Probabilistic Algorithms for Mining in Large Streams

Philippe Flajolet,
Algorithms; INRIA–Rocquencourt (France)

— ITC Paris, September 2009 —

Determine quantitative characteristics of LARGE data ensembles?

In-between:

- Computer Science (algorithms, complexity)
- Mathematics (combinatorics, probability, asymptotics)
- Application fields (texts, genomic seq's, networks, stats . . .)

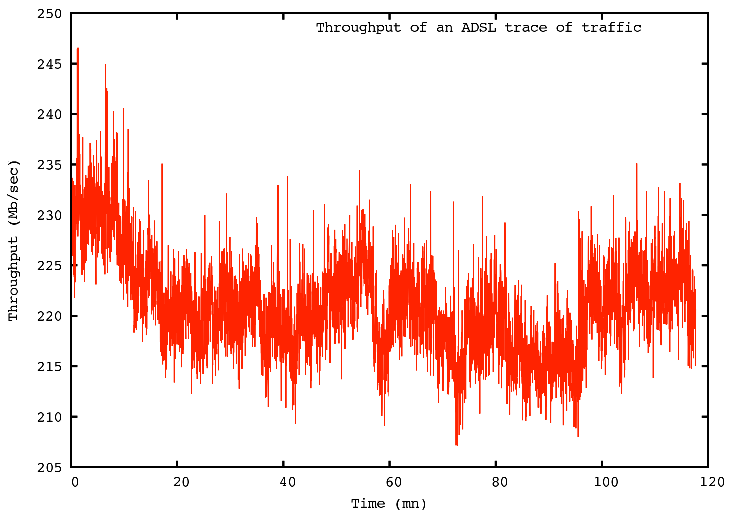
1 ALGORITHMICS OF MASSIVE DATA SETS

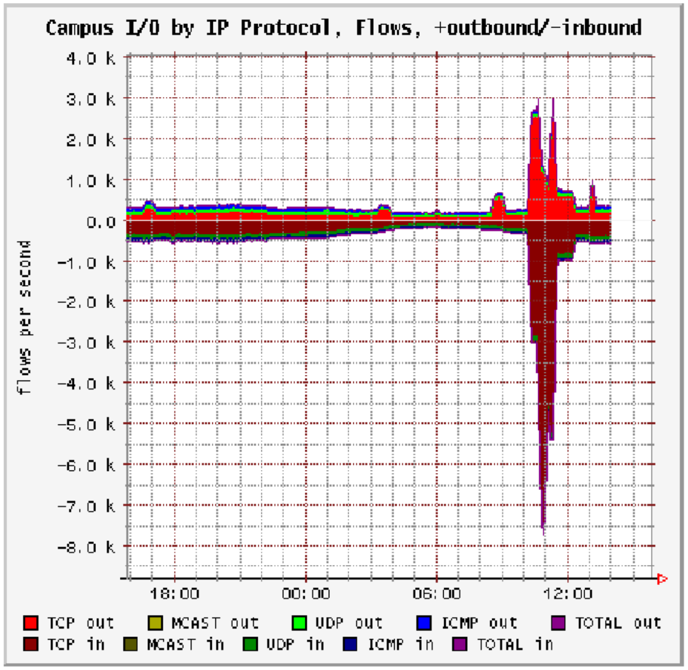

Routeurs $\approx$ Terabits/sec (10^{12}b/s).

Google indexes 10 billion pages & prepares 100 Petabytes of data (10^{17}B).

Stream algorithms = one pass;
memory $\leq$ one printed page

Example: Propagation of a virus and attacks on networks





(Raw ADSL traffic) *(Attack)*

Raw volume **Cardinality**

(based on Estan–Vargese)

The cardinality problem

— Data: stream $s = s_1 s_2 \cdots s_\ell$, $s_j \in \mathcal{D}$, $\ell \propto 10^9$.

— Output: Estimation of the cardinality n, $n \propto 10^7$.

— Conditions:

very little extra memory;
a single "simple" pass;
no statistical hypothesis.
accuracy within 1% or 2%.

More generally . . .

- Cardinality: number of distinct values;
- Icebergs: number of values with relative frequency $> 1/30$;
- Mice: number of values with absolute frequency < 10;
- Elephants: number of values with absolute frequency > 100;
- Moments: measure of the profile of data . . .

Applications: *networks;* quantitative *data mining; very large data bases and sketches; internet; fast rough analysis of sequences.*

2 ICEBERGS

A *k-iceberg* is a value whose relative frequency is $> 1/k$.

`abracadabraba babies babble bubbles alhambra`

very little extra memory;
a single "simple" pass;
no statistical hypothesis.
accuracy within 1% or 2%.

$k = 2$. Majority $\equiv$ 2-iceberg: `a b r a c a d a b r a ...`

The gang war $\equiv$ 1 register $\langle \texttt{value,counter} \rangle$

$k > 2$. Generalisation with $k - 1$ registers.

Provides a superset —no loss— of icebergs.
(+ Filter and combine with sampling.)

(Karp-Shenker-Papadimitriou 2003)

3 CARDINALITY

- **HASHING** provides values that are (quasi) uniformly random.
- **Randomness becomes reproducible**:

`canada` `uruguay` `france` $\cdots$ `uruguay` $\cdots$

3589 3589

A data stream $\leadsto$ a multi-set of uniform reals $[0, 1]$

An observable = a function of the hashed set.

An observable = a function of the hashed set.

— A. The minimum of values seen is `0.0000001101001`
— B. We have seen all patterns $\texttt{0.}x_1 \cdots x_{20}$ for $x_j \in \{0, 1\}$.

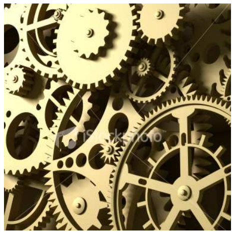
NB: "We have seen a total of 1968 `bits = 1` is *not* an observable.

Plausibly(??):

A indicates $n \approx 2^7$ (?); B indicates $n \geq 2^{20}$ (!).

(F.-Martin 1985), (Astrahan-Schkolnick-Whang 1987), (Alon-Matias-Szegedy 1999)...

3.1 Hyperloglog



The internals of the best algorithm known

Step 1. Choose the observable.

The observable O is the maximum of positions of the first 1

11000	10011	01010	10011	01000	00001	01111
1	1	2	1	2	**5**	2

= **a single integer register** < 32 $(n < 10^9)$
≡ **a small "byte"** (5 bits)

(F-Martin 1985); (Durand-F. 2003); (F-Fusy-Gandouet-Meunier 2007)

Step 2. Analyse the observable.

Theorem.

(i) Expectation: $\mathbb{E}_n(O) = \log_2(\varphi n) +$ oscillations $+ o(1)$.

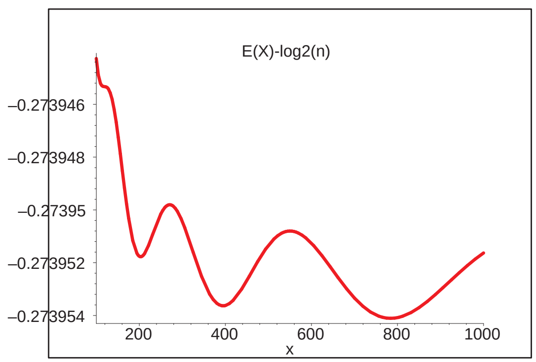
(ii) Variance: $\mathbb{V}_n(O) = \xi +$ oscillations $+ o(1)$.

Get *estimate* of the logarithmic value of n with a systematic bias (φ) and a dispersion (ξ) of $\approx \pm$ 1 binary order of magnitude.

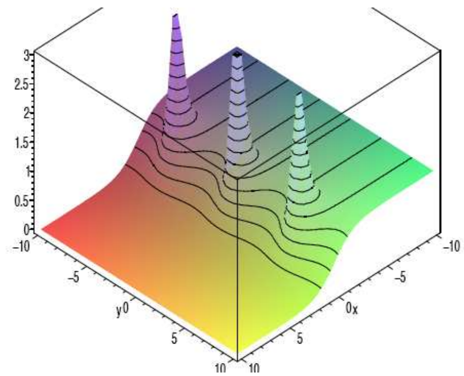
$\leadsto$ Correct bias; improve accuracy!

The Mellin transform: $\int_0^\infty f(x)x^{s-1}\,dx$.

- *Factorises linear superpositions of models at different scales;*
- *Relates asymptotics and complex singularities of* $\int$.



(singularities)



(asymptotics)

Algorithm Skeleton(S : stream):
initialise a register $R := 0$;
for $x \in S$ **do**
$h(x) = b_1 b_2 b_3 \cdots$;
$\rho := \text{position}_{1\uparrow}(b_1 b_2 \cdots)$;
$R := \max(R, \rho)$;
compute the estimator of $\log_2 n$.

= a single "small byte" of $\boxed{\log_2 \log_2 N}$ bits: 5 bits for $N = 10^9$;

= correction by $\varphi = e^{-\gamma}/\sqrt{2}$; ($\gamma$:= Euler's constant)

= unbiased; limited accuracy: $\pm$ one binary order of magnitude.

Step 3. Design a real-life algorithm.

Plan A: Repeat m times the experiment
& take arithmetic average. +Correct bias.

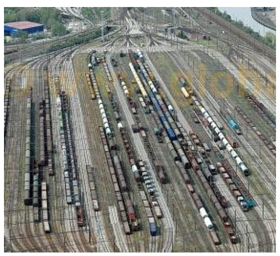
Estimate $\log_2 n$ with accuracy $\approx \pm \frac{1}{\sqrt{m}}$.

($m = 1000 \Longrightarrow$ accuracy = a few percents.)

Computational costs are multiplied by m.

+ Limitations due to dependencies ..

Plan B, "**Stochastic averaging**": Split data into m batches; compute finally an average of the estimates of each batch.



Algorithm HyperLoglog(S : stream; $m = 2^{10}$):
initialise m registers $R[\,]$:= 0;
for $x \in S$ **do**
    $h(x) = b_1 b_2 \cdots;\quad A := \langle b_1 \cdots b_{10} \rangle_{\text{base } 2};$
    ρ:=$\text{position}_{1\uparrow}(b_{11} b_{12} \cdots);$
    $R[A] := \max(R[A], \rho);$
compute the estimator of cardinality n.

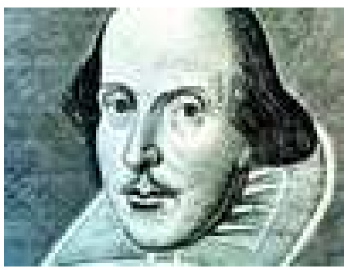
The complete algorithm has $O(12)$ instructions + hashing.
It computes the *harmonic mean* of $2^{R[j]}$; then multiplies by m.

Analysis-based algorithmic engineering: correct the systematic bias; then the non-asymptotic bias.

Mathematical analysis (combinatorial, probabilistic, asymptotic) enters design in a non-trivial fashion.
(Here: Mellin + saddle-point methods).
$\leadsto$ **Theorem:** *For m registers, the standard (relative) error is $\frac{1.035}{\sqrt{m}}$.*

With 1024 bytes, estimate cardinalities till 10^9 with standard error 1.5%.

Whole of Shakespeare: 128bytes $(m = 256)$



$\mapsto$

```
ghfffghfghgghggggghghheehfhfhhgghghghhfgffffhhhiigfhhffgfiihfhhh
igigighfgihfffghigihghigfhhgeegeghgghhhgghhfhidiigihighihehhhfgg
hfgighigffghdieghhhgghhfghhfiiheffghghihifgggffihgihfggighgiiif
fjgfgjhhjiifhjgehgghfhhfhjhiggghghihigghhihihgiighgfhlgjfgjjjmfl
```

Estimate $n^\circ \approx 30,897$ against $n = 28,239$ distinct words.
Error is $+9.4\%$ for **128 bytes**(!!)

3.2 Distributed applications

Given 90 phonebooks, how many different names?

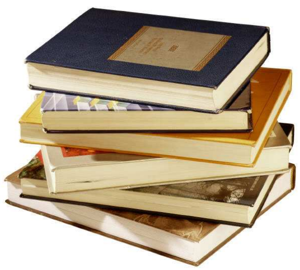
Collection of the registers $R_1, \ldots, R_m$ of $S \equiv$ signature of S.

Signature of union $=$ max/components ($\vee$):

$$\begin{cases} \mathbf{sign}(A \cup B) & = \mathbf{sign}(A) \vee \mathbf{sign}(B) \\ |A \cup B| & = \mathbf{estim}(\mathbf{sign}(A \cup B)). \end{cases}$$

Estimate within 1% the number of different names by sending 89 faxes, each of about one-quarter of a printed page.

3.3 Document comparison



Can one classify a million books, according to similarity, with a portable computer?

$$\begin{cases} |A| & = & \mathbf{estim}(\mathbf{sign}(A)) \\ |B| & = & \mathbf{estim}(\mathbf{sign}(B)) \\ |A \cup B| & = & \mathbf{estim}(\mathbf{sign}(A) \vee \mathbf{sign}(B)) \end{cases}$$

$$\boxed{\mathbf{simil}(A, B) = \frac{|A| + |B| - |A \cup B|}{|A \cup B|}}.$$

Given a library of N books (e.g.: $N = 10^6$) with total volume of V characters (e.g.: $V = 10^{11}$).

— Exact solution: quadratic time and/or linear storage

— Solution by signatures: linear time + $O(N^2)$ & small storage.

4 ADAPTIVE SAMPLING

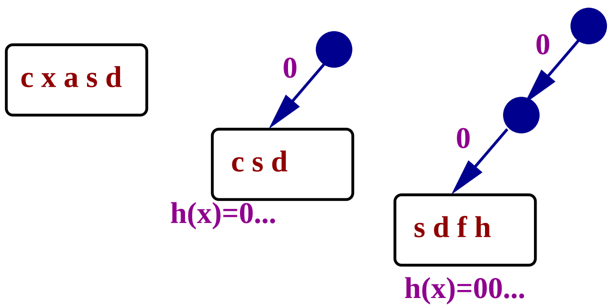
Can one localise the geographical center of a country given a file ⟨`persons & townships`⟩*?*

— Exact: yes! = eliminate duplicate cities (“projection”)

— Approximate (?): Use straight sampling
⟹ France = somewhere very near to PARIS(!!).

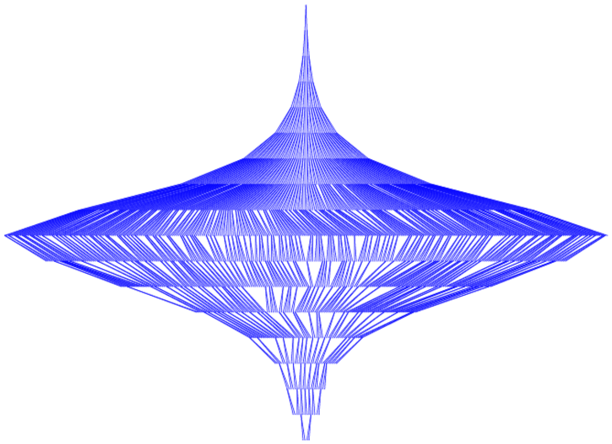
Sampling uniformly *over the domain of* **distinct** *values?*

Adaptive sampling:



Algorithm: Adaptive Sampling(S : stream);
$C := \emptyset$;{cache of capacity m}
$p := 0$; {depth}
for $x \in S$ **do**
  if $h(x) = 0^p \cdots$ **then** $C := C \cup \{x\}$;
  if overflow(C) **then** p := p+1; filter C;
return C $\{\approx m/2 \ldots m$ elements}.

(Wegman 1980) (F 1990) (Louchard 1997)



Analysis is related to the digital tree structure: data compression; text search; communication protocols; &c.

- Provides an unbiased sample of **distinct values**;
- Provides an unbiased **cardinality estimator**:

$$\text{estim}(S) := |C| \cdot 2^p.$$

Hamlet

• **Straight sampling** (13 elements):

and, and, be, both, i, in, is, leaue, my, no, ophe, state, the

`Google` (leaue$\mapsto$leave, ophe$\mapsto$ $\emptyset$) = `38,700,000`.

• **Adaptive sampling** (10 elements):

danskers, distract, fine, fra, immediately, loses, martiall, organe, passeth, pendant

`Google = 8`, all pointing to Shakespeare/ Hamlet $\leadsto$ *mice*, later!

5 MICE

Adaptive sampling plus counters!

— Hamlet: *danskers*^{1}, *distract*^{1}, *fine*^{9}, *fra*^{1}, *immediately*^{1}, *loses*^{1}, *martiall*^{1}, *organe*^{1}, *passeth*^{1}, *pendant*^{1}.

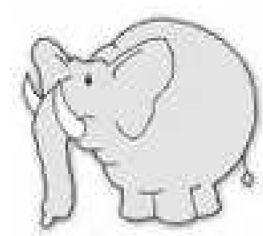
Cache of size =100, gives a sample of 79 elements.

$$\mathbf{1^{50}, 2^{14}, 3^{4}, 4^{2}, 5^{1}, 6^{1}, 9^{1}, 13^{1}, 15^{1}, 28^{1}, 43^{2}, 128^{1}}.$$

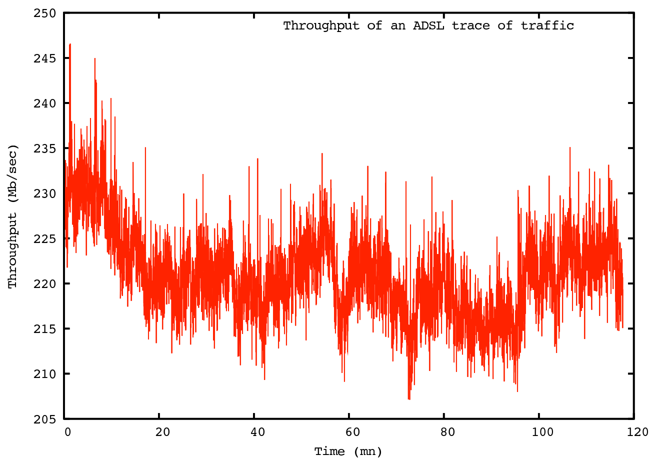
	1-Mice	2-Mice	3-Mice
Estimated	63%	17%	5%
Actual	60%	14%	6%

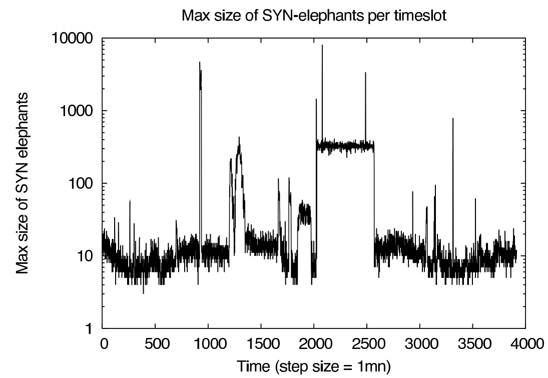
The ten most frequent words of Hamlet are *the, and, to, of, i, you, a, my, it, in*. They represent $>$ 20% of the whole text. With 20 words, capture 30%; with 50 words, 44%. **70 words capture 50% du texte!**.

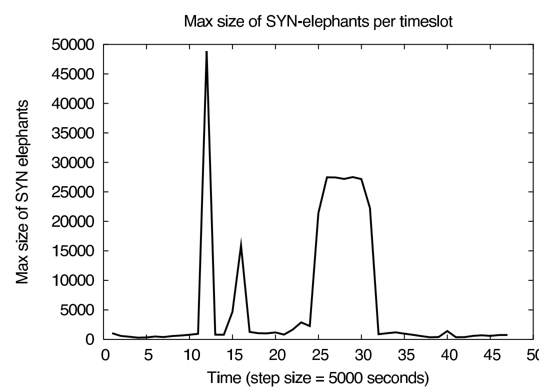
6 ELEPHANTS



A k-elephant is a value whose absolute frequency is $\geq k$.





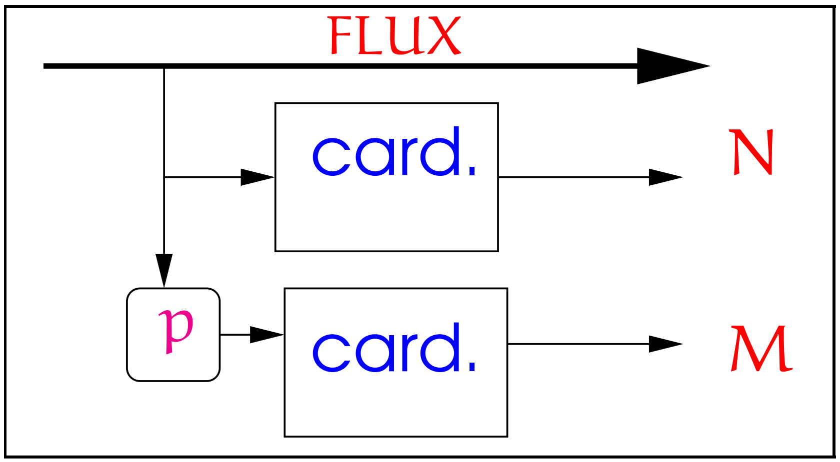


Network attacks by Denial of Service (Y. Chabchoub, Ph. Robert)

Complexity Theorem (Alon *et al.*) *It is not possible to determine the largest frequency with sub-linear memory.*

- One cannot find a needle in a haystack.
- But one can still find (easily) much information . . .

Bi-modal traffic: A stream composed of 1–mice and 10–elephants.

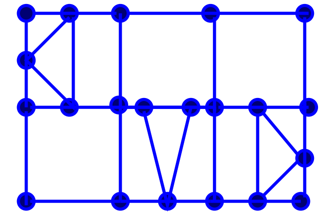


$$\begin{cases} & (p = \frac{1}{10}) \\ N = N_s + N_e + \text{noise} \\ M = \frac{1}{10} N_s + 0.65 N_e + \text{noise} \end{cases}$$

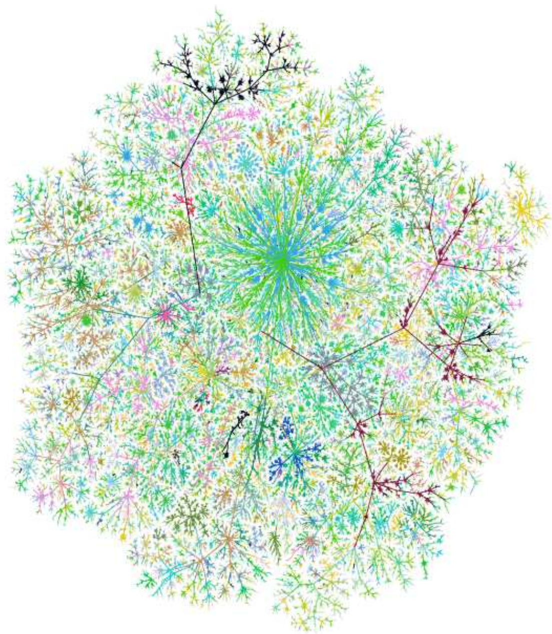
Solution: $$\boxed{N_e \approx \frac{10M - N}{5.5}}$$ (A. Jean-Marie, O. Gandouet, 2007)

7 APPLICATIONS

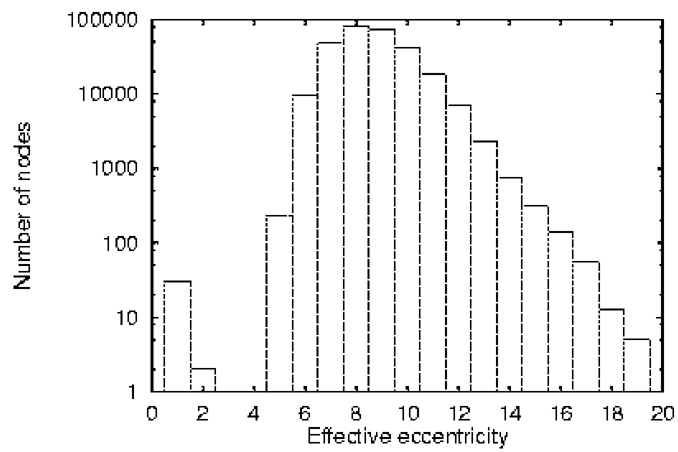
- Data mining in graphs
- Document classification: an experiment
- Fast mining in genomic sequences
- Profiling: frequency moments



- Number of symmetric links in large graph; number of triangles.
- The histogram of excentricities in the internet graph:



$\Longrightarrow$

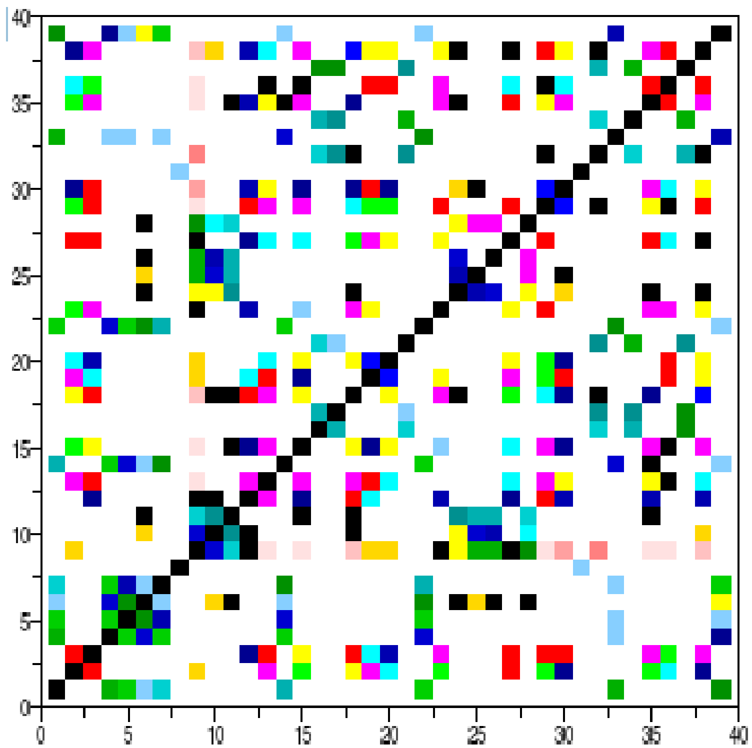


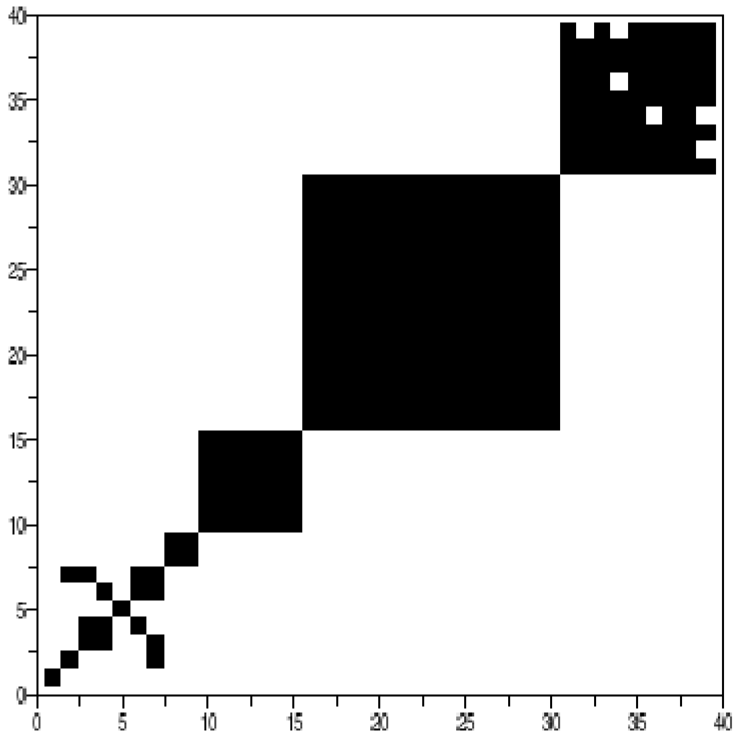
b) Histogram of diameters

Gain: $\times 300$.

(Palmer, Gibbons, Faloutsos^2, Siganos 2001) Internet graph: 285k nodes, 430kedges.

How many languages? **कश्यपः**

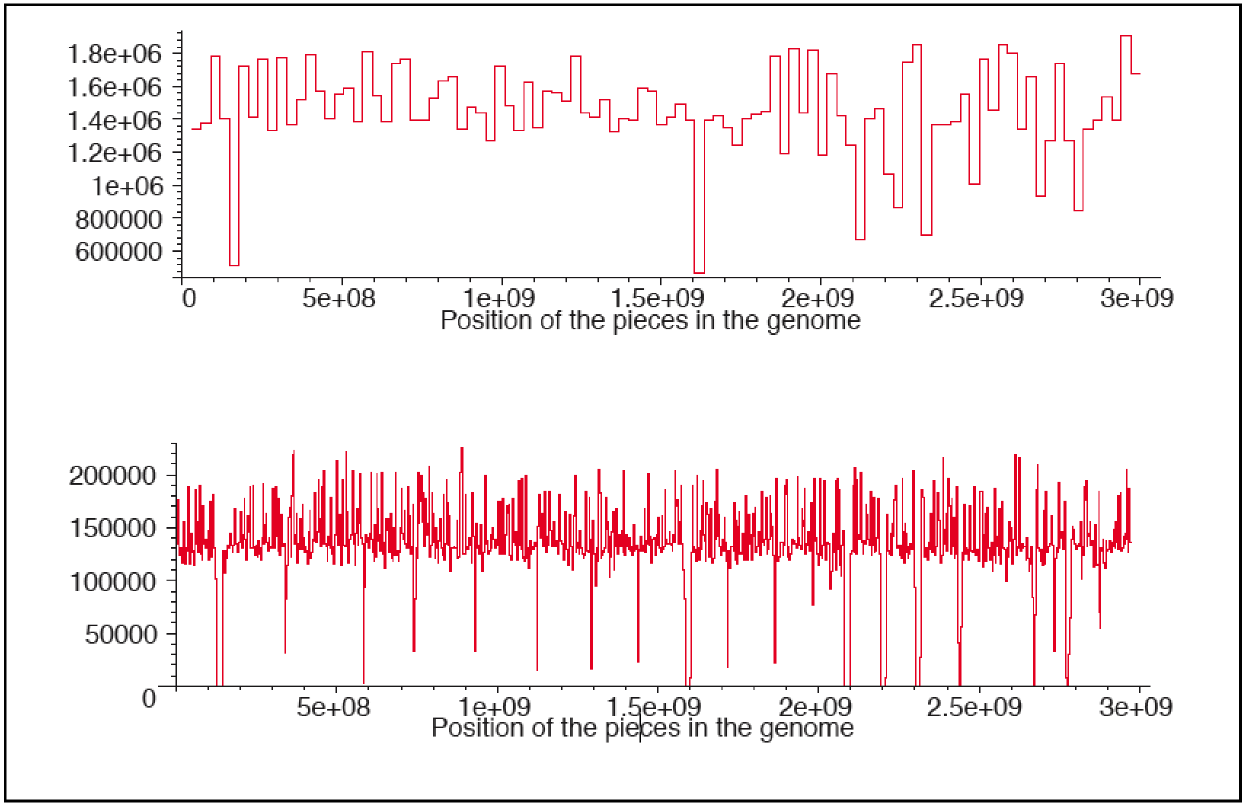




(Pranav Kashyap: word-level encrypted texts; classification by language; use $\vartheta \approx 20\%$.)

+ Use **shingles** (overlapping blocks = small phrases) for finer classification.

Genome



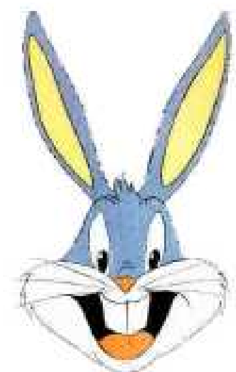
(Giroire 2006: # patterns of length 13 in genome)

Profiling: frequency moments

Alon-Matias-Szegedy: $F_p := \sum_v (f_v)^p$ where f_v:= frequency of value v.

dim = n $\longrightarrow$ dim $\approx \log n$

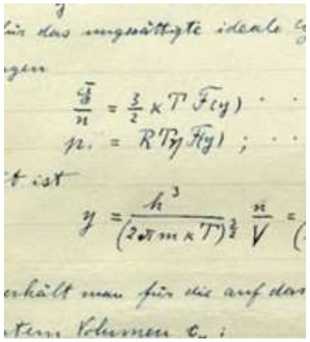
Johnson–Lindenstrauss
embeddings
dimension reduction
Indyk's beautiful ideas



Use of random Gaussian projections for F_2; Stable laws for $0 \leq p \leq 2$.

Conclusions

Possibilities (within limits!) of probabilistic algorithms.



Continuum: maths $\leadsto$ comp. sc. $\leadsto$ technology.