa
baa baa

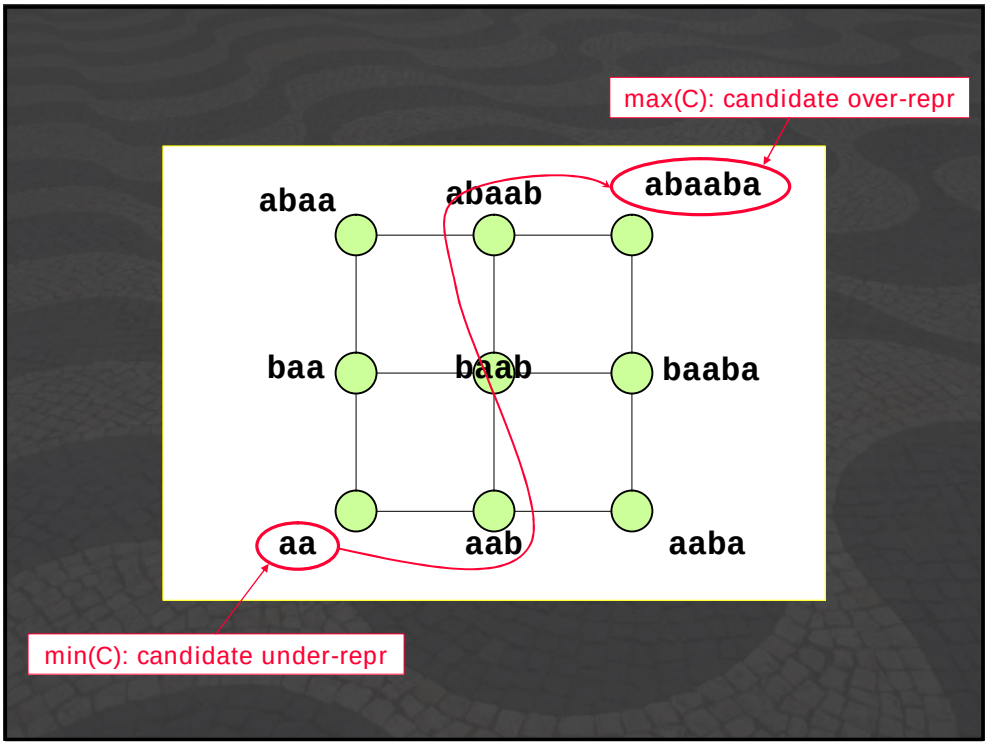
abaababaabaababaabababa
abaa abaa

abaababaabaababaabaababa
abaab abaab

abaababaabaababaabaababa
abaaba abaaba

abaaba baaba baabaaba baababa

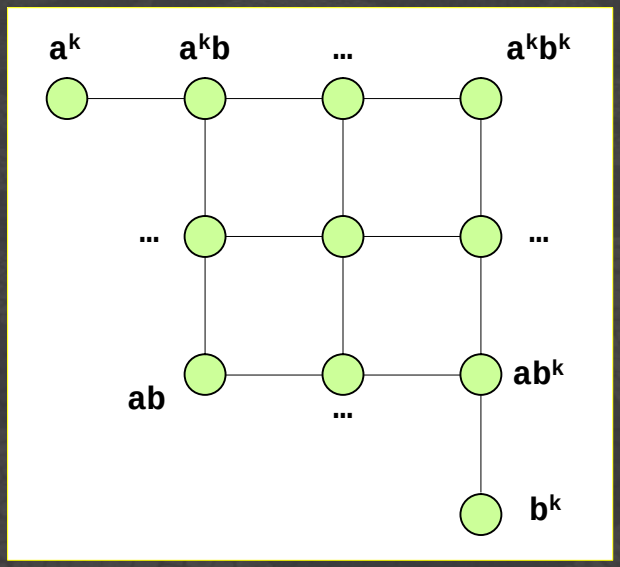
abaaba abaaba



x = abaababaabaababaababa

a (13)	b ba ab aba (8)	babaa babaab babaaba ababaa ababaab ababaaba aababaa aababaab aababaaba baababaa baababaab baababaaba abaababaa abaababaab abaababaaba (2)	…….. …….. …….. …….. …….. …….. …….. …….. …….. …….. …….. …….. …….. …….. …….. (1)
aa aab aaba baa baab baaba abaa abaab abaaba (4)	bab baba abab ababa ababb aababa baabab baababa (3)		

$x = a^k b^k$



The partition $\{C_1, C_2, \ldots, C_l\}$ of the set of all substrings of x, has to satisfy the following properties

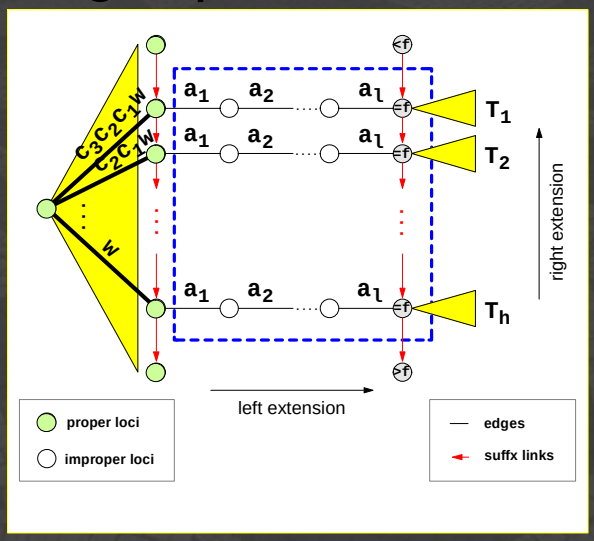
- $\min(C_i)$ and $\max(C_i)$ are unique
- all w in C_i belong to some $(\min(C_i), \max(C_i))$-path
- all w in C_i have the same count

for all $1 \leq i \leq l$.

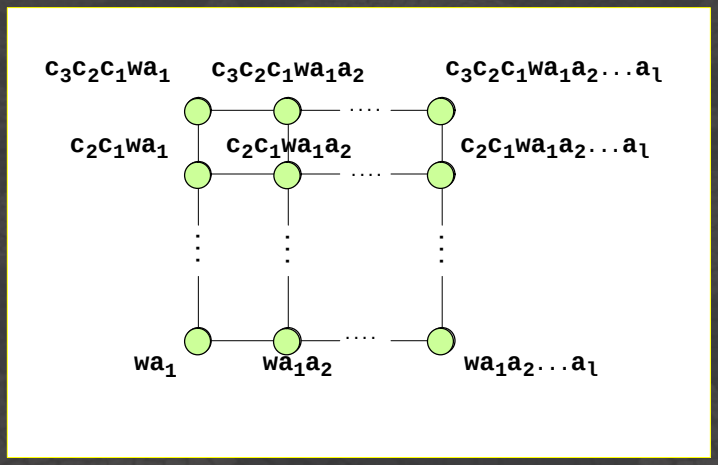
Suffix trees

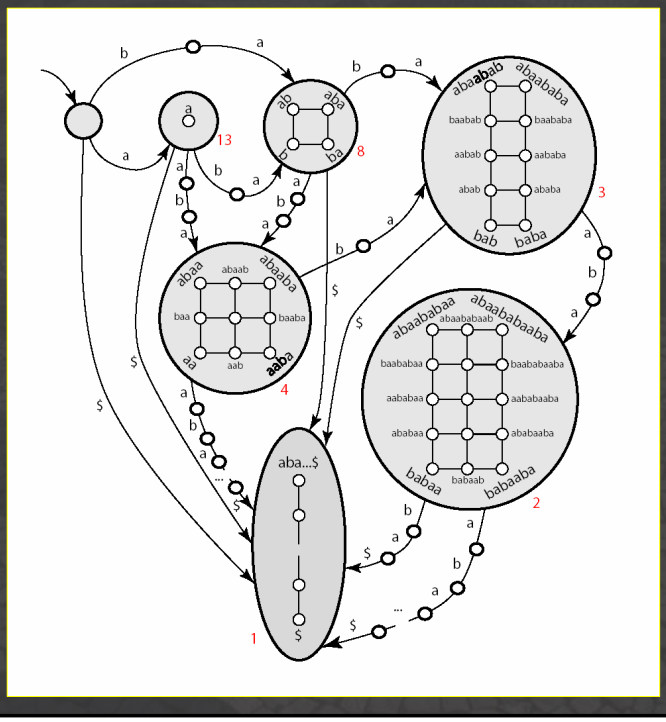
- Suffix trees can be built in *O(n)* time and space [W73,M76,U95,F97]
- Number of occurrences can be computed in *O(n)* time

Finding equivalence classes



Finding equivalence classes





Suffix Trees

- Equivalence classes can be computed in *O(n)* time (by merging isomorphic sub-trees)
- Expectations, variances and scores can be computed in *amortized constant time* per node [ABLX00]

Number of classes

Theorem

The number of classes is at most $2n$

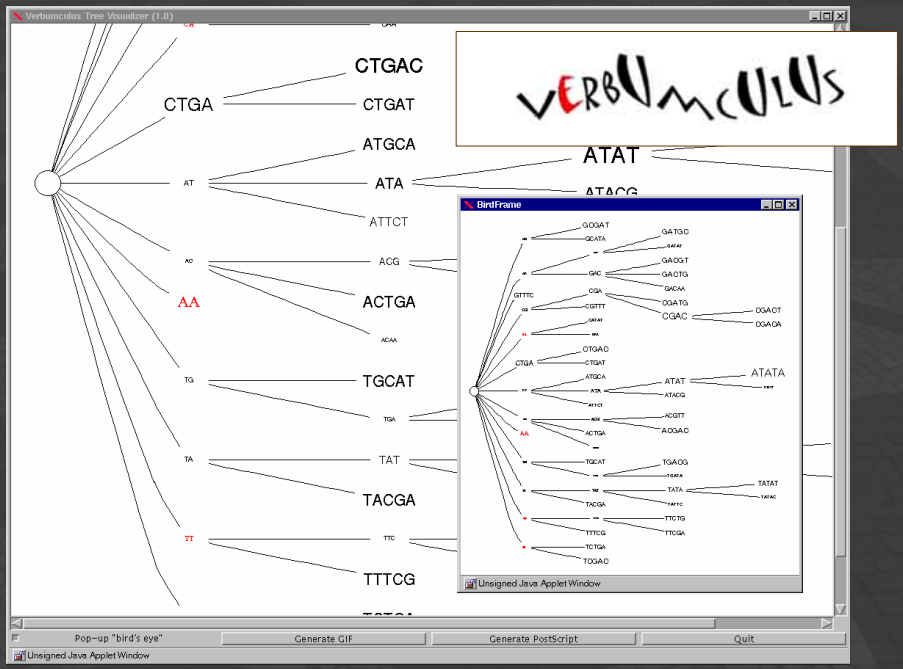
Algorithm

- Find the *O(n)* equivalence classes
- Compute expectation, variance and score on two words (*candidates*) in each equivalence class
- Visualize the scores of the candidates

Overall time/space complexity

Theorem:

The set of over- and under-represented words can be detected in $O(n)$ time and space



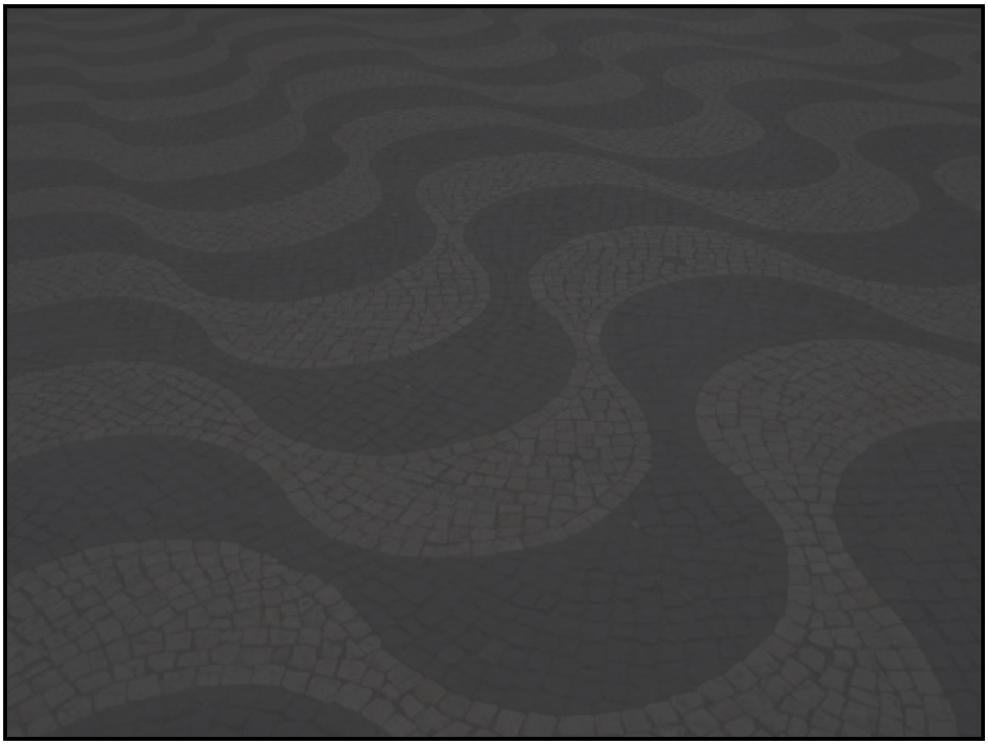
http://www.cs.ucr.edu/~stelo/Verbumculus/

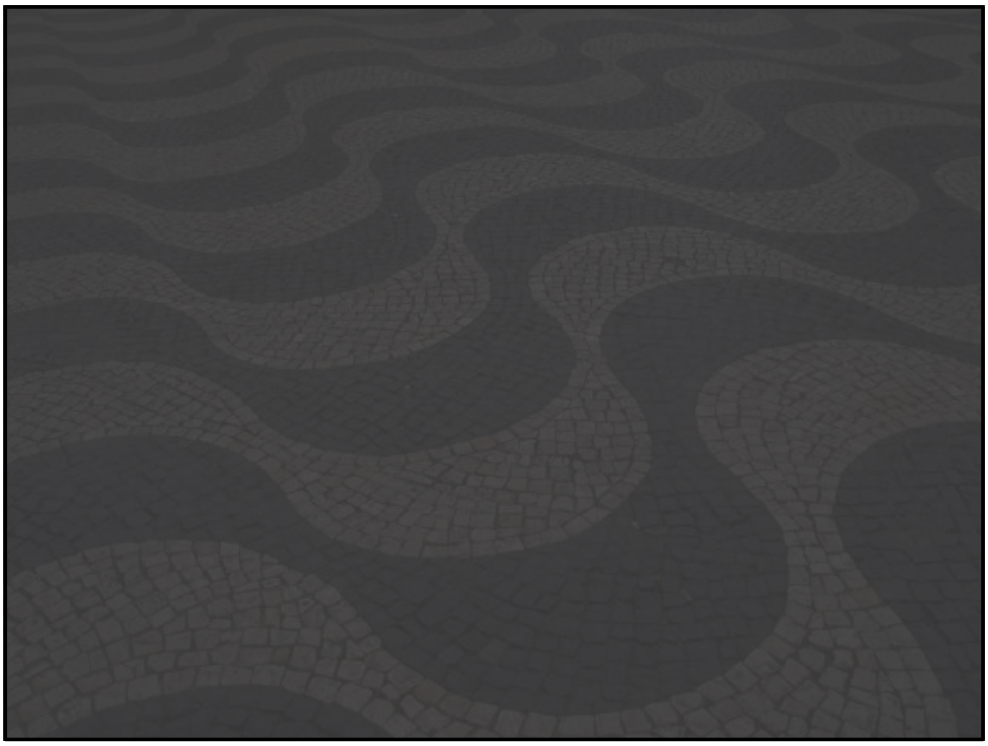
Conclusions

- Counts, expectations, variances and scores can be computed in *linear* time
- Exact patterns can be “discovered” in *linear* time and space
- Markov models and other types of counts can be handled within the same time-complexity

References

- “Monotony of Surprise and Large-Scale Quest for Unusual Words”, RECOMB, 2002, with A.Apostolico and M.E.Bock (to appear)
- “A Speed-up for the Commute between Subword Trees and DAWGs”, *Information Processing Letters*, 2001, with A.Apostolico (to appear)
- “Efficient Detection of Unusual Words”, *Journal of Computational Biology*, vol.7(1/2), 2000, with A.Apostolico, M.E.Bock and X.Xu
- “Linear Global Detectors of Redundant and Rare Substrings”, *IEEE Data Compression Conference*, 1999, with A.Apostolico and M.E.Bock





Example (1/2)

CCACCCTTTTGTGGGGCTTCTATTTCAAGGACCTTCATTATGGAAACAGGGCGAGGTTGT
TTGTTCTTCCTGCATGTTGCGCGCAGTGCGTAAGAAAGCGGGACGTAAGCAGTTTACCCA
TTCTAAAAGGGGCATTATCAGAATAAGAAGGCCCTATGAGGTATGATTGTAAAGCAAGTG
GTGTAAAATTGTGTGCTACCTACCGTATTAGTAGGAACAATTATGCAAGAGGGTTCGTGT
GCAAATAAAAAATATATATCTAGAAAAAGAGTAGGTAGGTCCTTCACAATATTGACTGAT
AGCGATCTCCTCACTATTTTTCACTTATATGCAGTATATTTGTCTGCTTATCTTTCATTA
AGTGGAATCATTTGTAGTTTATTCCTACTTTATGGGTATTTTCCAATCATAAAGCATACC
GTCGTAATTTAGCCGGGGAAAAGAAGAATGAGGCGGCAAATTTGGCGGCATTTCAT
TCATTCAAGTATAAAAGGGAGAGGTTTGACTAATTTTTTACTTGAGCTCCTTCTGGAGTG
CTCTTGTACGTTTCAAATTTTATTAAGGACCAAATATACAACAGAAAGAAGAAGAGCGGA

CACAGGCGCTACCATGAGAAATTTGTGGGTAATTAGATAATTGTTGGGATTCCATTGTTG
ATAAAGGCTATAATATTAGGTATACAGAATATACTAGAAGTTCTCCTCGAGGATATAGGA
ATCCTCAAAATGGAATCTATATTTCTACATACTAATATTACGATTATTCCTCATTCCGTT
TTATATGTTTATATTCATTGATCCTATTACATTATCAATCCTTGCCTTTCAGCTTCCTCT
AACATCGATGACAGGTTCTCATAACTTATGTCATCATCTTAACACCGTATATGATAATAT
ATTGATAATATAACTATTAGTTGATAGACGATAGTGGATTTTTATTCCAACAGAAGGAGT
GGATGGAAAAGTATGGAATTAAAGTAATCCATGTGGTAAATAAAATCACTAAGACTAGC
AACCACGTTTTGTTTTGTAGTTGAGAGTAATAGTTACAAATGGAAGATATATATCCGTTT
CGTACTCAGTGACGTACCGGCGTAGAAGTTGGCGGCATTTTGACAGATATATCAAAA
ATATTGTCATCAACTATACCATATACAACTTAGGATAAAAATACAGGTACAAAAACTATA

TTTCCTTCTGGTTCGTAGGCTTCTTCAAGTCCTTAATACCGCTTTTACCGACCCGATAGT
TATTAGTGTCCTTTTTTGTATAAGAATGGTTGATGCAAGTATTTTCTTCTTCGTTCACCA
AAGTTTTTGTCCTTGTCTAGCCACTCTTTCCTGATTGTGCATTACTATTAGATAACTGTAAT
TTGGTGCTTTTCCTGGAAAGTATACTTGTGATGTGGAAGTATTTTAAGTTCAAGTTTCTT
GTTTTCTTTCCTATTTATGCGGAAGGTACATAGAAGTTTGGCGGCAATACTTTTTCCG
CGGCTAATCCTATAGTAAAATGATCACTTTCATATAGAAAGTTGGTATATAAAGTGTCAA
CTAAGAGAGAAATAGTTCGAACCAGGTGTATTTTAAATCAACTATCGGGAAGTATGGACT
GGTGGTATAATCGAATTACATAGTCCTTTTACCTTCATTAGTAGTACTTAAGTGTCACCC
GCCTGGGGATTTTGCTCTCATAGAAGTAAAAGGGTAGTGCTATGGGAGCACATTAGGTAG
TTCAGTTACGTTTTATGGCAGTCACTGTTTTCGCAAAGACTCCCAGACACGGGCATTAAA

CACTCATCTCATAAGCTTAGCTGAATGGATAGGCTTGCTTTCTGATGGAAATTTGCCTTG
CTTTTCCAACTATTCCATTACTCAGGTTTTATTTTTTTATTTTGTAATATGGGGAGAAGG
CCGGCAGAATATTTACGGACAAATGAATAAATTGGATTGGATTGACTAGTGGAACGTGTA
AAGATCGCGATACTCCGTACCAATCACCGAAAGATTGCCCGTAACCGAAATGACTCCATT
CTCTGAATTTTTTGTGAAACCAATATCTGAGACTCTTCCTTCATCTTATCAACGTATTGT
TCAGTCAATTAAGTAAGAAGTATATTTGAGCGCAGCCTTAATCATATATAGCACCAGTTA
TATGTTTGCCCCTCTCTTGAGTTGAAAAACACATAATACATAGTACTGTACTTTTCTCTT
TTTCATCGTTGGCGAAAATATAATCTTTCTCAAAAATATATATATATGTATATATATCCT
TAGATTTGCCGTTGACAATAAGGTTGGCGGCAATCTACGAAATGCGAGGCGGTTAAAAG
AGAGTGACAACATTTCATAAAAATATTCTGATCTCAAACTGAAGACATAAAATAAGGAT

CAAATATCTACAATGCCGTCTGCTTTATGTCTTTTTCTAAAGGCATCGATTTTATGTGTG
GATAATTGCATCGCAGTAATATGTAGAGCACAATTTGTAGAAATCGGAATTGGAGGTATC
GGATCTTGTTGAATATCCACCAATGTCTTACCCCTGTATTTTAACAAGAGTTTACGCTGT
TATATGGTTAAAGGTGTGGACGCCTTGAAGGTTTACCTTACCGAATGACACCTTTACAAT
AGTCAGATCACGTTCTGTGGCGTTATCCAAAGTTAGCGCAGTTTTCCGATGGTCCAATGT
AATCATTAGAAATAGTAAAAACTGTGTAATGGTAAAGATTGTGTCACTGGAAAAAAACTG
CTACAAATAATAAATAAATAAAAAAATACGAAAGCACAGTACTACGGGTGCCTCCACAAA
TAGATAAGAAACCAAGCGGAGACATGCGTTTAGATGAGGATATAAATTATTTATACAACC
AGACTATATAAAAGAGCATCTAGTTTACCTGTTATGATGAATGGACATTCGCTACATATC
TTACTCTCTATTTGTTAAAAAAAATTACAAAGAGAACTACTGCATATATAAATAACATAC

ATAAAGGTTTCTTCTGTGGTTTAGATATGCTATACGGGGGAAACTTTGTTTACACACGGCTC
GCGGGAATCCTTAGGGGAAAACATTGCGCTGACTTTCCCCAGAGTTGTTGCCACAACATA
AGGTGGTTTGGMGTTGTTGAACAAATCCGTCCTTGGGTCATTGAATCWTTGCTTGGGGGT
ATCTCAAAAGAGCCGCAAACTAATAGCGCCGCACATTGGACGGATTTATCCGGTGGTGATCG
ACTAGGGGGGAAGAGGTCACGAGCTATTTTTTCTTGCAGAAAAAAAGTGTGAGCTTTTCC
GTAGCTAGACGTTATCAGGGCCTCAGCAATGGGAGCCACAGCGGAAAAACAATAAGAAT
GGTAAGCGCAATTACCTTTTCAGCGTTACATTCGTATGAAATTGGTGACGTTAATCTAAA
GATAGTGATGTCTGCAAAGGGGCGATTATTCTGAGGTTGAGGTATATAAGACTATTA
AAACTTGGTTCTTAGATATGGTGTTCGTTGCTCATTATTAAGTTTCAGGAAGAATATC
AACACATATCATAACAGGTTCTCAAAACTTTTGTTTATAATAACTAGTAACAAGAAAA

Text "events"

- Occurrences
  - distance constraints (non-overlapping, adjacent, max distance, ...)
  - sliding window
  - ...
- Colors
- ...

Exact or *approximate*?

Bernoulli Model (colors)

Let W_y be a r.v. for the number of colors of y in $\{x_1, x_2, \ldots, x_k\}$, and $E\left(Z_y^i\right)$ be the expected number of occurrences of y in the i-th sequence $(1 \le i \le k)$,

- $E\left(W_y\right) = \sum_{i=1}^{k} \left(1 - e^{-E\left(Z_y^i\right)}\right)$

Scores based on colors

$$z_7(y) = c(y) - E(W_y)$$

$$z_8(y) = \frac{c(y) - E(W_y)}{\sqrt{E(W_y)}}$$

$$z_9(y) = \frac{\left(c(y) - E(W_y)\right)^2}{E(W_y)}$$

where W_y is a r.v. for the number of colors of y

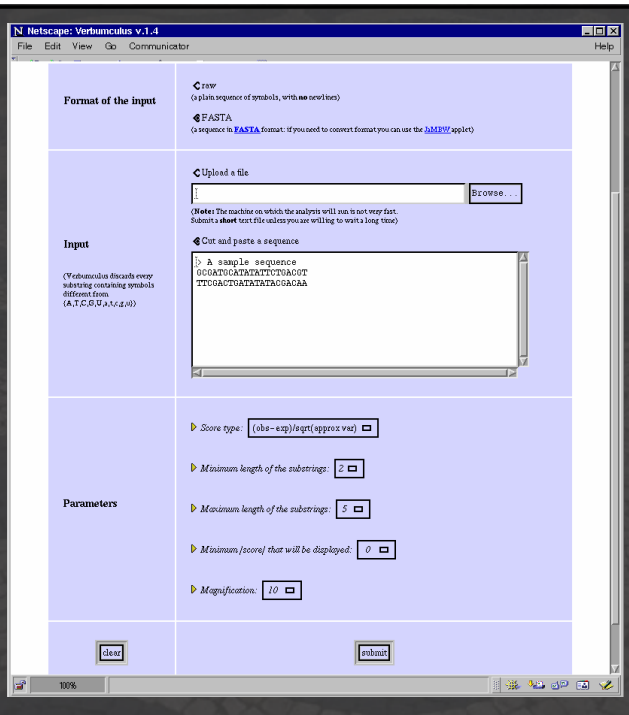
Main result

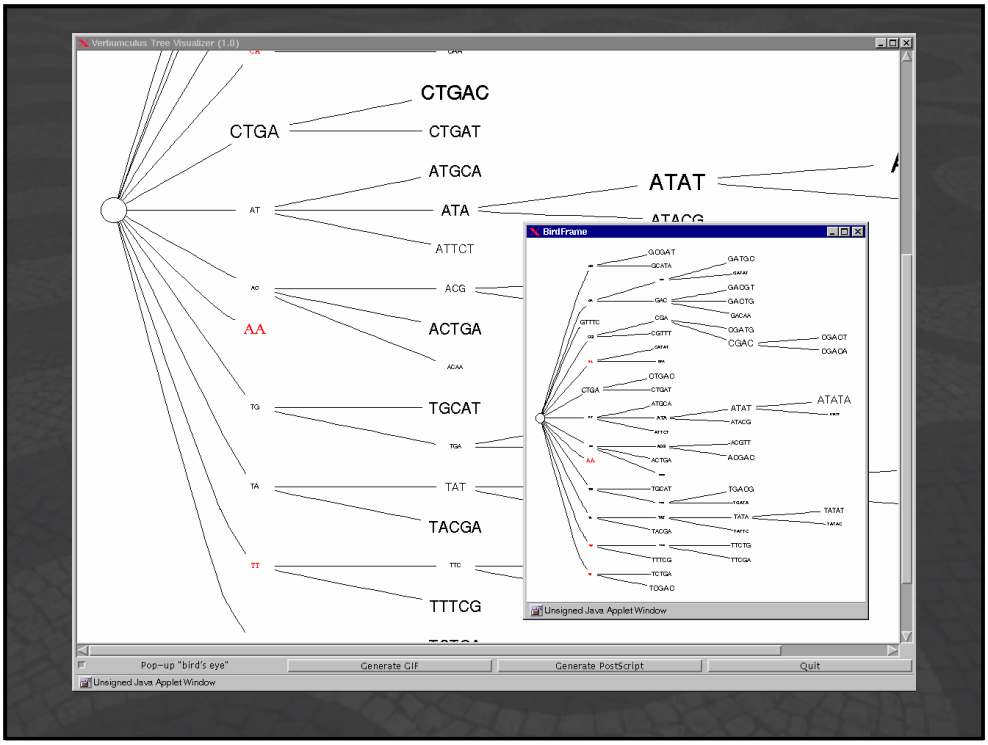
An efficient algorithm for the problem of detecting words that are, by some statistical measure, surprisingly frequent or rare in the context of larger sequences

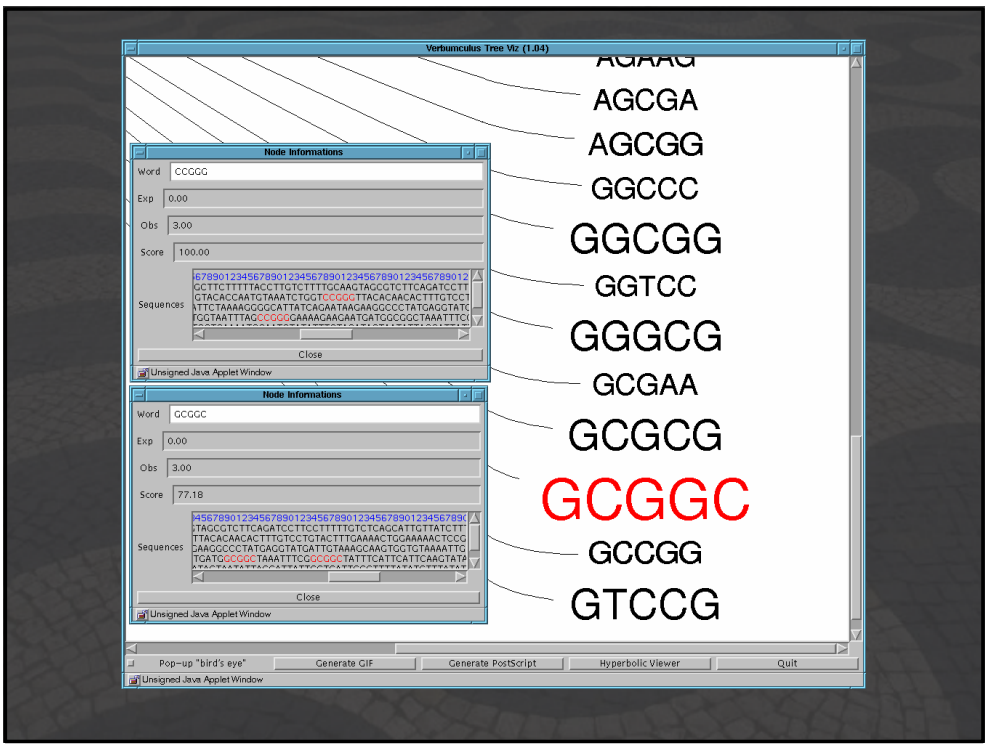
VERBUMCULUS

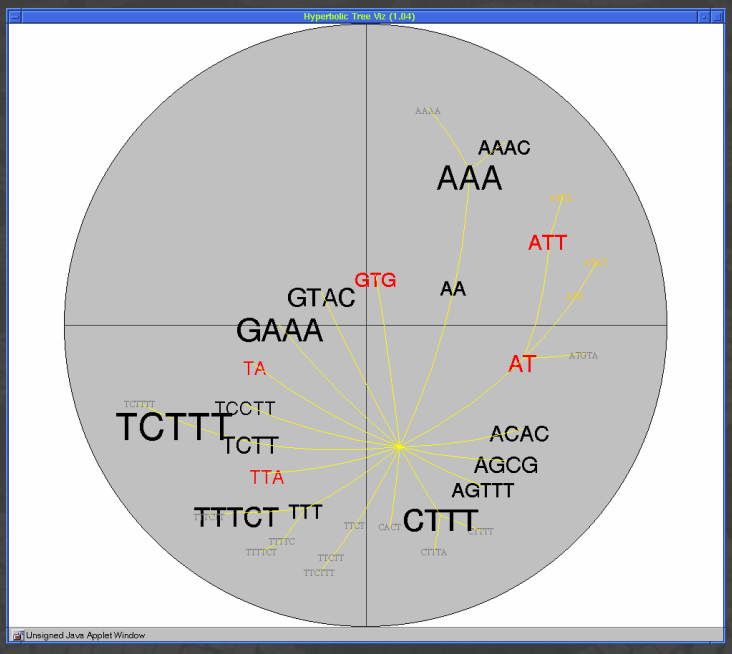
http://www.cs.ucr.edu/~stelo/Verbumculus/

- Verbumculus = Verbum + Dot + TreeViz
- Verbum builds and annotates the tree
- Dot and TreeViz draw the tree; the font size of the labels is **PROPORTIONAL** to the score
- C++/STL + Perl + Java ≈ 15,000 lines
- Solaris/Linux




Verbumculus Tree Visualizer (1.0)
CTGAC
CTGA
CTGAT
ATGCA
AT
ATA
ATAT
ATTCT
AC
ACG
AA
ACTGA
TG
TGCAT
TA
TAT
TACGA
TT
TTC
TTTCG
BirdFrame
Pop-up "bird's eye"
Generate GIF
Generate PostScript
Quit
Unsigned Java Applet Window


Verbumculus Tree Viz (1.04)
AGCGA
AGCGG
GGCCC
GGCGG
GGTCC
GGGCG
GCGAA
GCGCG
GCGGC
GCCGG
GTCCG
Node Informations
Word
CCGGG
Exp
0.00
Obs
3.00
Score
100.00
Sequences
Close
Unsigned Java Applet Window
Node Informations
Word
GCGGC
Exp
0.00
Obs
3.00
Score
77.18
Sequences
Close
Unsigned Java Applet Window
Pop-up "bird's eye"
Generate GIF
Generate PostScript
Hyperbolic Viewer
Quit
Unsigned Java Applet Window



Tests and Experiments

- Validation on simulated data
- Experiments on real data
  - promoters/regulatory elements discovery
  - UTRs analysis
  - mDNA analysis

Hypothesis: "Unusually frequent" patterns in the upstream sequence of a set of *co-expressed* genes are plausible binding sites implicated in transcriptional regulation

Sets of *co-expressed* genes can be identified, e.g., by DNA microarray experiments

Pattern Discovery Tools

- Exact patterns: Yeast-Tools, R'MES, WordUp (GCG), ...
- Flexible patterns: MEME (UCSD), YEBIS, SPEXS (EBI), Gibbs Sampler, BlockMaker, Teiresias (IBM), PRATT, Consensus, Winnower (UCSD), Projection (UW), ...

Typical algorithms

Naïve approach ☆

Enumerate and test all words composed by l symbols, for $1 \leqslant l \leqslant n$

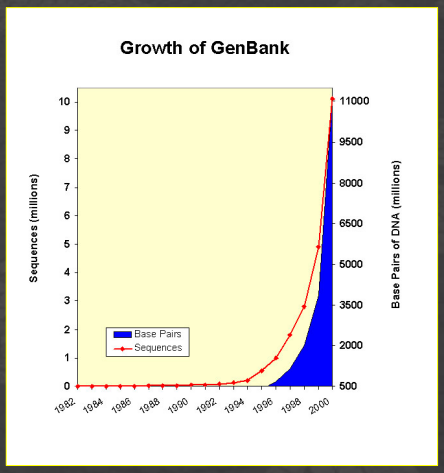
Naïve approach 🕐

Enumerate and test all words which occur in the sequences

Biomolecular Databases

- Massive
- Growing exponentially

Example: GenBank contains approximately 11,720,000,000 bases in 10,897,000 sequence records as of February 2001



$n = 1{,}000{,}000 \quad S = \{A, C, G, T\}$

Naïve approach ☆

Words to be tested $O(|S|^n)$

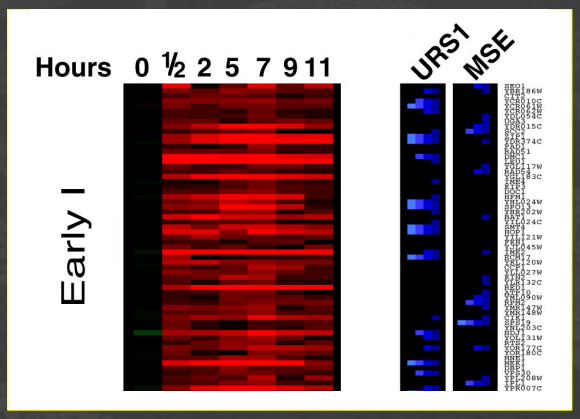
in this case $\propto 4^{1{,}000{,}000}$

Naïve approach 🕐

Words to be tested $O(n^2)$

in this case $\propto 1{,}000{,}000^2$

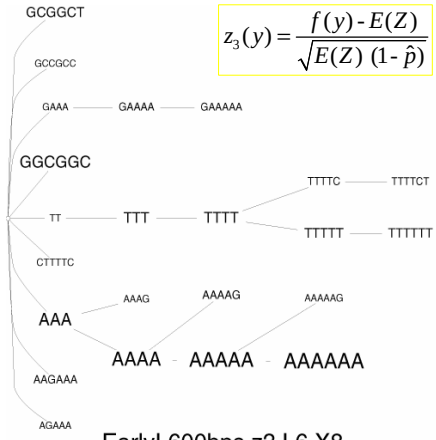
Cluster Early I



Dataset from "The Transcriptional Program of Sporulation in Budding Yeast", by S.Chu, J.L.DeRisi, M.B.Eisen, J.Mulholland, D.Bodstein, P.O.Brown, I.Herskowitz, *Science*, 1998

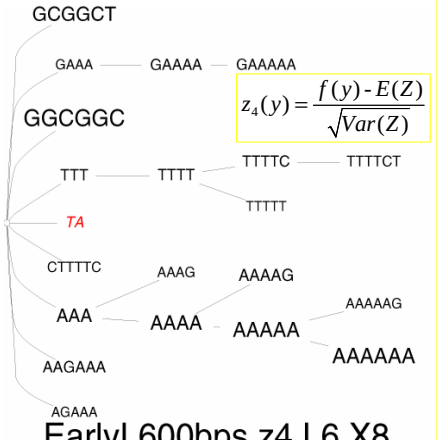
Analysis of EarlyI (1/3)

$$z_3(y) = \frac{f(y) - E(Z)}{\sqrt{E(Z)\,(1-\hat{p})}}$$



EarlyI.600bps.z3.L6.X8

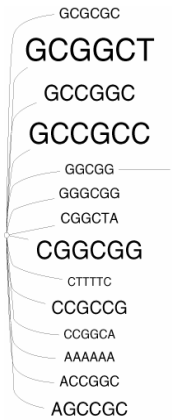
$$z_4(y) = \frac{f(y) - E(Z)}{\sqrt{Var(Z)}}$$



EarlyI.600bps.z4.L6.X8

Analysis of EarlyI (2/3)

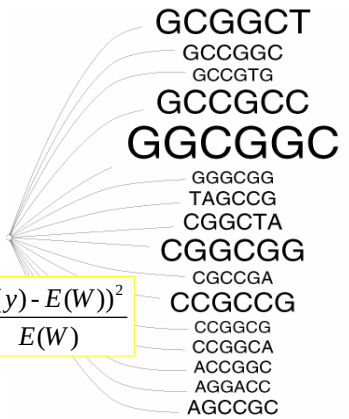
$$z_2(y) = \frac{f(y)}{E(Z)}$$



GGCGGC

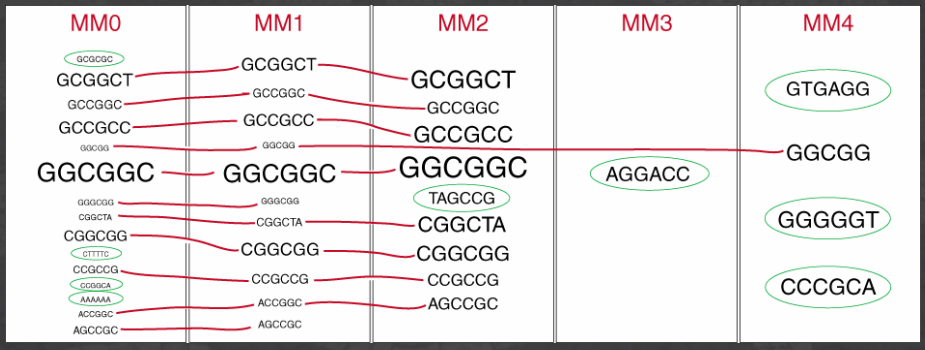
EarlyI.600bps.z2.L6.X4

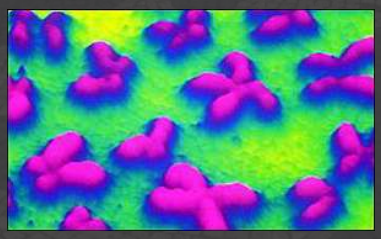
$$z_9(y) = \frac{(c(y) - E(W))^2}{E(W)}$$

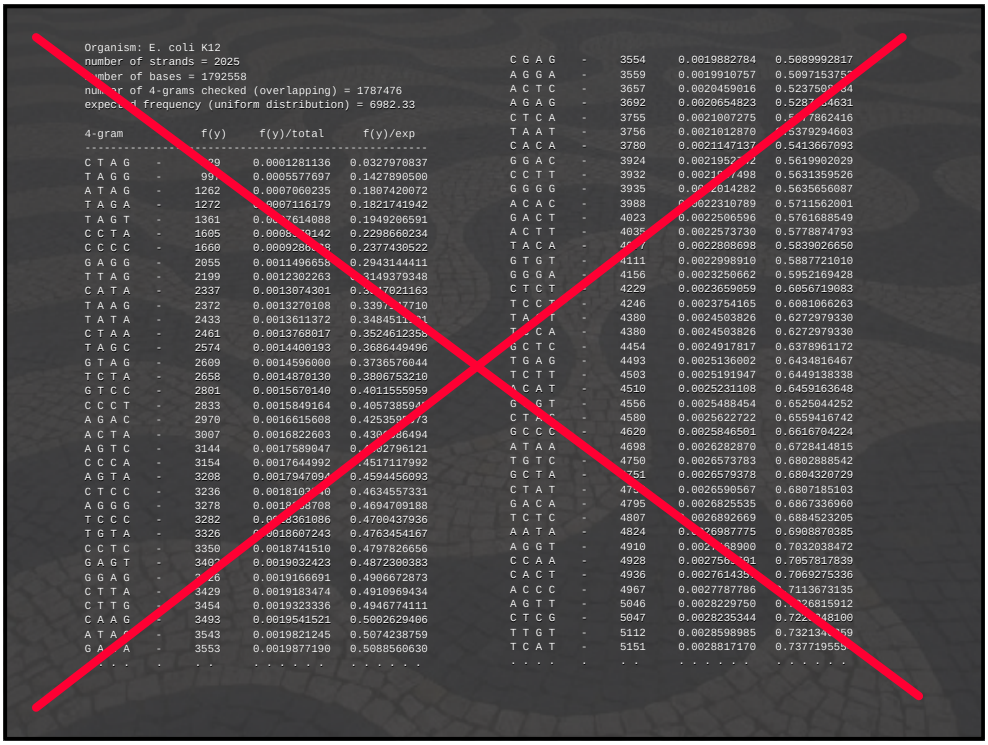


EarlyI.600bps.z8.L6.X8

Analysis of Earlyl (3/3)







```
Organism: E. coli K12
number of strands = 2025
number of bases = 1792558
number of 4-grams checked (overlapping) = 1787476
expected frequency (uniform distribution) = 6982.33

4-gram          f(y)     f(y)/total      f(y)/exp
-------------------------------------------------
C T A G    -      229     0.0001281136    0.0327970837
T A G G    -      999     0.0005577697    0.1427890500
A T A G    -     1262     0.0007060235    0.1807420072
T A G A    -     1272     0.0007116179    0.1821741942
T A G T    -     1361     0.0007614088    0.1949206591
C C T A    -     1605     0.0008980142    0.2298660234
C C C C    -     1660     0.0009286878    0.2377430522
G A G G    -     2055     0.0011496658    0.2943144411
T T A G    -     2199     0.0012302263    0.3149379348
C A T A    -     2337     0.0013074301    0.3346021163
T A A G    -     2372     0.0013270108    0.3396177710
T A T A    -     2433     0.0013611372    0.3483511791
C T A A    -     2461     0.0013768017    0.3523612358
T A G C    -     2574     0.0014400193    0.3685449496
G T A G    -     2609     0.0014596000    0.3735576044
T C T A    -     2658     0.0014870130    0.3805753210
G T C C    -     2801     0.0015670140    0.4011555959
C C C T    -     2833     0.0015849164    0.4057385947
A G A C    -     2970     0.0016615608    0.4253595273
A C T A    -     3007     0.0016822603    0.4306586494
A G T C    -     3144     0.0017589047    0.4502796121
C C C A    -     3154     0.0017644992    0.4517117992
A G T A    -     3208     0.0017947094    0.4594456093
C T C C    -     3236     0.0018103760    0.4634557331
A G G G    -     3278     0.0018338708    0.4694709188
T C C C    -     3282     0.0018361086    0.4700437936
T G T A    -     3326     0.0018607243    0.4763454167
C C T C    -     3350     0.0018741510    0.4797826656
G A G T    -     3402     0.0019032423    0.4872300383
G G A G    -     3426     0.0019166691    0.4906672873
C T T A    -     3429     0.0019183474    0.4910969434
C T T G    -     3454     0.0019323336    0.4946774111
C A A G    -     3493     0.0019541521    0.5002629406
A T A C    -     3543     0.0019821245    0.5074238759
G A T A    -     3553     0.0019877190    0.5088560630
. . . .    .     . .      . . . . . .    . . . . . .

C G A G    -     3554     0.0019882784    0.5089992817
A G G A    -     3559     0.0019910757    0.5097153757
A C T C    -     3657     0.0020459016    0.5237904634
A G A G    -     3692     0.0020654823    0.5287954631
C T C A    -     3755     0.0021007275    0.5377862416
T A A T    -     3756     0.0021012870    0.5379294603
C A C A    -     3780     0.0021147137    0.5413667093
G G A C    -     3924     0.0021953772    0.5619902029
C C T T    -     3932     0.0021998498    0.5631359526
G G G G    -     3935     0.0022014282    0.5635656087
A C A C    -     3988     0.0022310789    0.5711562001
G A C T    -     4023     0.0022506596    0.5761688549
A C T T    -     4035     0.0022573730    0.5778874793
T A C A    -     4057     0.0022808698    0.5839026650
G T G T    -     4111     0.0022998910    0.5887721010
G G G A    -     4156     0.0023250662    0.5952169428
C T C T    -     4229     0.0023659059    0.6056719083
T C C T    -     4246     0.0023754165    0.6081066263
T A G T    -     4380     0.0024503826    0.6272979330
T G C A    -     4380     0.0024503826    0.6272979330
G C T C    -     4454     0.0024917817    0.6378961172
T G A G    -     4493     0.0025136002    0.6434816467
T C T T    -     4503     0.0025191947    0.6449138338
A C A T    -     4510     0.0025231108    0.6459163648
G G G T    -     4556     0.0025488454    0.6525044252
C T A C    -     4580     0.0025622722    0.6559416742
G C C C    -     4620     0.0025846501    0.6616704224
A T A A    -     4698     0.0026282870    0.6728414815
T G T C    -     4750     0.0026573783    0.6802888542
G C T A    -     4751     0.0026579378    0.6804320729
C T A T    -     4781     0.0026590567    0.6807185103
G A C A    -     4795     0.0026825535    0.6867336960
T C T C    -     4807     0.0026892669    0.6884523205
A A T A    -     4824     0.0026987775    0.6908870385
A G G T    -     4910     0.0027468900    0.7032038472
C C A A    -     4928     0.0027569491    0.7057817839
C A C T    -     4936     0.0027614356    0.7069275336
A C C C    -     4967     0.0027787786    0.7113673135
A G T T    -     5046     0.0028229750    0.7226815912
C T C G    -     5047     0.0028235344    0.7228248100
T T G T    -     5112     0.0028598985    0.7321340759
T C A T    -     5151     0.0028817170    0.7377195555
. . . .    .     . .      . . . . . .    . . . . . .
```

Definition:

Given a substring w of x the implication of w in x, denoted by $imp_x(w)$, is the string uwv, such that

- every time w occurs in x, it is preceded by u and followed by v
- u and v are maximal

Definition:

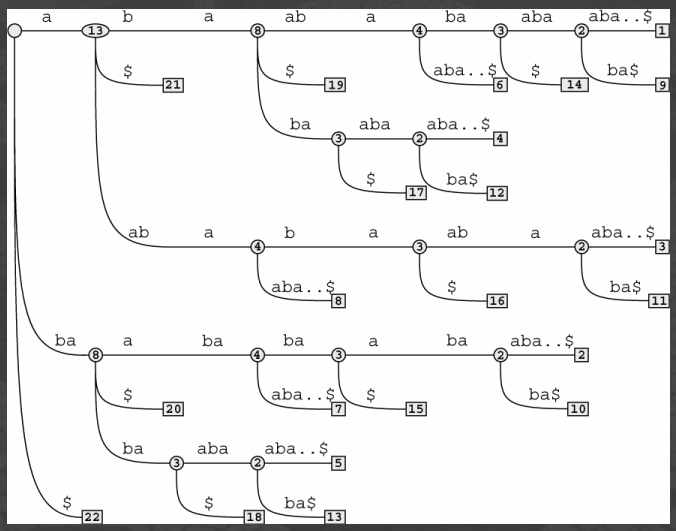
$$y \equiv_x w \Leftrightarrow imp_x(y) = imp_x(w)$$

Finding Equivalence Classes

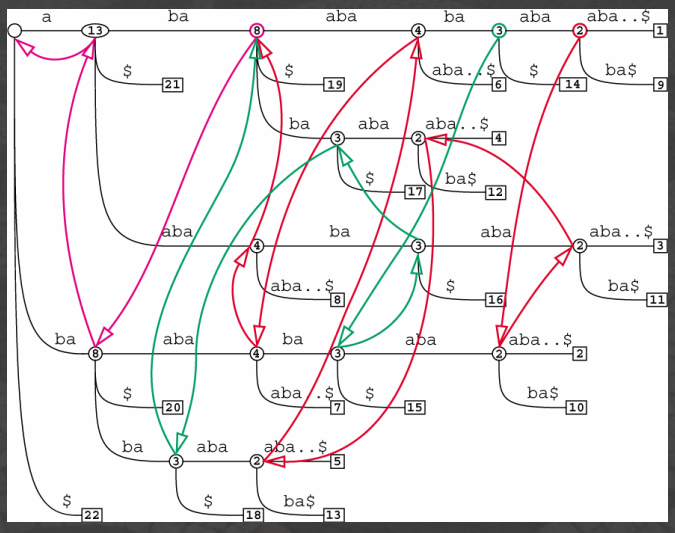
abaababaabaababaababa$

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

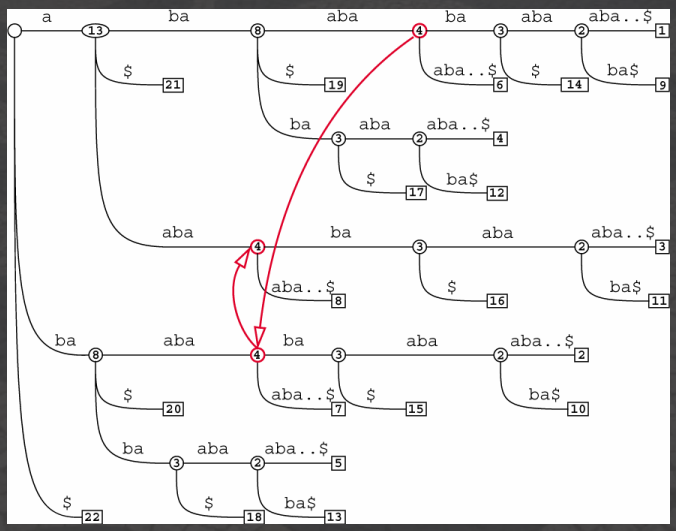
Finding Equivalence Classes



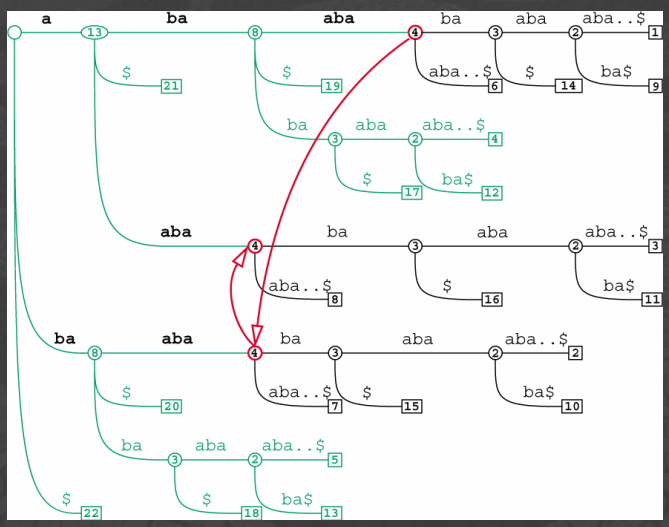
Finding Equivalence Classes



Finding Equivalence Classes



Finding Equivalence Classes



What's next?

- extension to other types of count and hidden Markov models
- estimation of statistical parameters by "shuffling"
- more experiments on biosequences and in other domains
- extension to approximate/flexible patterns

How to choose the threshold

$$\mathrm{P}\left(\left|\frac{f(y)-E(Z_y)}{\sqrt{Var(Z_y)}}\right|>2\right)=.0456$$