Pattern Discovery in Bioinformatics

Jaak Vilo
vilo@ut.ee

http://biit.cs.ut.ee
Topics

- Bioinformatics
- Pattern discovery
- Microarray data
- ...

[Image of a building]
Pattern Discovery

1. Choose the **language** (formalism) to represent the patterns (search space)
2. Choose the **rating** for patterns, to tell which is “better” than others
3. Design an **algorithm** that **finds the best patterns** from the pattern class, **fast**.

Brazma A, Jonassen I, Eidhammer I, Gilbert D.
Approaches to the automatic discovery of patterns in biosequences.
Bioinformatics:

- Have the right data
  (real, relevant, interesting)

- Interpret and report the results
  (make someone’s life easier)

- Contribute to the field of biology
Study of biological data with the goal to better understand biology  (JV)
Eukaryotic genome can be thought of as six Levels of DNA structure.

The loops at Level 4 range from 0.5kb to 100kb in length.

If these loops were stabilized then the genes inside the loop would not be expressed.
DNA determines function (?)

DNA → Protein → Structure

GenBank / EMBL Bank → SwissProt/TrEMBL → PDB/Molecular Structure Database

4 Nucleotides → 20+ Amino Acids

(3nt → 1 AA)

Function?
A Simple Gene

A: [Diagram of DNA with modifications highlighted]

B: [Diagram of DNA with modifications highlighted]

C: [Diagram of DNA with modifications highlighted]

DNA: ATCGAAAT TAGCTTTA +Modifications

Upstream/promoter

Downstream
Species and individuals

- Animals, plants, fungi, bacteria, …
- Species
- Individuals

www.tolweb.org
Recent findings suggest that each step regulating gene expression (from transcription to translation) is a subdivision of a continuous process. In this contemporary view of gene expression, each stage is physically and functionally connected to the next, ensuring that there is efficient transfer between manipulations and that no individual step is omitted (see text for details).
http://www.youtube.com/watch?v=bk7PW1FKMTI
Gene regulation

- How are all genetic entities regulated?
- Networks
  - parts lists and connections
  - parameters and dynamics
Possible mechanisms of action for secreted protein function in cell proliferation, either by intracellular second messengers pathways or by nuclear import. FT: transcription factor; Co-reg: Co-regulator.
Model of RNA Polymerase II
Transcription Initiation

The machinery depicted here encompasses over 85 polypeptides in ten (sub) complexes: core RNA polymerase II (RNAPII) consists of 12 subunits; TFIIH, 9 subunits; TFIIE, 2 subunits; TFIIF, 3 subunits; TFIIB, 1 subunit, TFIID, 14 subunits; core SRB/mediator, more than 16 subunits; Swi/Snf complex, 11 subunits; Srb10 kinase complex, 4 subunits; and SAGA, 13 subunits.

Dissecting the Regulatory Circuitry of a Eukaryotic Genome
Chen and Rajewsky *Nature Reviews Genetics* 8, 93–103 (February 2007) | doi:10.1038/nrg1990
In genetics, microRNAs (miRNA) are single-stranded RNA molecules of about 21-23 nucleotides in length, which regulate gene expression. miRNAs are encoded by genes that are transcribed from DNA but not translated into protein (non-coding RNA); instead they are processed from primary transcripts known as pri-miRNA to short stem-loop structures called pre-miRNA and finally to functional miRNA. Mature miRNA molecules are partially complementary to one or more messenger RNA (mRNA) molecules, and their main function is to downregulate gene expression. They were first described in 1993 by Lee and colleagues [1], yet the term microRNA was only introduced in 2001 in a set of three articles in Science (26 October 2001).[2]
Figure 1 | The molecular hallmarks of lin-4, the founding member of the microRNA family. a | The precursor structure and mature microRNA (miRNA) sequence of lin-4. b | Sequence complementarity between lin-4 (red) and the 3'-untranslated region (UTR) of lin-14 mRNA (blue). lin-4 is partially complementary to 7 sites in the lin-14 3' UTR; its binding to these sites of complementarity brings about repression of LIN-14 protein synthesis. RISC, RNA-induced silencing complex.
Figure 2 | The current model for the biogenesis and post-transcriptional suppression of microRNAs and small interfering RNAs. The nascent pri-microRNA (pri-miRNA) transcripts are first processed into ~70-nucleotide pre-miRNAs by Drosha inside the nucleus. Pre-miRNAs are transported to the cytoplasm by Exportin 5 and are processed into miRNA:miRNA* duplexes by Dicer. Dicer also processes long dsRNA molecules into small interfering RNA (siRNA) duplexes. Only one strand of the miRNA:miRNA* duplex or the siRNA duplex is preferentially assembled into the RNA-induced silencing complex (RISC), which subsequently acts on its target by translational repression or mRNA cleavage, depending, at least in part, on the level of complementarity between the small RNA and its target. ORF, open reading frame.
<table>
<thead>
<tr>
<th>Accession</th>
<th>MI0004635</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>mmu-mir-678</td>
</tr>
<tr>
<td>Symbol</td>
<td>MGI:Mirn678</td>
</tr>
<tr>
<td>Description</td>
<td>Mus musculus miR-678 stem-loop</td>
</tr>
</tbody>
</table>

GUGGACUGUGACUUGCAGAGCUGUGCUCCAAUAUGAGAGAUG
GCCAUGCACCCCGUGUCUCGGUGCAAGGACUGGAGGUGGCAGU
Alignment of microRNA targets

- **CtTCCA-TCCTT--G-ACCGAGAt**  ENSMUSG00000041359
- **tCTtCAtaCCcTcaGgACCGAGAC**  ENSMUSG00000032436
- **CCTttAGcCCTT--GggCtGgGA**  ENSMUSG00000028581
- **CCatttGaCtcc--aCAcTGA**  ENSMUSG00000034006
- **gCTCCAGgCCTT--GgACcTAggC**  ENSMUSG0000001053
- **CactgccTaacT--GCACtGAGAt**  ENSMUSG00000032470
- **ggaCCAGgttTT--GCACCaAGgC**  ENSMUSG00000053175
- **CCTCagGaCCTT--GtgcTGA**  ENSMUSG0000004040
- **agagaccTCgaa--GaAcTGA**  ENSMUSG00000031163
- **tagCCTGTCCTTctG--ACTGAGAC**  ENSMUSG0000006342
Sequence patterns in BI
Biological applications

■ DNA:
  – Gene regulation (promoters, TF binding)
  – Gene prediction (including TSS, to polyA site)
  – Repeats, duplications, tandem repeats, etc sequence features

■ RNA
  – Splicing of the mRNA
  – microRNA – targeting mRNA-s
  – Secondary structure, 3D structure

■ Proteins:
  – Protein families and their functional conserved elements
  – Active sites and protein-protein interactions
  – 3-D structure of proteins
Gene Regulatory Signal Finding

Goal: Detect Transcription Factor Binding Sites.

Eleazar Eskin: Columbia Univ.
How can we find TF binding sites?
How to detect signals in DNA?

- Biologists in past have created some experimental data – few examples
  - Generalise from these

- Indirect evidence of being co-regulated
  - Search for common signals

- New techniques (lab)
  - Identify regions in which binding occurs (ChIP chip)
  - SELEX
Position weight matrices (PWM, PSSM,...)

Examples

ACGTGA
ACGATG
AGGTGG
ACGAGG
TCGTGA
ACGAGG
ACGAGA
TCGTGA

Counts

A: 6 0 0 4 0 4
C: 0 7 0 0 0 0
G: 0 1 8 0 7 4
T: 2 0 0 4 1 0

“Logo”

PWM:

p/f log p/f
Motif matching

- Find all occurrences of the given motif(s)
- Databases of biologically valid motifs
- We’ll touch it a bit later
Motif discovery

- Hypothesis – a (sub)set of sequences may share a common signal.
Common biological role

- Genes known to have related roles and hence “needed” at the same time
  - e.g. same Gene Ontology class

- Measurements by microarrays
  - genes coordinately expressed should have common regulators (and signals)
Microarrays

■ Measure gene expression activity
  – genes – mRNA
  – tiling – anywhere in the genome

■ Measure in vitro TF binding
  – ChIP-chip

■ Methylation etc features of DNA
How to know what’s in the cells?

Cells and mRNA’s
How to know what’s in the cells?

Cells and mRNA’s
Microarray, the “measurement device”
Microarray, after hybridisation
Microarray, 2 colors mixed
TIGR 32k Human Arrays
Affymetrix Wafer and Chip Format

49 - 400 chips/wafer

up to ~ 1.3 million features/chip

one oligonucleotide sequence per “pixel”

20 - 50 µm

1.0 cm
From microarray images to gene expression data

Raw data
Array scans

Intermediate data
Image quantifications

Final data
Samples

Gene expression levels
Golub et al, Science Oct 15th 1999

- 38 samples of acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL)
- 6817 genes
- classificator built based on 50 best correlated genes
- tested on 34 new samples, 29 of them predicted accurately
Cluster of co-expressed genes, pattern discovery in regulatory regions

Expression profiles

600 basepairs

Upstream regions

Retrieve

Find patterns over-represented within cluster

Genome Research 1998;
ISMB (Intelligent Systems in Mol. Biol.) 2000
Binomial or hypergeometric distribution distribution tail

Background - ALL upstream sequences

\[ \pi \text{ occurs 3 times} \]

Cluster:

\[ P(3, 6, 0.2) \text{ is probability of having } \geq 3 \text{ matches in 6 sequences} \]

\[ P(\pi, 3, 6, 0.2) = 0.0989 \]

5 out of 25, \( p = 0.2 \)
ChIP-chip (or sequencing)
ChIP-chip (or sequencing)
ChIP-chip (or sequencing)
ChIP-chip (or sequencing)

I

II

III

IV

Microarray or sequencing
Clustering and Gene set enrichment

- Analysis of (any) HT data (cluster, visualise, test of significance, ...)

- Produces gene lists
  - partitioning produces “bags” or sets
  - sorting produces ranked lists

- How to interpret these results?
- What to do next?
K-means: $k = 200$ vs $50$
Genes in this cluster: 13

Sample pattern:

Some links:
- All genes in Heatmapper
- All genes in GOSE
- All genes in URLMAP

All clusters

<table>
<thead>
<tr>
<th>Gene ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1410713_AT</td>
<td>PCGD1</td>
</tr>
<tr>
<td>1434773_A_AT</td>
<td>SLC2A1</td>
</tr>
<tr>
<td>1446559_AT</td>
<td>MSX2</td>
</tr>
<tr>
<td>1449031_AT</td>
<td>CITED1</td>
</tr>
<tr>
<td>1449369_AT</td>
<td>TMPRSS2</td>
</tr>
<tr>
<td>1417145_AT</td>
<td>NFIL1</td>
</tr>
<tr>
<td>1451687_A_AT</td>
<td>TCF2</td>
</tr>
<tr>
<td>1451628_X_AT</td>
<td>NP_001035170.1</td>
</tr>
<tr>
<td>1458232_AT</td>
<td>DKK1</td>
</tr>
<tr>
<td>1435053_S_AT</td>
<td>PLEKH1</td>
</tr>
<tr>
<td>1421524_AT</td>
<td>CFH1</td>
</tr>
<tr>
<td>1426600_AT</td>
<td>SLC2A1</td>
</tr>
<tr>
<td>1439341_AT</td>
<td>AK220484</td>
</tr>
</tbody>
</table>
Genes in the dataset: 8402
Genes clustered on this page (correlation with patterns more than 0.8): 6990
Genes not clustered on this page (correlation with patterns less than 0.8): 1412
Genes genes not clustered up or down (correlation with patterns less than 0.9): 983

<table>
<thead>
<tr>
<th>Start of rise</th>
<th>1 day</th>
<th>2 days</th>
<th>3 days</th>
<th>4 days</th>
<th>5 days</th>
<th>6 days</th>
<th>7 days</th>
<th>8 day</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 0 (2622)</td>
<td>44</td>
<td>167</td>
<td>254</td>
<td>1590</td>
<td>1782</td>
<td>1205</td>
<td>1224</td>
<td>461</td>
</tr>
<tr>
<td>day 1 (637)</td>
<td>5</td>
<td>180</td>
<td>304</td>
<td>265</td>
<td>96</td>
<td>101</td>
<td>79</td>
<td>13</td>
</tr>
<tr>
<td>day 2 (238)</td>
<td>5</td>
<td>12</td>
<td>9</td>
<td>5</td>
<td>25</td>
<td>29</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>day 3 (815)</td>
<td>11</td>
<td>4</td>
<td>14</td>
<td>21</td>
<td>38</td>
<td>61</td>
<td>115</td>
<td>601</td>
</tr>
<tr>
<td>day 4 (1856)</td>
<td>4</td>
<td>14</td>
<td>13</td>
<td>98</td>
<td>229</td>
<td>453</td>
<td>1572</td>
<td></td>
</tr>
<tr>
<td>day 5 (2129)</td>
<td>14</td>
<td>2</td>
<td>110</td>
<td>261</td>
<td>572</td>
<td>1745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>day 6 (977)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NONE</td>
</tr>
<tr>
<td>Day</td>
<td>Up/Down</td>
<td>Correlation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Genes that correlate with all provided up or down patterns were smaller than 0.8 (552)
Clustering – observe your first “patterns”
<table>
<thead>
<tr>
<th>Affymetrix_id</th>
<th>GB_LIST</th>
<th>Entrez Gene</th>
<th>MGI_id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1418589_at</td>
<td>NM_010801</td>
<td>17849</td>
<td>MGI:1341919</td>
<td>Mlf1</td>
</tr>
<tr>
<td>1420376_at</td>
<td>NM_019942</td>
<td>56525</td>
<td>MGI:1888939</td>
<td>Sept6</td>
</tr>
<tr>
<td>1419440_at</td>
<td>NM_134096</td>
<td>100614</td>
<td></td>
<td>AWO49604</td>
</tr>
<tr>
<td>1423579_at</td>
<td>HM_203944</td>
<td>71409</td>
<td>MGI:191859</td>
<td>Synl2</td>
</tr>
<tr>
<td>1440200_at</td>
<td>NM_022047</td>
<td>65254</td>
<td>MGI:1929772</td>
<td>Dpsl5</td>
</tr>
<tr>
<td>1441245_at</td>
<td>NM_023047</td>
<td>65254</td>
<td>MGI:1929772</td>
<td>Dpsl5</td>
</tr>
<tr>
<td>1421341_at</td>
<td>NM_015732</td>
<td>12006</td>
<td>MGI:1270852</td>
<td>Akin2</td>
</tr>
<tr>
<td>1434699_at</td>
<td>NM_008158</td>
<td>14761</td>
<td>Gpr27</td>
<td>G protein-coupled receptor 27 (gpr27), mRN</td>
</tr>
<tr>
<td>1444054_at</td>
<td>NM_021567</td>
<td>59092</td>
<td>MGI:1890471</td>
<td>Pcbp4</td>
</tr>
<tr>
<td>1444055_at</td>
<td>NM_021567</td>
<td>59092</td>
<td>MGI:1890471</td>
<td>Pcbp4</td>
</tr>
<tr>
<td>1438255_at</td>
<td>NM_009886</td>
<td>12614</td>
<td>MGI:1109833</td>
<td>Celsr1</td>
</tr>
<tr>
<td>1423470_at</td>
<td>NM_019550</td>
<td>56195</td>
<td>MGI:1806489</td>
<td>Plbp2</td>
</tr>
<tr>
<td>1445503_at</td>
<td>NM_205232</td>
<td>279618</td>
<td></td>
<td>PREDICTED: similar to PROC149 protein [Mus musculus], mRNA</td>
</tr>
<tr>
<td>1423448_at</td>
<td>NM_008449</td>
<td>16574</td>
<td>MGI:1098269</td>
<td>Klf5c</td>
</tr>
<tr>
<td>1455266_at</td>
<td>NM_008449</td>
<td>16574</td>
<td>MGI:1098269</td>
<td>Klf5c</td>
</tr>
<tr>
<td>1436512_at</td>
<td>NM_177305</td>
<td>320902</td>
<td>Arl7</td>
<td>ADP-ribosylation factor-like 7</td>
</tr>
<tr>
<td>1417377_at</td>
<td>NM_001025600</td>
<td>54728</td>
<td>MGI:1899272</td>
<td>Igf4a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Igf4a</td>
</tr>
<tr>
<td>1419700_at</td>
<td>NM_008935</td>
<td>19126</td>
<td>MGI:1100886</td>
<td>Prom1</td>
</tr>
<tr>
<td>1422790_at</td>
<td>NM_001004357</td>
<td>66797</td>
<td>MGI:1914074</td>
<td>Cntnap2</td>
</tr>
<tr>
<td>145352_at</td>
<td>NM_035099</td>
<td>93840</td>
<td>MGI:2132272</td>
<td>Lgip</td>
</tr>
<tr>
<td>1422243_at</td>
<td>NM_021509</td>
<td>59012</td>
<td>MGI:1921582</td>
<td>Mxod1</td>
</tr>
<tr>
<td>1453524_at</td>
<td>NM_172775</td>
<td>235611</td>
<td>MGI:2154238</td>
<td>Pkm1b</td>
</tr>
<tr>
<td>1429299_at</td>
<td>NM_207203</td>
<td>73072</td>
<td>LOC37027</td>
<td>LOC37027</td>
</tr>
<tr>
<td>1423157_at</td>
<td>NM_031498</td>
<td>14365</td>
<td>MGI:108476</td>
<td>Fzd3</td>
</tr>
<tr>
<td>1423310_at</td>
<td>NM_011627</td>
<td>21983</td>
<td>MGI:134264</td>
<td>Tphg</td>
</tr>
<tr>
<td>1417010_at</td>
<td>NM_001012330</td>
<td>30928</td>
<td>MGI:1353609</td>
<td>Zip238</td>
</tr>
<tr>
<td>1434670_at</td>
<td>NM_008447</td>
<td>18672</td>
<td>Klf5a</td>
<td>kinesin family member 5A</td>
</tr>
<tr>
<td>1425494_at</td>
<td>NM_027881</td>
<td>71270</td>
<td>MGI:1918970</td>
<td>Osbp3</td>
</tr>
<tr>
<td>1421705_at</td>
<td>NM_021543</td>
<td>18530</td>
<td>MGI:1305900</td>
<td>Pcdh8</td>
</tr>
<tr>
<td>1447285_at</td>
<td>NM_021543</td>
<td>18530</td>
<td>MGI:1305900</td>
<td>Pcdh8</td>
</tr>
<tr>
<td>1434709_at</td>
<td>NM_176930</td>
<td>319504</td>
<td>MGI:2442157</td>
<td>C1700706707Rik</td>
</tr>
<tr>
<td>1435600_at</td>
<td>NM_172913</td>
<td>244579</td>
<td>MGI:3039939</td>
<td>Tnrc9</td>
</tr>
<tr>
<td>1442630_at</td>
<td>NM_145515</td>
<td>226778</td>
<td>MGI:2664902</td>
<td>Mark1</td>
</tr>
<tr>
<td>1418564_at</td>
<td>NM_007563</td>
<td>12183</td>
<td>MGI:1098262</td>
<td>Bpum</td>
</tr>
<tr>
<td>1449885_at</td>
<td>NM_138751</td>
<td>192215</td>
<td>MGI:2177570</td>
<td>Tmem47</td>
</tr>
<tr>
<td>1437252_at</td>
<td>NM_030719</td>
<td>80909</td>
<td>MGI:1933384</td>
<td>Gats5</td>
</tr>
<tr>
<td>1450108_at</td>
<td>NM_008440</td>
<td>16560</td>
<td>MGI:1083491</td>
<td>Klf1a</td>
</tr>
<tr>
<td>1432334_at</td>
<td>NM_018769</td>
<td>54722</td>
<td>MGI:106003</td>
<td>Dnase5h</td>
</tr>
<tr>
<td>1453409_at</td>
<td>NM_170832</td>
<td>68166</td>
<td>MGI:1915410</td>
<td>Sipre1</td>
</tr>
<tr>
<td>1451191_at</td>
<td>NM_007759</td>
<td>12904</td>
<td>MGI:88491</td>
<td>Crabp2</td>
</tr>
</tbody>
</table>
Find a common function(pattern)

- Experiment or analysis identifies a set of genes

- What is a common theme to these genes?

- Biological function: Gene Ontology; molecular pathway; shared regulatory motif or miRNA target site
Previously known functions:
Your “query”
g:Profiler—a web-based toolset for functional profiling of gene lists from large-scale experiments

Jüri Reimand¹, Meelis Kull¹,²,³, Hedi Peterson²,³, Jaanus Hansen¹ and Jaak Vilo¹,²,³,*
GO Evidence Codes

From reviews or introductions:

- TAS - Traceable Author Statement
- NAS - Non-traceable Author Statement
- IC - Inferred by Curator
- ISS - Inferred from Sequence or structural Similarity
- IEA - Inferred from Electronic Annotation
- ND - Not Determined

From primary literature:

- IDA - Inferred from Direct Assay
- IMP - Inferred from Mutant Phenotype
- IGI - Inferred from Genetic Interaction
- IPI - Inferred from Physical Interaction
- IEP - Inferred from Expression Pattern

automated
KEGG: Biosynthesis of...
p-value

- tail of the hypergeometric distribution

- Multiple testing
  - multiple sets to compare against
  - different sizes of queries
  - different sizes (and nrs) of reference sets
SCS - Set Counts and Sizes threshold
Motif discovery in sequences

- Deterministic and probabilistic
- Pattern driven vs sequence driven
- Descriptive or Discriminative
Cluster of co-expressed genes, pattern discovery in regulatory regions

Expression profiles

Retrieves

600 basepairs

Upstream regions

Find patterns over-represented within cluster

*Genome Research* 1998;
*ISMB (Intelligent Systems in Mol. Biol.)* 2000
Pattern vs cluster “strength”

The pattern probability vs. the average silhouette for the cluster

The same for randomised clusters

Vilo et al., ISMB 2000
Suffix tree – represent all suffixes

CATAT => suffix tree

O(n) time and space
SPEXS - Sequence Pattern Exhaustive Search
Jaak Vilo, 1998, 2002

- **User-definable pattern language**: substrings, character groups, wildcards, flexible wildcards (c.f. PROSITE)
- Fast exhaustive search over pattern language
- “Lazy suffix tree construction”-like algorithm (Kurtz, Giegerich)
- **Analyze multiple sets of sequences simultaneously**
- Restrict search to most frequent patterns only (in each set)
- **Report** most frequent patterns, patterns over- or underrepresented in selected subsets, or patterns significant by various statistical criteria, e.g. by binomial distribution
Algorithm 3.1 Frequent substring generation

Input: String $S$ of length $n$, minimum number $K$ for occurrences of substrings
Output: Suffix trie containing substrings that occur at least $K$ times in string $S$

Method:

1. $\text{Root} \leftarrow \text{new node}$; $\text{Root}.\text{char} \leftarrow \lambda$
2. $\text{Root}.\text{pos} \leftarrow (1, 2, \ldots, |S|)$
3. $\text{enqueue}(Q, \text{Root})$
4. while $N \leftarrow \text{dequeue}(Q)$
5. // Group the positions according to characters in $S$
6. $\text{foreach}$ character $c \in \Sigma$
7. $\text{Set}(c) \leftarrow \emptyset$
8. $\text{foreach} p \in N.\text{pos}$
9. $\text{add} p + 1$ to $\text{Set}(S[p])$
10. // Insert new child nodes for substrings that are sufficiently frequent
11. $\text{foreach}$ character $c \in \Sigma$ such that $|\text{Set}(c)| \geq K$
12. $N.\text{child}(c) \leftarrow \text{new node } P$ with label $P.\text{char} = c$
13. $P.\text{pos} \leftarrow \text{Set}(c)$
14. $\text{enqueue}(Q, P)$
15. delete $N.\text{pos}$
16. return $\text{Root}$

The tree is constructed by systematically extending the leaf nodes. Thus, the position lists are needed only for the leaves during the tree construction,
Algorithm 4.1  Generation of frequent patterns with character group positions

Input: String $S$ of length $n$, character groups $\Gamma$, threshold $K$
Output: Pattern trie with nodes defining patterns over $\Sigma \cup \Gamma$ that occur at least $K$ times in string $S$

Method:

1. $Root \leftarrow$ new node; $Root.char \leftarrow \lambda$
2. $Root.pos \leftarrow (1, 2, \ldots, n)$
3. $enqueue(Q, Root)$
4. while $N \leftarrow dequeue(Q)$
5.  // Construct the position list for pattern defined by node $N$
6.   if $N.char \in \Sigma$ then $Pos \leftarrow N.pos$
7.   else $Pos \leftarrow \bigcup_{c \in N.char} N.sibling(c).pos$
8.  // Group the positions according to characters in string $S$
9.  foreach $c \in \Sigma$
10.   $Set(c) \leftarrow \emptyset$
11.  foreach $p \in Pos$
12.   add $p + 1$ to $Set(S[p])$
13.  foreach character $c \in \Sigma$
14.    if ($|Set(c)| \geq K$) or ($\exists g \in \Gamma$ s.t. $c \in g$ and $\Sigma_{f \in g}|Set(f)| \geq K$)
15.      then
16.         $N.child(c) \leftarrow$ new node $P$; $P.char \leftarrow c$
17.         $P.pos \leftarrow Set(c)$
18.         if ($|Set(c)| \geq K$) $enqueue(Q, P)$
19.      endif
20.  foreach character $g \in \Gamma$
21.    if $\Sigma_{c \in g}|Set(c)| \geq K$ then
22.      if $\exists f \in (\Gamma \cup \Sigma)$, $f \subset g$, $\Sigma_{c \in f}|Set(c)| = \Sigma_{c \in g}|Set(c)|$
23.        then // More specific pattern covers already the same positions
24.        else
25.            $N.child(g) \leftarrow$ new node $P$; $P.char \leftarrow g$
26.            $enqueue(Q, P)$
27.        endif
28.    endif
29.    else
30.      if all nodes $N.sibling(c)$, $c \in \Sigma \cup \Gamma$ have been expanded
31.        delete all $N.sibling(c).pos$, where $c \in \Sigma$
32.    endif
33.  endif
Algorithm 5.1 Frequent subsequence generation

Input: Input strings $S^m = \{S_1, \ldots, S_m\}$, threshold $K$

Output: Pattern trie with nodes defining patterns over $\Sigma \cup \{\ast \times \Sigma\}$ that occur in at least $K$ different strings from $S^m$

Method:

1. $S \leftarrow S_1\# \ldots \#S_m\#$
2. Root $\leftarrow$ new node with empty label $Root$.char
3. Root.pos $\leftarrow (1, 2, \ldots, |S|)$
4. enqueue(Q, Root)
5. while $N \leftarrow$ dequeue(Q)
6. // Group the positions to separate sets by actual characters in string
7. foreach $p \in N$.pos
8. add $p + 1$ to $Set(S[p])$
9. // Create position lists for '*' extensions.
10. $w \leftarrow 0$
11. while $S[p + w] \neq \#'\#'$ // Until the delimiter
12. add $p + w + 1$ to $Set('S[p+w]')$
13. $w \leftarrow w + 1$
14. // Construct the nodes defining the frequent patterns
15. foreach character $c \in (\Sigma \cup \ast \Sigma)$ where $|setoffstrings(Set(c))| \geq K$
16. $N$.child(c) $\leftarrow$ new node $P$ with $P$.char $\leftarrow c$
17. $P$.pos $\leftarrow Set(c)$
18. delete $Set(c)$
19. delete $N$.pos

For practical reasons the algorithm can be implemented with the restriction that the wildcards have some maximal width $w \leq W$, and there is only
Algorithm 6.1  Generating the patterns that occur frequently in the set of input strings

Input:  Strings $S^m = \{S_1, \ldots, S_m\}$, pattern class, threshold $K$

Output: Patterns from pattern class in the order of frequency.

Method:
1. $S \leftarrow \#S_1\# \ldots \#S_m\#$
2. Root $\leftarrow$ new node; Root.char $\leftarrow \lambda$
3. Root.pos $\leftarrow (1, 2, \ldots, |S|)$
4. Enqueue(Q, Root, size(Root.pos)) // Priority queue based on the set size
5. Size $= size($Root.pos$)$
6. while $N \leftarrow$ dequeue(Q)
7.    if $size(N.pos) < Size$
8.       // Print best patterns occurring Size times (or in Size strings in S
9.       while ($PN = dequeue(PQ)$) output pattern defined by $PN$
10.      Size $= size(N.pos)$
11.     Enqueue(PQ, N, cost(N))
12.    // Generate all possible extensions $P$ to pattern label($N$) from position list $N.pos$
13.     foreach extension $P$ of $N$
14.        if pattern $P$ and position list $P.pos$ fulfill the criteria
15.            then
16.                if Exists node $M$ s.t. position list $P.pos$ is equal to $M.pos$
17.                    then join the subtrees $M$ and $P$
18.                else $N.sibling \leftarrow P$ and Enqueue(Q, P, size(P.pos))
19.            // Print all the remaining patterns
20.        while($PN = dequeue(PQ)$) output pattern defined by $PN$

The function $size(N(\alpha).pos)$ used in Algorithm 6.1 can be either the number of positions where pattern $\alpha$ occurs, or the number of different strings from $S^m$ matched by $\alpha$. The order of frequency can hence be based either on the number of occurrences or the number of different strings that are matched by the pattern.
Sequence patterns: the basis of the SPEXS

![Diagram of sequence patterns: GCAT, GCATA, GCATA, GCATA.C]

- GCAT (4 positions)
- GCATA (3 positions)
- GCATA
- GCATA.C
Implementation example

<table>
<thead>
<tr>
<th>Input 1</th>
<th>Input 2</th>
<th>P – pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACGTG</td>
<td>AGTAC</td>
<td>e.g. AC</td>
</tr>
<tr>
<td>CACGA</td>
<td>ATGAA</td>
<td>AC.pos = 2, 9, 23</td>
</tr>
<tr>
<td>TATCG</td>
<td>GCAGG</td>
<td>ACG.pos = 3, 10</td>
</tr>
</tbody>
</table>

Convert into internal representation:

............. | ............. | ............. | ............. | ............. = 36
ACGTG#CACGA#TATCG#AGTAC#ATGAA#GCAGG#
111111-111111-111111-222222-222222-222222
Algorithm 3.19  The SPEXS algorithm

Input: String \( S \), pattern class \( P \), output criteria, search order, and fitness measure \( F \)

Output: Patterns \( \pi \) from pattern class \( P \) fulfilling all the criteria, and output in the order of fitness \( F \)

Method:
1. Convert input sequences into a single sequence, initiate the data structures
2. \( \text{Root} \leftarrow \text{new node} \)
3. \( \text{Root}.\text{label} = \lambda \)
4. \( \text{Root}.\text{pos} \leftarrow (1, 2, \ldots, n) \)  // Assume empty pattern to match everywhere
5. \( \text{enqueue}(Q, \text{Root}, \text{order}) \)
6. while \( N \leftarrow \text{dequeue}(Q) \)
7. Create all possible extensions \( P \in P \) of \( N \) using \( N.\text{pos} \) and \( S \)
8. foreach extension \( P \) of \( N \)
9. if pattern \( P \) and position list \( P.\text{pos} \) fulfill the criteria
10. then
11. \( N.\text{child} \leftarrow P \)
12. calculate \( F(P, S) \)
13. \( \text{enqueue}(Q, P, \text{order}) \)  // Insert to \( Q \) for further extensions
14. if \( P \) fulfills the output criteria store \( P \) into output queue \( O \)
15. end
16. Report the list of top-ranking patterns from output queue \( O \)
SPEXS – general algorithm

1. $S =$ input sequences ( $|S| = n$ )
2. $\lambda =$ empty pattern, $\lambda.\text{pos} = \{1,\ldots,n\}$
3. enqueue( order , $\lambda$, priority)

4. while $p =$ dequeue( order )
5. generate all allowed extensions (p’, p’.pos) of p
6. enqueue( output, p’, fitness(p’))
7. enqueue( order, p’, priority(p’))

8. while $p =$ dequeue( output )
9. Output p

Jaak Vilo: Discovering Frequent Patterns from Strings.

Jaak Vilo: Pattern Discovery from Biosequences
Order

Breadth-first

1
2
3
4
5
6
7
8
9
10

Depth-first

1
4
3
2
7
6
5
10
9
8
Order

Frequent-first

50

40

34

24

6 4

6 4
SPEXS: count and memorize

\[\text{i...v....x.....v.....x} \]
\[\text{abracadabra} \]
\[\{1, 4, 6, 8, 11, 13, 15, 18, 20\} \]
\[\{2, 5, 7, 9, 12, 14, 16, 19, 21\} \]
SPEXS: extend ...
SPEXS: find frequent first

i...v....x.....v....x
abracadabra
dadadada

{2,9,16}
{7,12,14}
{2,5,7,9,12,14,16,19,21}
SPEXS: group positions

i...v.....x.....v.....x

abracadabradadabrasca

{2,9,16} {7,12,14} [bd] {2,7,9,12,14,16,19,21}
The wildcards

GCAT.*X
The wildcards

```
GCAT.*A
```
The wildcards

GCAT\.{3,6}X
The wildcards: not too many
Multiple data sets

D1

4/3 (6)

D2

3/3 (12)

D3

2/2 (9)
GPCR coupling

Current perspective

Signal: Agonist

GPCR: intracellular messengers

G-protein

Effector enzymes

Intracellular messengers
Our Computational Approach

- Membrane topology 7TMHMM
- Intracellular domains of \( \approx 100 \) receptor sequences with
- well-characterised, and non-promiscuous coupling
  (split into \( G_s \), \( G_{i/o} \) and \( G_{q/11} \))

Steffen Möller, Jaak Vilo, Michael D.R. Croning

Prediction of the coupling specificity of G protein coupled receptors to their G proteins.
Steffen Möller, Jaak Vilo, Michael D.R. Croning
Prediction of the coupling specificity of G protein coupled receptors to their G proteins.
Receptor Match Positions

Möller, Vilo, Croning, *ISMB 2001*
Improving upon discrete patterns
101 Sequences relative to ORF start

YGR128C + 100

GATGAG.T
1:52/70 2:453/508  R:7.52345  BP:1.02391e-33
G.GATGAG.T
AAAATTTT
1:63/77 2:833/911  R:4.95687  BP:5.02807e-32
TGAAAA.TTT
1:45/53 2:333/350  R:8.85687  BP:1.69905e-31
TG..AA..TTT
TG..AAA..TTTT
1:40/43 2:254/26(1)  R:4.95687  BP:5.02807e-32
TGAAA...TTT
1:54/65 2:608/64(1)  R:7.45961  BP:1.02332e-31

...
.G.GATGAG.T. 39 seq

39 seq (vs 193)
p = 2.5e-33
-1: .G.GATGAG.T. 61 seq (vs 1292)
p = 1.4e-19
-3: .G.GATGAG.T. 98 seq
<table>
<thead>
<tr>
<th>Pattern</th>
<th>In cluster</th>
<th>Total nr</th>
<th>Ratio</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.GATGAG.T</td>
<td>39</td>
<td>193</td>
<td>13.24</td>
<td>2.490e-33</td>
</tr>
<tr>
<td>TG.AAA.TTT</td>
<td>53</td>
<td>538</td>
<td>6.46</td>
<td>3.248e-31</td>
</tr>
<tr>
<td>TGAAAAA.TTT</td>
<td>45</td>
<td>333</td>
<td>8.86</td>
<td>1.699e-31</td>
</tr>
<tr>
<td>-1:G.GATGAG.T</td>
<td>61</td>
<td>1295</td>
<td>3.09</td>
<td>1.441e-19</td>
</tr>
<tr>
<td>-1:TG.AAA.TTT</td>
<td>89</td>
<td>3836</td>
<td>1.52</td>
<td>6.126e-12</td>
</tr>
<tr>
<td>-1:TGAAAAA.TTT</td>
<td>76</td>
<td>2190</td>
<td>2.27</td>
<td>1.654e-18</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>395</td>
<td>10.29</td>
<td>6.909e-50</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>1227</td>
<td>4.43</td>
<td>1.703e-44</td>
</tr>
<tr>
<td></td>
<td>69</td>
<td>593</td>
<td>7.63</td>
<td>1.585e-48</td>
</tr>
</tbody>
</table>

*Jaak Vilo: Pattern Discovery from Biosequences*
PhD Thesis, Department of Computer Science, University of Helsinki, Finland
These hits result in a PWM:
PWM based on all previous hits, here shown highest-scoring occurrences in blue.
All against all approximate matching

For every subsequence of every sequence

Match approximately against all the sequences.

Approximate hits define PWM matrices (not all positions vary equally).

Look for ALL PWM-s derived from data that are enriched in data set (vs. background).

Hendrik Nigul, Jaak Vilo
Dynamic programming

- Small nr of edit operations allows to limit the search efficiently around main diagonal
Suffix Tree

{1:24, 2:12, 2:23…}
Trie based all against all approximate matching

- trieindex
- trieagrep
- trieallagrep
- triematrix
More directions for PD
Multiple alignment
Artificial setup
Challenge problem


- Plant into every sequence a string $X$ of length $l$, with $d$ characters randomly altered.

- What was the original $X$?

- $(l,d)$-problem
(4, 1) - problem

“Seed” -

ACTG
CCTG
GCTG
TCTG
AATG
AGTG
ATTG

12 possible planted versions
Graph constructed by WINNOWER

For (15,4)-signal - connect all words with distance at most 8:

atgaccggat act gat AgAAgAAAGG t GGG at aat ggagt acgat aa

at gact tc AAt AAAAAc GGc GGG gct ct cccgat t t t gagt at ccct ggg

gcaat cgccgaacc aagct gagaat t ggat gt cAAAAAt AAt GGAGt GGC ac

gt caat cg gaaaaa ac ggt ggaggat t tc AAAAAAGGGG t Ggaccgct t

real signals
spurious signals

signal edges
spurious edges

from: Eleazar Eskin
Pairs of motifs
Composite Patterns

Conserved Region  \[ \text{Unconserved Spacing} \]  Conserved Region

Co-occurring patterns
- (GuhaThakurta, Stormo 2001)
- Fixed Order (Dyad Problem)
  - (van Helden et. al 2000)
  - (Gelfand et. al 2000)
  - (Marsan, Sagot 2000)

from: Eleazar Eskin
Patterns with Mismatches

AAAAAAAGGGGGGGG- (10, 15) - CTGATTCCAATACAG

Mismatches \( d=8 \)

Instances:
AcAAAAcAGGGGt GG- 11- CTGAcTCTt ATAAaAG
AAAcAAAgatGGt G- 12- CTGcgtTCTAAtt cAG
AtAAAAatcGGGc GG- 10- CTGATcCTAt TACcG
AAAAAtAAGGGGc GG- 14- CgGAcTCTAAtgCAG

Eleazar Eskin
Sample Sequences

AAAAAAAGGGGGGGG- (10, 15) - CTGATTCCAATACAG

act gat AAAAAAAAGGGGGGGGggcgt acacat tagCTGATTCCAATACAGacgt
aa AAAAAAAAAGGGGGGGGaaacttttccgaat aCTGATTCCAATACAGcagt
at gactt AAAAAAAAAGGGGGGGGtgcctc cccgat tttcCTGATTCCAATACAGc
agg AAAAAAAAAGGGGGGGGagccct aacggact t aatCCTGATTCCAATACAGta
ggagg AAAAAAAAAGGGGGGGGagccct aacggact t aatCCTGATTCCAATACAG

Eleazar Eskin
Sample Sequences

AAAAAAAAGGGGGGGG- (10, 15) - CTGATTCCAATACAG

act gat AAAAt AAAGc GGGa Gggcgt acacat tagCaGAc TCCAATT gAGacgt
aa AAt AAAAAaaa GGcG aacattttccgaata CTGAc TCCAAagCAG gat cagt
at gactt AACAAAt AgGGCa GGGt gct ct ccgccgattttc CTGcTa CCAAgAt AGc
agg AAt AAAAAa GGaGGGG gcccct aacggacttaat CCaGATTgCAc TAaAat a
ggagg AAaAAAAaAGGa GGGagcccct aacggacttaat Ct TGAaTCCt ATACAc

Eleazar Eskin
Traditional Approach: Weaknesses

Traditional Approach: Find each conserved region separately.

Problem: Each region too “weak”.

Eleazar Eskin
Traditional Approach: Solution

Traditional Approach: Find each conserved region separately.

Problem: Each region too “weak”.

Our approach: Find both regions simultaneously.

Conserved Region \[\rightarrow\] Unconserved Spacing \[\rightarrow\] Conserved Region

single pattern after preprocessing.

Eleazar Eskin
Combinations and modules

- Regulatory signals do not work alone
- Motif co-occurrences
Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the *Drosophila* genome

Benjamin P. Berman*, Yutaka Nibu*, Barret D. Pfeiffer†, Pavel Tomancak*‡, Susan E. Celniker†§, Michael Levine*, Gerald M. Rubin*†‡, and Michael B. Eisen*§¶

- 700bp widows with at least 13 binding site occurrences
(A) High stringency matches
700bp widows with at least 13 binding site occurrences

(B) High stringency matches and clustering filter

(C) Expanded view of even-skipped region
Using multiple species

- Phylogenetic footprinting
- Phylogenetic shadowing
- Conservation cross species
Evolutionary relationship among organisms based on similarity of the primary sequences of their CYTOCHROME c proteins.
Phylogenetic footprinting


Men and mice are alike
The kangaroo genome

Matthew J. Wakefield & Jennifer A. Marshall Graves

Phylogenetic footprinting of the 3' untranslated region of the SLC16A2 (X2R) gene from human, mouse and tammar wallaby. Human, mouse and tammar wallaby were compared with VISTA (Dubchak et al., 2000). The human/mouse comparison indicates sequence similarity across the entire untranslated region, masking any localized regions of high functional constraint. In tammar/human and tammar/mouse comparisons the background level of conservation is lower, permitting the observation of localized regions of functional constraint. Similar results were obtained with the alternative phylogenetic footprinting program PipMaker2 (data not shown). bp, base pairs.
Phylogenetic footprinting

Study the same gene in many species

If preserved during evolution then must be important for something!!!
What if species too similar?

- Almost entire genome is highly similar
- Signal gets “lost”
Phylogenetic shadowing

- Use many closely related species (monkeys, apes, ...)

- All regions that differ, are shadowed out

- These regions that do not have differences in (almost) any, are probably important

Phylogenetic shadowing

![Phylogenetic shadowing diagram]

http://chr21.molgen.mpg.de/images/projects/BACH1_small.jpg
Alignment of functional elements


Locating potential enhancer elements by comparative genomics using the EEL software

Kimmo Palin¹, Jussi Taipale² & Esko Ukkonen³

¹Department of Computer Science, P.O. Box 68 (Gustaf Hällströminkatu 2b) FIN-00014, University of Helsinki, Finland. ²Molecular/Cancer Biology Program, Institute of Biomedicine, University of Helsinki and Department of Molecular Medicine, National Public Health Institute (KTL), Biomedicum, P.O. Box 63 (Haartmaninkatu 8), FIN-00014, University of Helsinki, Finland. ³Helsinki Institute for Information Technology, P.O. Box 68, FIN-00014, University of Helsinki, Finland. Correspondence should be addressed to K.P. (Kimmo.Palin@helsinki.fi).

Published online 27 June 2006; doi:10.1038/nprot2006.56

\[
\text{Score} = \lambda \Delta G - \mu \overline{X} - \frac{\nu \Delta X^2 - \xi \phi^2}{2 \overline{X}}
\]

Figure 1 | The scoring function of EEL. Binding sites with affinity \( \Delta G \) are conserved in human and mouse sequences. The average distance to adjacent
Problem

- Target vs background data?
- Strong vs weak
- No clear cut
Discovering Motifs in Ranked Lists of DNA Sequences

Eran Eden¹*, Doron Lipson¹, Sivan Yogev¹,², Zohar Yakhini¹,³*
¹ Computer Science Department, Technion, Haifa, Israel, ² IBM Research Laboratories, Haifa, Israel ³ Agilent Laboratories, Santa Clara, California, United States of America

- (c1) the cutoff used to partition data into a target set and background set of sequences is often chosen arbitrarily;
- (c2) lack of an exact statistical score and p-value for motif enrichment
- (c3) a need for an appropriate framework that accounts for multiple motif occurrences in a single promoter.
- (c4) motif discovery methods tend to report presumably significant motifs even when applied on randomly generated data. These motifs are clear cases of false positives and should be avoided.
DRIM receives a list of DNA sequences as input and a criterion by which the sequences should be ranked, for example, TF binding signals as measured by ChIP-chip.

(i) The sequences are ranked according to the criterion.

(ii) A “blind search” is performed over all the motifs that reside in the restricted motif space (in this study, the restricted motif space contains \( \sim 100,000 \) motifs, see Methods, The DRIM software). For each motif an occurrence vector is generated. Each position in the vector is the number of motif occurrences in the corresponding sequence. (The figure shows the vector for the motif CACGTGW).

(iii) The motif significance is computed using the mHG scheme, and the optimal partition into target and background sets in terms of motif enrichment is identified. The promising motif seeds are passed as input to the heuristic motif search model and the rest are filtered out.

(iv,v) The motif seeds are expanded in an iterative manner (the mHG is computed in each lap), until a local optimum motif is found.

(vi) The exact mHG p-value of the motif is computed. If it has a p-value \(< 10^{-5}\), then it is predicted as a true motif (the choice of this threshold is explained in Results, Proof of principle). The output of the system is the motif representation above IUPAC, its PSSM, mHG p-value, and optimal set partition cutoff.

Figure 1. DRIM Flow Chart

doi:10.1371/journal.pcbi.0030039.g001
Figure 1. DRIM Flow Chart
\[ \text{Prob}(X \geq b) = HGT(b; N, B, n) = \sum_{i=b}^{\min(n, B)} \binom{n}{i} \frac{\binom{N-n}{B-i}}{\binom{N}{B}} \]
Figure 8. Two-Dimensional Grid Used for Calculating mHG p-Value
In this example \( N = 20, B = 10, p = 0.1 \). Light-shaded area describes all attainable values of \( n \) and \( b \). Dark-shaded area describes the subset \( R \): all values of \( n \) and \( b \) for which \( \text{HGT}(b;N,B,n) \leq p \). Two \((0,0) \rightarrow (N,B)\) paths are depicted, representing the binary label vectors \( \lambda_1 = \{1,1,1,0,1,0,1,1,1,0,1,0,0,0,0,1,0,1,0\} \) and \( \lambda_2 = \{0,0,0,1,0,1,1,0,0,0,1,1,0,0,1,0,1,1,1\} \). The path \( \lambda_1 \) traverses \( R \), demonstrating that \( mHG (\lambda_1) \leq p \). The path \( \lambda_2 \) does not traverse \( R \), demonstrating that \( mHG (\lambda_1) > p \).
Pattern languages

- Substrings
- Character groups
- Unrestricted wildcards
- Restricted wildcards
- Combine all above
- Closures

ATCGA
ATC[GC].[^A]
AT.*CG
AT.{2,5}CG
A.T[GC].{1,3}[GT]AC
TGC…………GCA
TGAAAA+TTT+

Allow mismatches, insertions, deletions
Probabilistic versions of the above
Probabilistic motifs

- Gary Stormo lab
- EM-algorithm
  - MEME (Bailey, Elkan)
- Gibbs Sampling
  - AlignAce (Roth et al)
  - (Rocke, Tompa)
- Neural networks
- HMM models, SCFG
The advantages and disadvantages of discrete patterns

Advantages
- simple and easily interpretable objects
- easier to discover from scratch (i.e., if no additional information to sequences are given), particularly in noisy data

Disadvantages
- limited descriptive power (no weights can be attributed to alternatives)
- No probability of a match
Fitness measures

- Ratio (times over-represented)
- ROC AUC
- Probability (p-value)
- Domain specific (biological) score
Multiple testing due

- Large pattern (search) space
- Many data sets analysed
- Different cut-off thresholds
Search algorithms

- Pattern driven
  - generate all possible patterns, evaluate

- Data Driven
  - e.g. align data sets, “read out patterns”
  - EM, Gibbs, ...
  - (all probabilistic methods)
Search algorithm

■ Pattern driven
  – generate all possible patterns, evaluate

■ Data Driven
  – e.g. align data sets, “read out patterns”

■ Combined
  – Use data as a guide for exhaustive search through pattern space
Regular pattern tools

- SPEXS (Jaak Vilo)
- Pratt (Inge Jonassen, U. of Bergen)
- TEIRESIAS (IBM Research, Rigoutsos, Floratos)
- MobyDick etc (Harmen Bussemaker)
- “RSA-tools” (Jacques van Helden)
- Martin Tompa
- Marsan & Sagot (suffix tree + gapped motifs)
- Jensen & Knudsen (suffix tree based substrings)
- Verbumculus (Stefano Lonardi, A. Apostolico)
- …
Motif Discovery on Promotor Sequences

submitted in partial satisfaction of the requirements for the degree Diplom-Informatiker

by

Maximilian Häußler

Universität Potsdam
Institut für Informatik
and IRISA/INRIA Rennes

Supervisors:
  Prof. Dr. Torsten Schaub
  Dr. Jacques Nicolas

13th June 2005
<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Model</th>
<th>Search</th>
<th>Scoring</th>
<th>Publication</th>
<th>Software</th>
<th>University, Country</th>
<th>Tested on</th>
</tr>
</thead>
<tbody>
<tr>
<td>by Staden</td>
<td>String</td>
<td>Enum,PD</td>
<td>?</td>
<td>(Staden, 1989)</td>
<td>None</td>
<td>IT,Bari</td>
<td>EPD</td>
</tr>
<tr>
<td>Wordup</td>
<td>String</td>
<td>Enum,PD</td>
<td>$\chi^2$</td>
<td>(Pesole et al., 1992)</td>
<td>None</td>
<td>US,NIH</td>
<td>Proteins</td>
</tr>
<tr>
<td>Gibbs Sampler</td>
<td>Matrix</td>
<td>Gibbs</td>
<td>Wilcoxon</td>
<td>(Lawrence et al., 1993)</td>
<td>Web, Src</td>
<td>US,UCSD</td>
<td>Proteins</td>
</tr>
<tr>
<td>CoResearch</td>
<td>String</td>
<td>Enum,PD</td>
<td>IC</td>
<td>(Wolpert et al., 1996)</td>
<td>Bin:SunOS</td>
<td>DE,GSF</td>
<td>Mammals</td>
</tr>
<tr>
<td>R'MES</td>
<td>Matrix</td>
<td>?</td>
<td>Markov+?</td>
<td>(Schbath, 1997)</td>
<td>Removed</td>
<td>FR,INRA</td>
<td>None</td>
</tr>
<tr>
<td>Oligo-Analysis</td>
<td>String</td>
<td>Enum</td>
<td>Sig(Genome),Pos</td>
<td>(Helden et al., 1998)</td>
<td>Web</td>
<td>BE,Brux.</td>
<td>Yeast</td>
</tr>
<tr>
<td>Teiresias</td>
<td>String</td>
<td>Convolution</td>
<td>None</td>
<td>(Rigoutsos &amp; Floratos, 1998)</td>
<td>Web,B:Lin/Win</td>
<td>US,IBM</td>
<td>Protein</td>
</tr>
<tr>
<td>Yebis</td>
<td>HMM</td>
<td>Enum</td>
<td>$\chi^2$,IC</td>
<td>(Yada et al., 1998)</td>
<td>Web</td>
<td>JUJST</td>
<td>Human (GSF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Hertz &amp; Stormo, 1999)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winnerro 3</td>
<td>String</td>
<td>Graph</td>
<td>Mismatch</td>
<td>(Pevzner &amp; Sze, 2000)</td>
<td>N.a.</td>
<td>US,UCSD</td>
<td>Bac.</td>
</tr>
<tr>
<td>SP-STAR</td>
<td>String</td>
<td>Graph</td>
<td>Mismatch</td>
<td>(Pevzner &amp; Sze, 2000)</td>
<td>N.a.</td>
<td>US,UCSD</td>
<td>Bac.</td>
</tr>
<tr>
<td>SMILE</td>
<td>String,Dyad</td>
<td>Suffix</td>
<td>$z_\chi^2$</td>
<td>(Marzan &amp; Sagot, 2000)</td>
<td>Src</td>
<td>FR,IGM</td>
<td>Bac.</td>
</tr>
</tbody>
</table>

Table 2.1.: Some discovery algorithm implementations as of oct 2004
<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Model</th>
<th>Search</th>
<th>Scoring</th>
<th>Publication</th>
<th>Software</th>
<th>University or Country</th>
<th>Tested on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbumculus</td>
<td>String</td>
<td>Suffix</td>
<td>z</td>
<td>(Apostolico et al., 2000)</td>
<td>Bi, Lin, Sol</td>
<td>US, UCR</td>
<td>Yeast</td>
</tr>
<tr>
<td>MobyDick by Anderson &amp; al.</td>
<td>String</td>
<td>Dictionary</td>
<td>z</td>
<td>(Bussomaker et al., 2006a)</td>
<td>None</td>
<td>US, Rockefeller</td>
<td>Yeast</td>
</tr>
<tr>
<td>YMF</td>
<td>String</td>
<td>Enum, PD</td>
<td>Spec</td>
<td>(Anderson &amp; Parker, 2000)</td>
<td>Perl Src</td>
<td>US, HHMI</td>
<td>Yeast</td>
</tr>
<tr>
<td></td>
<td>String</td>
<td>Suf. Enum.</td>
<td>z(Markov), Distr</td>
<td>(Sinha &amp; Tompa, 2000) (Sinha &amp; Tompa, 2002)</td>
<td>Web, (Src)</td>
<td>US, UW</td>
<td>Yeast</td>
</tr>
<tr>
<td>Bioprospector</td>
<td>Matrix, Dyad</td>
<td>Gibbs</td>
<td>IC</td>
<td>(Lu et al., 2001)</td>
<td>Web, (Src)</td>
<td>US, Stanford</td>
<td>Yeast, Bac</td>
</tr>
<tr>
<td>Co-Bind</td>
<td>Matrix, Dyad</td>
<td>Gibbs</td>
<td>IC</td>
<td>(GhumaThakurta &amp; Stormo, 2001)</td>
<td>Src</td>
<td>US, WUSTL</td>
<td>Yeast, Bac</td>
</tr>
<tr>
<td>IIB</td>
<td>String</td>
<td>Enum</td>
<td>z(Markov)</td>
<td>(Kooibas et al., 2001)</td>
<td>None</td>
<td>DE, FU-Berlin</td>
<td></td>
</tr>
<tr>
<td>Mitra</td>
<td>String, Dyad</td>
<td>Prefix/Graph</td>
<td>Back, Mism</td>
<td>(Eskin &amp; Pevzner, 2002)</td>
<td>Web</td>
<td>US, Columbia</td>
<td>Yeast, Bac</td>
</tr>
<tr>
<td>Spexs</td>
<td>String</td>
<td>Suffix</td>
<td>F(Bin)</td>
<td>(Vico, 2002)</td>
<td>Web</td>
<td>UK, EBI</td>
<td>Yeast, Protein</td>
</tr>
<tr>
<td>Mitra-FSSM</td>
<td>Matrix</td>
<td>Suffix</td>
<td>IC</td>
<td>(Eskin, 2004)</td>
<td>None</td>
<td>Israel, HU</td>
<td>Yeast</td>
</tr>
<tr>
<td>MOPAC</td>
<td>String</td>
<td>Enum, SD</td>
<td>None</td>
<td>(Ganesh et al., 2003)</td>
<td>None</td>
<td>US, Texas A&amp;M</td>
<td>Yeast</td>
</tr>
<tr>
<td>RISA</td>
<td>String</td>
<td>Graph?</td>
<td>None</td>
<td>(Dunlova et al., 2003)</td>
<td>None</td>
<td>Rus, BAS</td>
<td>Art, Yeast</td>
</tr>
<tr>
<td>Multiprofiler</td>
<td>String</td>
<td>Multiprof.</td>
<td>Mismatches</td>
<td>(Keich et al., Pevzner, 2002a)</td>
<td>None</td>
<td>US, UCSD</td>
<td>Art,</td>
</tr>
<tr>
<td>Projection</td>
<td>String</td>
<td>Hashing</td>
<td>p(Dist)</td>
<td>(Buhler &amp; Tompa, 2002)</td>
<td>Src</td>
<td>US, WUSTL</td>
<td>Yeast, Mammals</td>
</tr>
<tr>
<td>Woeder</td>
<td>String</td>
<td>Enum, PD</td>
<td>Core, IC, z, MAP</td>
<td>(Pavesi et al., 2001) (Pavesi et al., 2004b)</td>
<td>Web, Src</td>
<td>IT, Milan</td>
<td>Yeast</td>
</tr>
<tr>
<td>DMotifs</td>
<td>String</td>
<td>Enum, PD</td>
<td>TnoM</td>
<td>(Sinha, 2003)</td>
<td>None</td>
<td>US, Jh. of W.</td>
<td>Yeast</td>
</tr>
<tr>
<td>LOGOS</td>
<td>HMDM</td>
<td>EM</td>
<td>IC</td>
<td>(Xing et al., 2004)</td>
<td>None</td>
<td>US, Berk</td>
<td>Yeast, Dros</td>
</tr>
<tr>
<td>cWinnower</td>
<td>String</td>
<td>Graph</td>
<td>Mismatch</td>
<td>(Li et al., 2004)</td>
<td>None</td>
<td>US, NASA</td>
<td>Artic</td>
</tr>
<tr>
<td>EC</td>
<td>String</td>
<td>Genetic</td>
<td>Fitness</td>
<td>(Bogel et al., 2004)</td>
<td>Request</td>
<td>US, Eli Lilly</td>
<td>Mammals</td>
</tr>
<tr>
<td>GMS-MP</td>
<td>GWM</td>
<td>Gibbs</td>
<td>MAP</td>
<td>(Zhou &amp; Liu, 2004)</td>
<td>Bin(Win)</td>
<td>US, Harvard</td>
<td>Mammals</td>
</tr>
<tr>
<td>MDScan</td>
<td>Matrix</td>
<td>Enum + Gibbs</td>
<td>MAP(hmm)</td>
<td>(Liu et al., 2002)</td>
<td>Src(Lic)</td>
<td>US, Harvard</td>
<td>Yeast</td>
</tr>
<tr>
<td>Kamvysselis</td>
<td>String</td>
<td>Enum</td>
<td>see text</td>
<td>(Kamvysselis et al., 2003)</td>
<td>None</td>
<td>GM</td>
<td>Yeast</td>
</tr>
<tr>
<td>MoDEL</td>
<td>String</td>
<td>EC</td>
<td>IC</td>
<td>(Hernandez et al., 2004)</td>
<td>CH, SIB</td>
<td>Non-DNA</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2.: Some discovery algorithms, part II

Legend: Dyad: Supports the search for two elements close to each other. PD: Pattern-driven, SD: Sample-driven, IC: Information Content, Suffix: Suffixtree, Prefix: Prefixtree
Anno 2007  (BIIT and Quretec)
Tartu, ESTONIA