Analytic Pattern Matching: From DNA to Twitter

Wojciech Szpankowski[*]
Purdue University
W. Lafayette, IN 47907

March 10, 2015

Center for
Science of Information
NSF Science & Technology Center

Information Theory, Learning, and Big Data, Berkeley, 2015

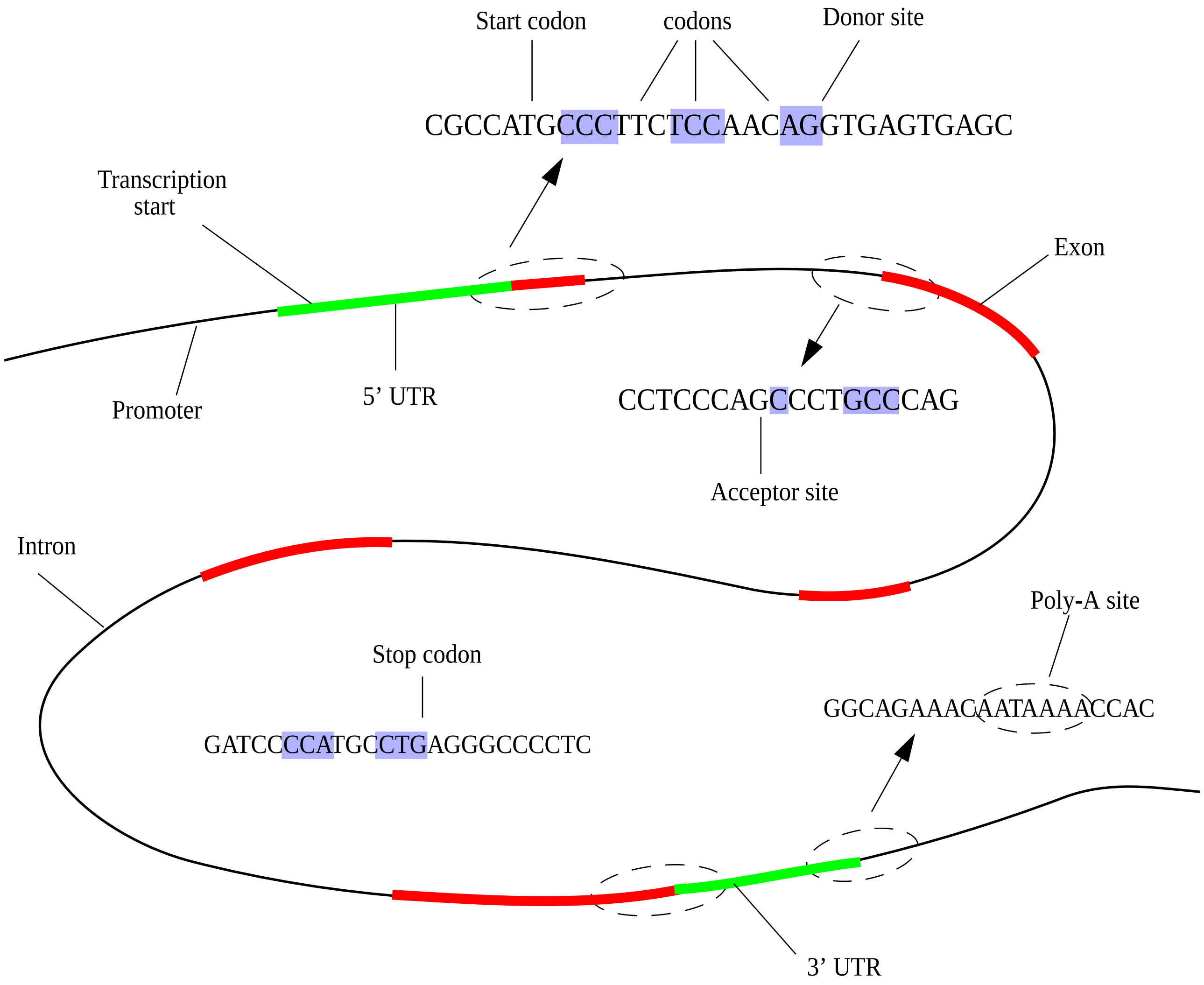
[*] Jont work with Philippe Jacquet

Outline

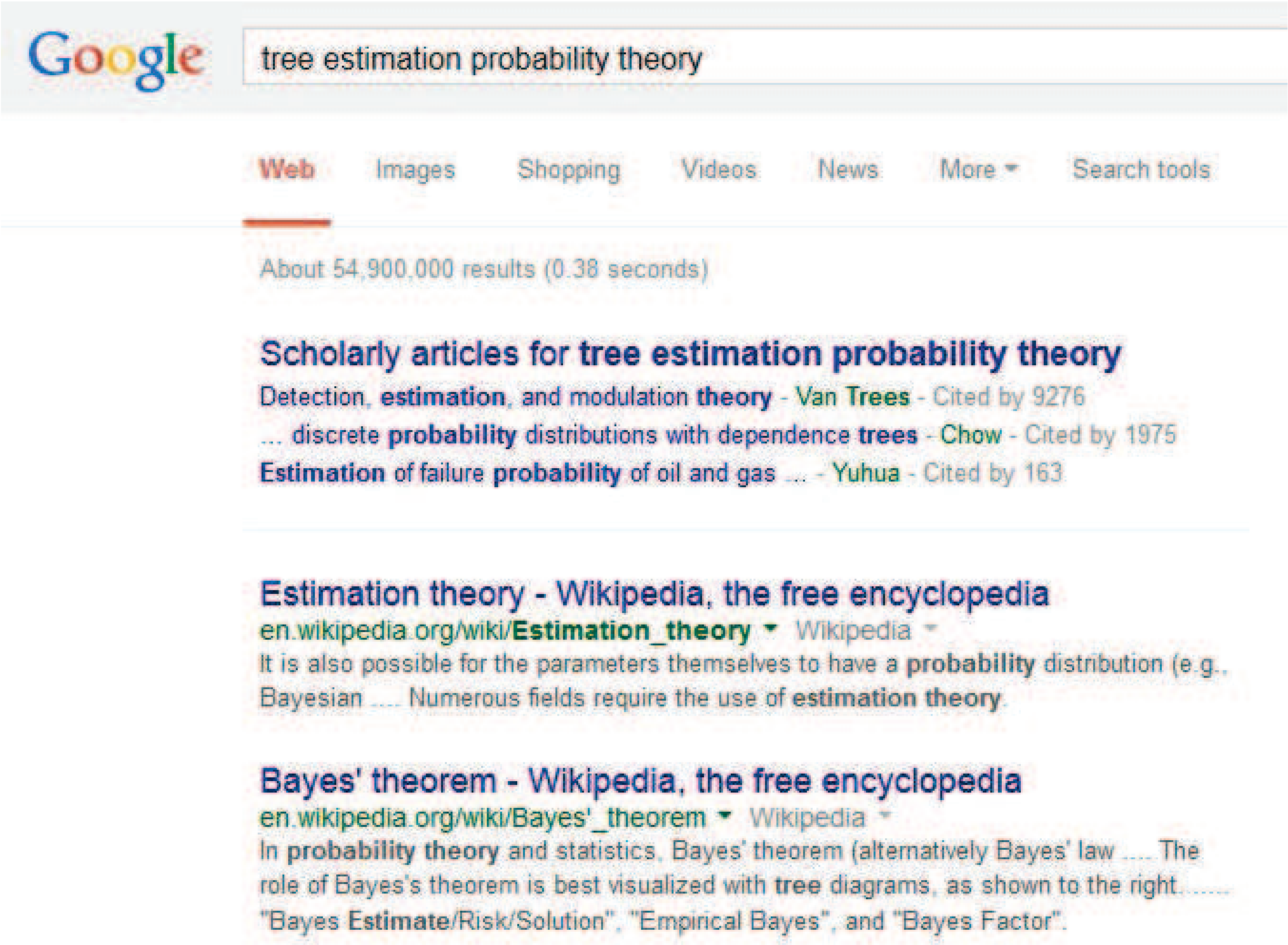
1. Motivations
   - Finding Biological Signals
   - Searching Google
   - Classifying Twitter
2. Pattern Matching Problems
   - Exact String Matching
   - Constrained String Matching
   - Generalized String Matching
   - Subsequence String Matching
   - String Complexity
3. Example of an Analysis:
   Exact String Matching.

Motivation – Biology & String Matching

Biological world is highly stochasticand inhomogeneous (S. Salzberg).



Motivation – Google & Subsequnce Matching



tree estimation probability theory

About 54,900,000 results (0.38 seconds)

Scholarly articles for tree estimation probability theory

Detection, **estimation**, and modulation **theory** - Van **Trees** - Cited by 9276

... discrete **probability** distributions with dependence **trees** - Chow - Cited by 1975

Estimation of failure **probability** of oil and gas ... - Yuhua - Cited by 163

Estimation theory - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/**Estimation_theory** - Wikipedia

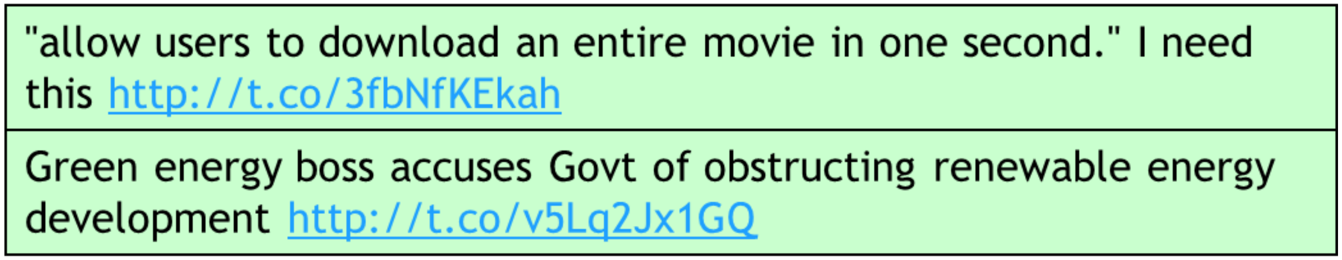
It is also possible for the parameters themselves to have a **probability** distribution (e.g., Bayesian .... Numerous fields require the use of **estimation theory**.

Bayes' theorem - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Bayes'_theorem - Wikipedia

In **probability theory** and statistics, Bayes' theorem (alternatively Bayes' law ..... The role of Bayes's theorem is best visualized with **tree** diagrams, as shown to the right. ...... "Bayes **Estimate**/Risk/Solution", "Empirical Bayes", and "Bayes Factor".

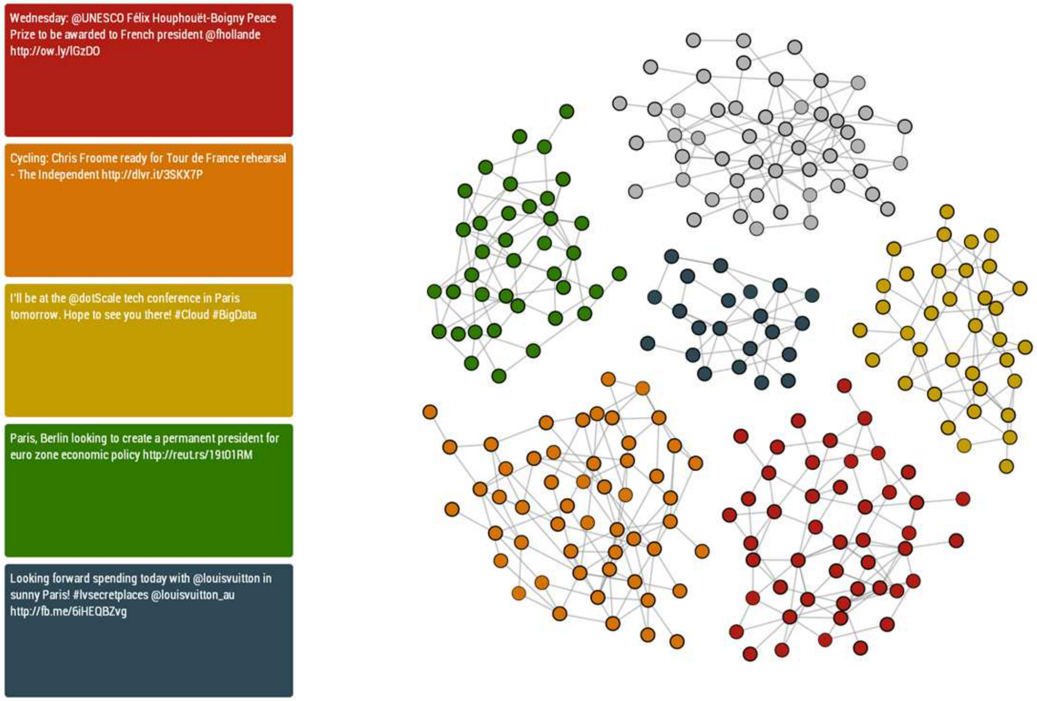
Motivation – Twitter & String Complexity



"allow users to download an entire movie in one second." I need this http://t.co/3fbNfKEkah

Green energy boss accuses Govt of obstructing renewable energy development http://t.co/v5Lq2Jx1GQ

Figure 1: Two similar twitter texts have many common words



Wednesday: @UNESCO Félix Houphouët-Boigny Peace Prize to be awarded to French president @fhollande http://ow.ly/lGzDO

Cycling: Chris Froome ready for Tour de France rehearsal - The Independent http://dlvr.it/3SKX7P

I'll be at the @dotScale tech conference in Paris tomorrow. Hope to see you there! #Cloud #BigData

Paris, Berlin looking to create a permanent president for euro zone economic policy http://reut.rs/19t01RM

Looking forward spending today with @louisvuitton in sunny Paris! #lvsecretplaces @louisvuitton_au http://fb.me/6iHEQBZvg

Figure 2: Twitters Classification

Outline Update

1. Motivations
2. Pattern Matching Problems
   - Exact String Matching
   - Constrained String Matching
   - Generalized String Matching
   - Subsequence String Matching
3. Example of an Analysis: Exact String Matching.

Pattern Matching

Let $\mathcal{W}$ and T be (set of) strings generated over a finite alphabet $\mathcal{A}$.

We call $\mathcal{W}$ the pattern and T the text. The text T is of length n and is generated by a probabilistic source.

We shall write

$$T_m^n = T_m \dots T_n.$$

The pattern $\mathcal{W}$ can be a single string

$$\mathcal{W} = w_1 \dots w_m, \quad w_i \in \mathcal{A}$$

or a set of strings

$$\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_d\}$$

with $\mathcal{W}_i \in \mathcal{A}^{m_i}$ being a set of strings of length m_i.

Basic question:

how many times $\mathcal{W}$ occurs in T (or how long to wait until $\mathcal{W}$ occurs in T).

Define

$$O_n(\mathcal{W}) = \#\{i : \ T_{i-m+1}^i = \mathcal{W}, \ m \le i \le n\}.$$

Varations on Pattern Matching

(Exact) String Matching

In the exact string matching the pattern $\mathcal{W} = w_1 \dots w_m$ is a given string (i.e., consecutive sequence of symbols).

Generalized String Matching

In the generalized pattern matching a set of patterns (rather than a single pattern) is given, that is,

$$\mathcal{W} = (\mathcal{W}_0, \mathcal{W}_1, \dots, \mathcal{W}_d), \quad \mathcal{W}_i \in \mathcal{A}^{m_i}$$

where $\mathcal{W}_i$ itself for $i \geq 1$ is a subset of $\mathcal{A}^{m_i}$ (i.e., a set of words of a given length m_i).
The set $\mathcal{W}_0$ is called the forbidden set.

Three cases to be considered:

$\mathcal{W}_0 = \emptyset$ — one is interested in the number of patterns from $\mathcal{W}$ occurring in the text.

$\mathcal{W}_0 \neq \emptyset$ — we study the number of $\mathcal{W}_i$, $i \geq 1$ pattern occurrences under the condition that no pattern from $\mathcal{W}_0$ occurs in the text.

$\mathcal{W}_i = \emptyset$, $i \geq 1$, $\mathcal{W}_0 \neq \emptyset$ — restricted pattern matching.

Pattern Matching Problems

Hidden Words or Subsequence Pattern Matching

In this case we search in text for a subsequence $\mathcal{W} = w_1 \ldots w_m$ rather than a string, that is, we look for indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ such that

$$T_{i_1} = w_1, \; T_{i_2} = w_2, \cdots, \; T_{i_m} = w_m.$$

We also say that the word $\mathcal{W}$ is "hidden" in the text.

For example:

$$\begin{aligned} \mathcal{W} &= \textbf{date} \\ T &= \textbf{hidden pattern} \end{aligned}$$

occurs four times as a subsequence in the text as hidden pattern but not even once as a string.

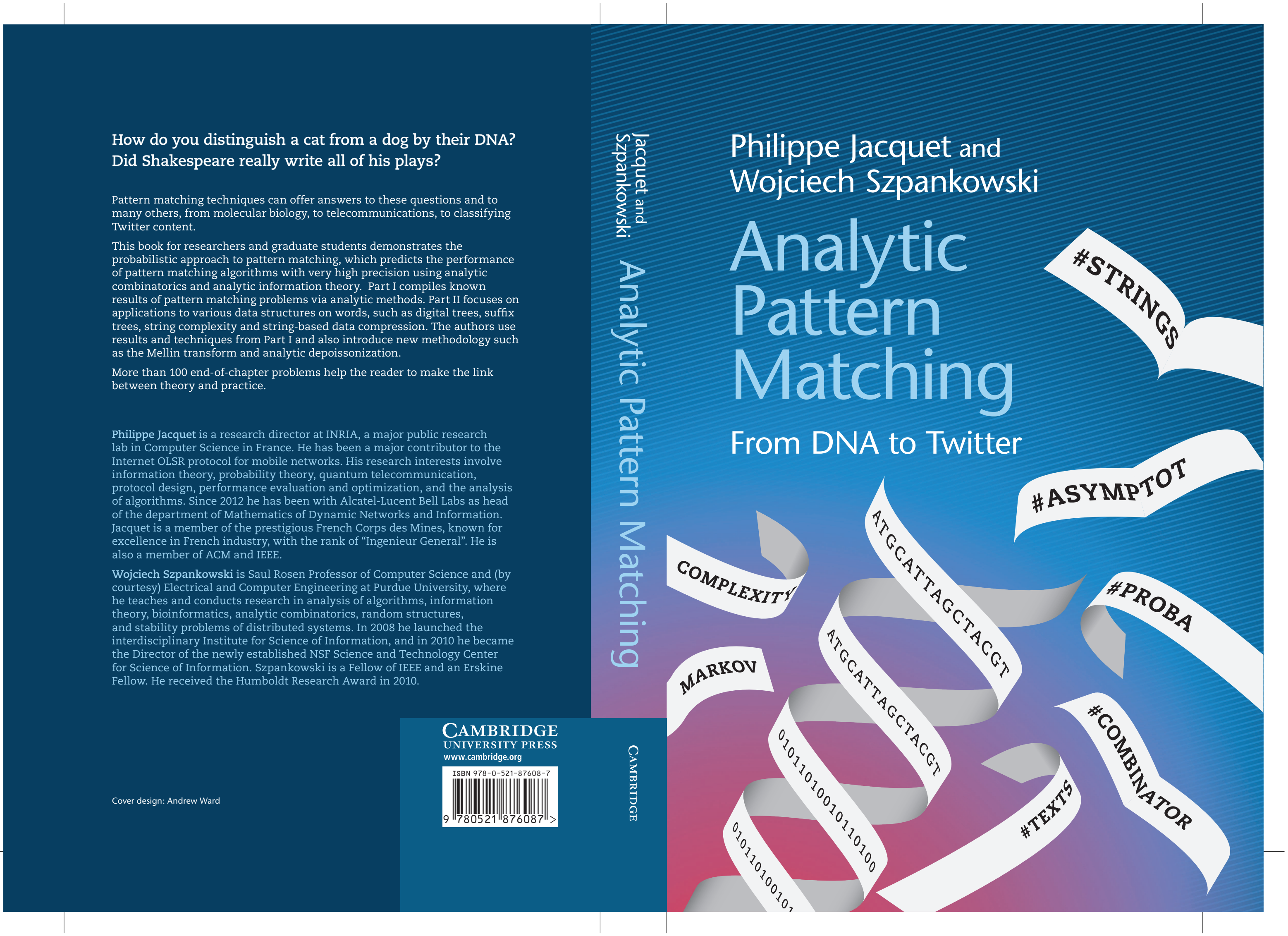
Self-Repetitive Pattern Matching

n this case the pattern $\mathcal{W}$ is part of the text:

$$\mathcal{W} = T_1^m.$$

We may ask when the first m symbols of the text will occur again. This is important in Lempel-Ziv like compression algorithms.

New Book on Pattern Matching



How do you distinguish a cat from a dog by their DNA? Did Shakespeare really write all of his plays?

Pattern matching techniques can offer answers to these questions and to many others, from molecular biology, to telecommunications, to classifying Twitter content.

This book for researchers and graduate students demonstrates the probabilistic approach to pattern matching, which predicts the performance of pattern matching algorithms with very high precision using analytic combinatorics and analytic information theory. Part I compiles known results of pattern matching problems via analytic methods. Part II focuses on applications to various data structures on words, such as digital trees, suffix trees, string complexity and string-based data compression. The authors use results and techniques from Part I and also introduce new methodology such as the Mellin transform and analytic depoissonization.

More than 100 end-of-chapter problems help the reader to make the link between theory and practice.

Philippe Jacquet is a research director at INRIA, a major public research lab in Computer Science in France. He has been a major contributor to the Internet OLSR protocol for mobile networks. His research interests involve information theory, probability theory, quantum telecommunication, protocol design, performance evaluation and optimization, and the analysis of algorithms. Since 2012 he has been with Alcatel-Lucent Bell Labs as head of the department of Mathematics of Dynamic Networks and Information. Jacquet is a member of the prestigious French Corps des Mines, known for excellence in French industry, with the rank of "Ingenieur General". He is also a member of ACM and IEEE.

Wojciech Szpankowski is Saul Rosen Professor of Computer Science and (by courtesy) Electrical and Computer Engineering at Purdue University, where he teaches and conducts research in analysis of algorithms, information theory, bioinformatics, analytic combinatorics, random structures, and stability problems of distributed systems. In 2008 he launched the interdisciplinary Institute for Science of Information, and in 2010 he became the Director of the newly established NSF Science and Technology Center for Science of Information. Szpankowski is a Fellow of IEEE and an Erskine Fellow. He received the Humboldt Research Award in 2010.

Cover design: Andrew Ward

CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org

ISBN 978-0-521-87608-7
9 780521 876087

Jacquet and Szpankowski Analytic Pattern Matching CAMBRIDGE

Philippe Jacquet and
Wojciech Szpankowski

Analytic Pattern Matching

From DNA to Twitter

9780521876087: Jacquet & Szpankowski: PPC: C M Y K

Book Contents: Part I. ANALYSIS

Chapter 1: **Probabilistic Models**

Chapter 2: **Exact String Matching** (DNA Applications)
2.1 Formulation of the problem
2.2 Language representation
2.3 Generating functions
2.4 Moments
2.5 Limit laws

Chapter 3: **Constrained Exact String Matching** (Constrained Coding)
3.1 Enumeration of (d, k) sequences
3.7 Application: Significant signals in neural data

Chapter 4: **Generalized String Matching** (Biological Applications)
4.1 String matching over a reduced set
4.2 Generalized string matching via automata
4.3 Generalized string matching via a language approach

Chapter 5: **Subsequence String Matching** (Google Applications)
5.1 Problem formulation
5.2 Mean and variance analysis
5.3 Autocorrelation polynomial revisited
5.4 Central limit laws
5.5 Limit laws for fully constrained pattern
5.6 Generalized subsequence problem

Book Contents: Part II. APPLICATIONS

Outline Update

1. Motivations
2. Pattern Matching Problems
3. Example of an Analysis:
   Exact String Matching

Analysis: Exact String Matching

In the exact string matching the pattern $\mathcal{W} = w_1 \ldots w_m$ is a given string and one searches for its occurrences in a random text T_1^n.

Memoryless Source: The text is a realization of an independently, identically distributed sequence of random variables such that a symbol $s \in \mathcal{A}$ occurs with probability $P(s)$.

Extensions to **Markovian Source** are relatively easy.

Objective: probabilistic laws for

$$O_n(\mathcal{W}) = \#\{i : \ T_{i-m+1}^i = \mathcal{W}, \ m \le i \le n\}.$$

Tools. Symbolic calculus and analytic tools of languages:

Language $\mathcal{L}$ is a collection of words satisfying some properties.
Generating function $L(z)$ of language $\mathcal{L}$ is defined as

$$L(z) = \sum_{u \in \mathcal{L}} P(u) z^{|u|}$$

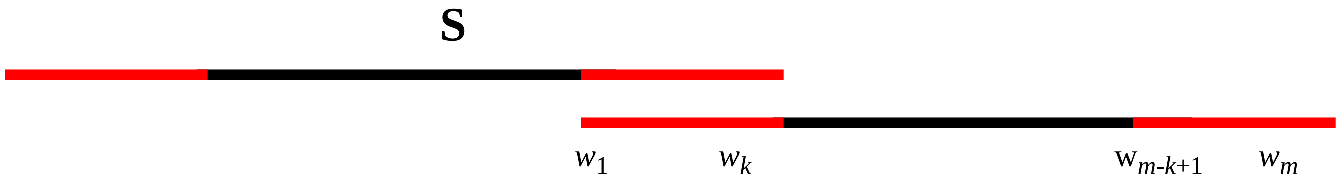
where $P(w)$ is the stationary probability u occurrence, $|u|$ is the length of w.

Autocorrelation Set and Polynomial

Given a pattern $\mathcal{W}$, we define the autocorrelation set $\mathcal{S}$ as:

$$\mathcal{S} = \{w_{k+1}^m : \ w_1^k = w_{m-k+1}^m\}, \quad w_1^k = w_{m-k+1}^m$$

and $\mathcal{WW}$ is the set of positions k satisfying $w_1^k = w_{m-k+1}^m$.



The generating function of $\mathcal{S}$ is $S(z)$ known also as the autocorrelation polynomial.

$$S(z) = \sum_{k \in \mathcal{WW}} P(w_{k+1}^m) z^{m-k}.$$

Example: Let $\mathcal{W} = bab$ over the alphabet $\mathcal{A} = \{a, b\}$.

$$\mathcal{WW} = \{1, 3\} \quad \text{and} \quad \mathcal{S} = \{\epsilon, ab\},$$

where ϵ is the empty word, since

```
b   a   b
        b   a   b
```

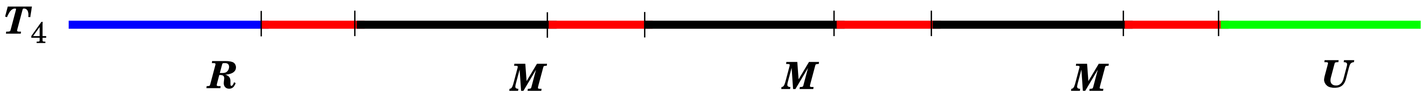
For the unbiased memoryless source: $S(z) = 1 + P(ab)z^2 = 1 + \frac{z^2}{4}$.

Language $\mathcal{T}_r$ and Associated Languages

Define $\mathcal{T}_r$ as set of words containing exactly $r \geq 1$ occurrences of $\mathcal{W}$:

$$\mathcal{T}_r = \mathcal{R} \cdot \mathcal{M}^{r-1} \cdot \mathcal{U}.$$

which can be illustrated as



(i) We define $\mathcal{R}$ as the set of words containing only one occurrence of $\mathcal{W}$, located at the right end. For example, for $\mathcal{W} = aba$, we have $ccaba \in \mathcal{R}$.

(ii) We also define $\mathcal{U}$ as

$$\mathcal{U} = \{u : \ \mathcal{W} \cdot u \cdot \in \mathcal{T}_1\}$$

that is, a word $u \in \mathcal{U}$ if $\mathcal{W} \cdot u$ has exactly one occurrence of $\mathcal{W}$ at the left end of $\mathcal{W} \cdot u$,

$$bba \in \mathcal{U}, \qquad ba \notin \mathcal{U}.$$

(iii) Let $\mathcal{M}$ be the language:

$$\mathcal{M} = \{u : \ \mathcal{W} \cdot u \in \mathcal{T}_2 \text{ and } \mathcal{W} \text{ occurs at the right of } \mathcal{W} \cdot u\},$$

that is, $\mathcal{M}$ is a language such that $\mathcal{W}\mathcal{M}$ has exactly two occurrences of $\mathcal{W}$ at the left and right end of a word from $\mathcal{M}$ (e.g., $ba \in \mathcal{M}$ since $ababa$).

Language Relations & Generating Functions

Lemma 1. (i) *The languages $\mathcal{M}, \mathcal{U}$ and $\mathcal{R}$ satisfy:*

$$\begin{aligned}\bigcup_{k\geq 1} \mathcal{M}^k &= \mathcal{A}^* \cdot \mathcal{W} + \mathcal{S} - \{\epsilon\},\\ \mathcal{U}\cdot\mathcal{A} &= \mathcal{M} + \mathcal{U} - \{\epsilon\}, \quad \mathcal{W}\cdot\mathcal{M} = \mathcal{A}\cdot\mathcal{R} - (\mathcal{R} - \mathcal{W}),\end{aligned}$$

where $\mathcal{A}^*$ is the set of all words.
(ii) *The generating functions associated with languages $\mathcal{M}, \mathcal{U}$ and $\mathcal{R}$ satisfy*

$$\begin{aligned}\frac{1}{1-M(z)} &= S_{\mathcal{W}}(z) + P(\mathcal{W})\frac{z^m}{1-z},\\ U_{\mathcal{W}}(z) &= \frac{M(z)-1}{z-1}, \qquad R(z) = P(\mathcal{W})z^m \cdot U_{\mathcal{W}}(z)\end{aligned}$$

Theorem 1. *The generating functions* $T_r(z) = \sum_{n\geq 0} \Pr\{O_n(\mathcal{W}) = r\} z^n$ *and* $T(z,u) = \sum_{r=1}^{\infty} T_r(z) u^r$ *satisfy*

$$\begin{aligned}T_r(z) &= R(z) M_{\mathcal{W}}^{r-1}(z) U_{\mathcal{W}}(z)\ , \qquad r \geq 1\\ T(z,u) &= R(z)\frac{u}{1-uM(z)} U_{\mathcal{W}}(z).\end{aligned}$$

Main Results: Asymptotics

Theorem 2. (i) *Moments. The expectation satisfies, for $n \geq m$:*

$$\mathbf{E}[O_n(\mathcal{W})] = P(\mathcal{W})(n - m + 1),$$

while the variance is

$$\mathbf{Var}[O_n(\mathcal{W})] = nc_1 + c_2$$

with

$$\begin{aligned} c_1 &= P(\mathcal{W})(2S(1) - 1 - (2m - 1)P(\mathcal{W}) \, , \\ c_2 &= P(\mathcal{W})((m-1)(3m-1)P(\mathcal{W}) - (m-1)(2S(1) - 1) - 2S'(1)). \end{aligned}$$

(ii) *Case $r = O(1)$. Let $\rho_{\mathcal{W}}$ be the smallest root of*

$$D_{\mathcal{W}}(z) = (1 - z)S_{\mathcal{W}}(z) + z^m P(\mathcal{W}) = 0.$$

Then

$$\Pr\{O_n(\mathcal{W}) = r\} \quad \sim \quad \sum_{j=1}^{r+1} (-1)^j a_j \binom{n}{j-1} \rho_{\mathcal{W}}^{-(n+j)}$$

where

$$a_{r+1} = \frac{\rho_{\mathcal{W}}^m P(\mathcal{W}) \left(\rho_{\mathcal{W}} - 1\right)^{r-1}}{\left(D'_{\mathcal{W}}(\rho_{\mathcal{W}})\right)^{r+1}} \, ,$$

and the remaining coefficients can be easily computed, too.

Central Limit and Large Deviations

(iii) CLT: Case $r = EO_n + x\sqrt{\mathbf{Var}O_n}$ for $x = O(1)$. Then:

$$\Pr\{O_n(\mathcal{W}) = r\} = \frac{1}{\sqrt{2\pi c_1 n}} e^{-\frac{1}{2}x^2} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right) .$$

(iv) Large Deviations: Case $r = (1+\delta)EO_n$. Let $a = (1+\delta)P(\mathcal{W})$ with $\delta \neq 0$. For complex t, define $\rho(t)$ to be the root of

$$1 - e^t M_{\mathcal{W}}(e^\rho) = 0 ,$$

while ω_a and σ_a are defined as

$$\begin{aligned} -\rho'(\omega_a) &= a \\ -\rho''(\omega_a) &= \sigma_a^2 \end{aligned}$$

Then

$$\Pr\{O_n(\mathcal{W}) \sim (1+\delta)EO_n\} = \frac{e^{-(n-m+1)I(a)+\delta_a}}{\sigma_a\sqrt{2\pi(n-m+1)}}$$

where $I(a) = a\omega_a + \rho(\omega_a)$ and δ_a is a constant.

Biology – Weak Signals and Artifacts

Denise and Regnier (2002) observed that in biological sequence whenever a word is overrepresented, then its subwords are also overrepresented.
For example, if $\mathcal{W}_1 = AATAAA$, then

$$\mathcal{W}_2 = ATAAAN$$

is also overrepresented.
Overrepresented subwords are called artifact, and it is important to disregard automatically noise created by artifacts.

New Approach:

> Once a dominating signal has been detected, we look for a weaker signal by comparing the number of observed occurrences of patterns to the conditional expectations **not** the regular expectations.

To solve this harder quastion one needs a new approach thru Generalized Pattern Matching discussed in **Chapter 4**. Thea, as in Denise and Regnier (2002) we find

$$\mathbf{E}[O_n(\mathcal{W}_2)|O_n(\mathcal{W}_1) = k] \sim \alpha n.$$

When $\mathcal{W}_1$ is overrepresented the constant α differs significantly from $\mathbf{E}[O_n(\mathcal{W}_2)]$.

Polyadenylation Signals in Human Genes

Beaudoing et al. (2000) studied several variants of the well known AAUAAA polyadenylation signal in mRNA of humans genes. To avoid artifacts Beaudoing et al cancelled all sequences where the overrepresented hexamer was found.

Using our approach Denise and Regnier (2002) discovered/eliminated all artifacts and found new signals in a much simpler and reliable way.

Hexamer	Obs.	Rk	Exp.	Z-sc.	Rk	Cd.Exp.	Cd.Z-sc.	Rk
AAUAAA	3456	1	363.16	167.03	1			1
AAAUAA	1721	2	363.16	71.25	2	1678.53	1.04	1300
AUAAAA	1530	3	363.16	61.23	3	1311.03	6.05	404
UUUUUU	1105	4	416.36	33.75	8	373 .30	37.87	2
AUAAAU	1043	5	373.23	34.67	6	1529.15	12.43	4078
AAAAUA	1019	6	363.16	34.41	7	848.76	5.84	420
UAAAAU	1017	7	373.23	33.32	9	780.18	8.48	211
AUUAAA	1013	I	373.23	33.12	10	385.85	31.93	3
AUAAAG	972	9	184.27	58.03	4	593.90	15.51	34
UAAUAA	922	10	373.23	28.41	13	1233.24	–8.86	4034
UAAAAA	922	11	363.16	29.32	12	922.67	9.79	155
UUAAAA	863	12	373.23	25.35	15	374.81	25.21	4
CAAUAA	847	13	185.59	48.55	5	613.24	9.44	167
AAAAAA	841	14	353.37	25.94	14	496.38	15.47	36
UAAAUA	805	15	373.23	22.35	21	1143.73	–10.02	4068

That's It

THANK YOU