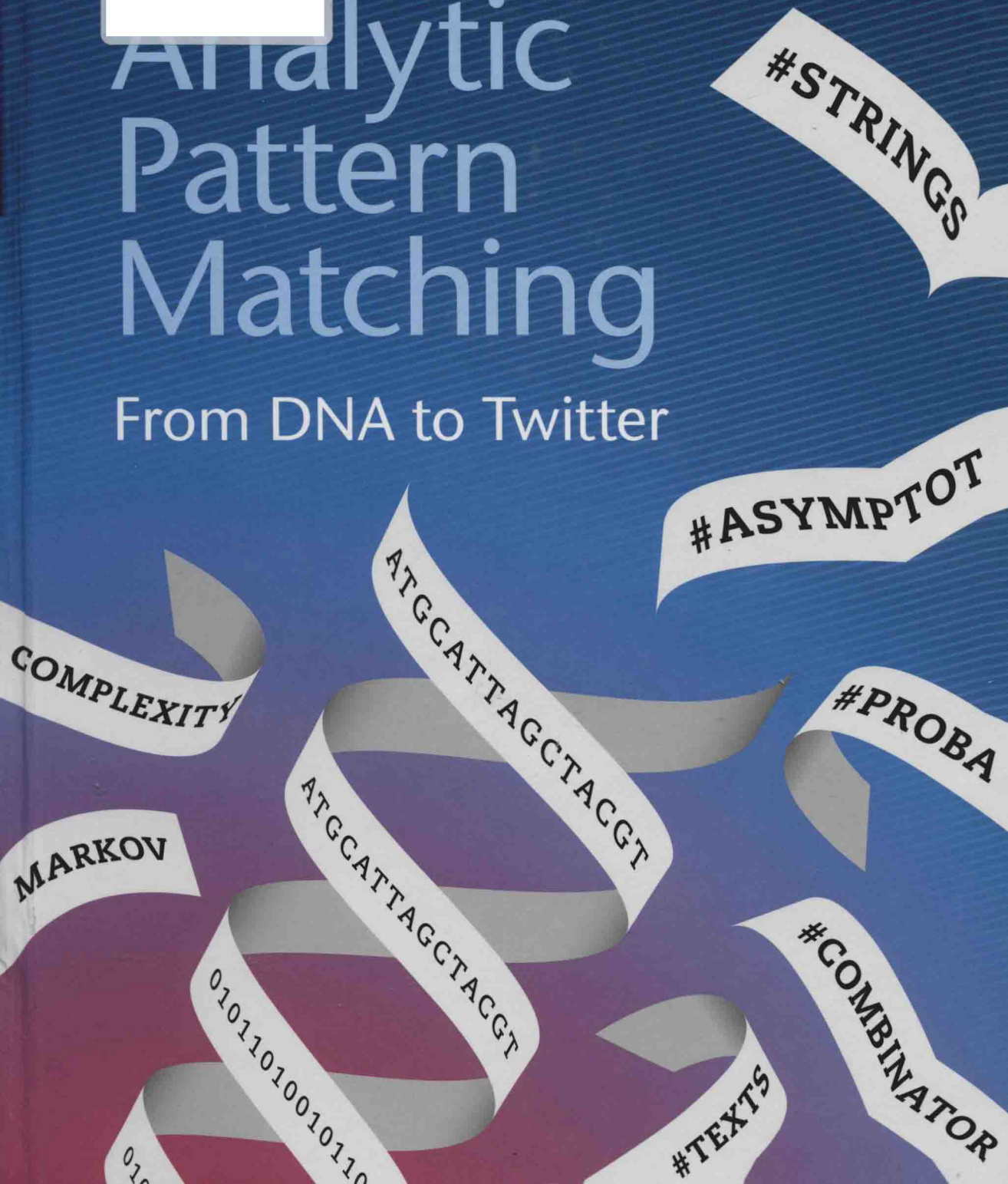
Philippe Jacquet and
Wojciech Szpankowski

Analytic Pattern Matching

From DNA to Twitter



ANALYTIC PATTERN MATCHING

From DNA to Twitter

PHILIPPE JACQUET
Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt

WOJCIECH SZPANKOWSKI
Purdue University, Indiana

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9780521876087

First published 2015

Printed in the United Kingdom by Clays, St Ives plc

A catalog record for this publication is available from the British Library

Library of Congress Cataloging in Publication data

Jacquet, Philippe, 1958–
Analytic pattern matching from DNA to Twitter / Philippe Jacquet, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt Wojciech Szpankowski, Purdue University, Indiana.
pages cm
Includes bibliographical references and index.
ISBN 978-0-521-87608-7 (Hardback)
1. Pattern recognition systems. I. Szpankowski, Wojciech, 1952– II. Title.
TK7882.P3J33 2015
519.2–dc23
2014017256

ISBN 978-0-521-87608-7 Hardback

ANALYTIC PATTERN MATCHING

How do you distinguish a cat from a dog by their DNA? Did Shakespeare really write all his plays? Pattern matching techniques can offer answers to these questions and to many others, in contexts from molecular biology to telecommunications to the classification of Twitter content.

This book, intended 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 results for pattern matching problems that can be obtained 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 will help the reader to make the link between theory and practice.

Dedicated to PHILIPPE FLAJOLET,
our mentor and friend.

Foreword

Early computers replaced calculators and typewriters, and programmers focused on scientific computing (calculations involving numbers) and string processing (manipulating sequences of alphanumeric characters, or strings). Ironically, in modern applications, string processing is an integral part of scientific computing, as strings are an appropriate model of the natural world in a wide range of applications, notably computational biology and chemistry. Beyond scientific applications, strings are the *lingua franca* of modern computing, with billions of computers having immediate access to an almost unimaginable number of strings.

Decades of research have met the challenge of developing fundamental algorithms for string processing and mathematical models for strings and string processing that are suitable for scientific studies. Until now, much of this knowledge has been the province of specialists, requiring intimate familiarity with the research literature. The appearance of this new book is therefore a welcome development. It is a unique resource that provides a thorough coverage of the field and serves as a guide to the research literature. It is worthy of serious study by any scientist facing the daunting prospect of making sense of huge numbers of strings.

The development of an understanding of strings and string processing algorithms has paralleled the emergence of the field of analytic combinatorics, under the leadership of the late Philippe Flajolet, to whom this book is dedicated. Analytic combinatorics provides powerful tools that can synthesize and simplify classical derivations and new results in the analysis of strings and string processing algorithms. As disciples of Flajolet and leaders in the field nearly since its inception, Philippe Jacquet and Wojciech Szpankowski are well positioned to provide a cohesive modern treatment, and they have done a masterful job in this volume.

Robert Sedgewick
Princeton University

Preface

Repeated patterns and related phenomena in words are known to play a central role in many facets of computer science, telecommunications, coding, data compression, data mining, and molecular biology. One of the most fundamental questions arising in such studies is the frequency of pattern occurrences in a given string known as the text. Applications of these results include gene finding in biology, executing and analyzing tree-like protocols for multiaccess systems, discovering repeated strings in Lempel–Ziv schemes and other data compression algorithms, evaluating string complexity and its randomness, synchronization codes, user searching in wireless communications, and detecting the signatures of an attacker in intrusion detection.

The basic *pattern matching* problem is to find for a given (or random) pattern w or set of patterns $\mathcal{W}$ and a text X how many times $\mathcal{W}$ occurs in the text X and how long it takes for $\mathcal{W}$ to occur in X for the first time. There are many variations of this basic pattern matching setting which is known as *exact string matching*. In approximate string matching, better known as *generalized string matching*, certain words from $\mathcal{W}$ are expected to occur in the text while other words are *forbidden* and cannot appear in the text. In some applications, especially in constrained coding and neural data spikes, one puts restrictions on the text (e.g., only text without the patterns 000 and 0000 is permissible), leading to *constrained string matching*. Finally, in the most general case, patterns from the set $\mathcal{W}$ do not need to occur as strings (i.e., consecutively) but rather as subsequences; that leads to *subsequence pattern matching*, also known as *hidden pattern matching*.

These various pattern matching problems find a myriad of applications. Molecular biology provides an important source of applications of pattern matching, be it exact or approximate or subsequence pattern matching. There are examples in abundance: finding signals in DNA; finding split genes where exons are interrupted by introns; searching for starting and stopping signals in genes; finding tandem repeats in DNA. In general, for gene searching, hidden pattern matching (perhaps with an exotic constraint set) is the right approach for find-

ing meaningful information. The hidden pattern problem can also be viewed as a close relative of the longest common subsequence (LCS) problem, itself of immediate relevance to computational biology but whose probabilistic aspects are still surrounded by mystery.

Exact and approximate pattern matching have been used over the last 30 years in source coding (better known as data compression), notably in the Lempel–Ziv schemes. The idea behind these schemes is quite simple: when an encoder finds two (longest) copies of a substring in a text to be compressed, the second copy is not stored but, rather, one retains a pointer to the copy (and possibly the length of the substring). The holy grail of universal source coding is to show that, without knowing the statistics of the text, such schemes are asymptotically optimal.

There are many other applications of pattern matching. Prediction is one of them and is closely related to the Lempel–Ziv schemes (see Jacquet, Szpankowski, and Apostol (2002) and Vitter and Krishnan (1996)). Knowledge discovery can be achieved by detecting repeated patterns (e.g., in weather prediction, stock market, social sciences). In data mining, pattern matching algorithms are probably the algorithms most often used. A text editor equipped with a pattern matching predictor can guess in advance the words that one wants to type. Messages in phones also use this feature.

In this book we study pattern matching problems in a probabilistic framework in which the text is generated by a probabilistic source while the pattern is given. In Chapter 1 various probabilistic sources are discussed and our assumptions are summarized. In Chapter 6 we briefly discuss the algorithmic aspects of pattern matching and various efficient algorithms for finding patterns, while in the rest of this book we focus on *analysis*. We apply analytic tools of combinatorics and the analysis of algorithms to discover general laws of pattern occurrences. Tools of analytic combinatorics and analysis of algorithms are well covered in recent books by Flajolet and Sedgewick (2009) and Szpankowski (2001).

The approach advocated in this book is the analysis of pattern matching problems through a formal description by means of regular languages. Basically, such a description of the *contexts* of one, two, or more occurrences of a pattern gives access to the expectation, the variance, and higher moments, respectively. A systematic translation into the *generating functions* of a complex variable is available by methods of analytic combinatorics deriving from the original Chomsky–Schützenberger theorem. The structure of the implied generating functions at a pole or algebraic singularity provides the necessary asymptotic information. In fact, there is an important phenomenon, that of *asymptotic simplification*, in which the essentials of combinatorial-probabilistic features are reflected by the singular forms of generating functions. For instance,

variance coefficients come out naturally from this approach, together with a suitable notion of correlation. Perhaps the originality of the present approach lies in this joint use of combinatorial-enumerative techniques and analytic-probabilistic methods.

We should point out that pattern matching, hidden words, and hidden meaning were studied by many people in different contexts for a long time before computer algorithms were designed. Rabbi Akiva in the first century A.D. wrote a collection of documents called *Maaseh Merkava* on secret mysticism and meditations. In the eleventh century the Spaniard Solomon Ibn Gabirol called these secret teachings *Kabbalah*. Kabbalists organized themselves as a secret society dedicated to the study of the ancient wisdom of Torah, looking for mysterious connections and hidden truths, meaning, and words in Kabbalah and elsewhere. Recent versions of this activity are *knowledge discovery and data mining, bibliographic search, lexicographic research, textual data processing*, and even *web site indexing*. Public domain utilities such as `agrep`, `grappe`, and `webglimpse` (developed for example by Wu and Manber (1995), Kucherov and Rusinowitch (1997), and others) depend crucially on approximate pattern matching algorithms for subsequence detection. Many interesting algorithms based on regular expressions and automata, dynamic programming, directed acyclic word graphs, and digital tries or suffix trees have been developed. In all the contexts mentioned above it is of obvious interest to distinguish pattern occurrences from the statistically unavoidable phenomenon of noise. The results and techniques of this book may provide some answers to precisely these questions.

Contents of the book

This book has two parts. In Part I we compile all the results known to us about the various pattern matching problems that have been tackled by analytic methods. Part II is dedicated to the application of pattern matching to various data structures on words, such as digital trees (e.g., tries and digital search trees), suffix trees, string complexity, and string-based data compression and includes the popular schemes of Lempel and Ziv, namely the Lempel–Ziv'77 and the Lempel–Ziv'78 algorithms. When analyzing these data structures and algorithms we use results and techniques from Part I, but we also bring to the table new methodologies such as the Mellin transform and analytic depoissonization.

As already discussed, there are various pattern matching problems. In its simplest form, a pattern $\mathcal{W} = w$ is a single string w and one searches for some or all occurrences of w as a block of consecutive symbols in the text. This problem is known as *exact string matching* and its analysis is presented in Chapter 2 where we adopt a symbolic approach. We first describe a language that contains all occurrences of w. Then we translate this language into a generating function

that will lead to precise evaluation of the mean and variance of the number of occurrences of the pattern. Finally, we establish the central and local limit laws, and large deviation results.

In Chapter 2 we assume that the text is generated by a random source without any constraints. However, in several important applications in coding and molecular biology, often the text itself must satisfy some restrictions. For example, codes for magnetic recording cannot have too many consecutive zeros. This leads to consideration of the so-called (d, k) sequences, in which runs of zeros are of length at least d and at most k. In Chapter 3 we consider the exact string matching problem when the text satisfies extra constraints, and we coin the term *constrained pattern matching*. We derive moments for the number of occurrences as well as the central limit laws and large deviation results.

In the *generalized string matching* problem discussed in Chapter 4 the pattern $\mathcal{W}$ is a set of patterns rather than a single pattern. In its most general formulation, the pattern is a pair $(\mathcal{W}_0, \mathcal{W})$ where $\mathcal{W}_0$ is the so-called *forbidden set*. If $\mathcal{W}_0 = \emptyset$ then $\mathcal{W}$ is said to appear in the text X whenever a word from $\mathcal{W}$ occurs as a string, with overlapping allowed. When $\mathcal{W}_0 \neq \emptyset$ one studies the number of occurrences of strings from $\mathcal{W}$ under the condition that there is no occurrence of a string from $\mathcal{W}_0$ in the text. This is *constrained* string matching, since one restricts the text to those strings that do not contain strings from $\mathcal{W}_0$. Setting $\mathcal{W} = \emptyset$ (with $\mathcal{W}_0 \neq \emptyset$), we search for the number of text strings that do not contain any pattern from $\mathcal{W}_0$. In this chapter we present a complete analysis of the generalized string matching problem. We first consider the so-called *reduced set of patterns* in which one string in $\mathcal{W}$ cannot be a substring of another string in $\mathcal{W}$. We generalize our combinatorial language approach from Chapter 2 to derive the mean, variance, central and local limit laws, and large deviation results. Then we analyze the generalized string pattern matching. In our first approach we construct an automaton to recognize a pattern $\mathcal{W}$, which turns out to be a de Bruijn graph. The generating function of the number of occurrences has a matrix form; the main matrix represents the transition matrix of the associated de Bruijn graph. Our second approach is a direct generalization of the language approach from Chapter 2. This approach was recently proposed by Bassino, Clement, and Nicodeme (2012).

In the last chapter of Part I, Chapter 5, we discuss another pattern matching problem called subsequence pattern matching or hidden pattern matching. In this case the pattern $\mathcal{W} = w_1 a_2 \cdots w_m$, where w_i is a symbol of the underlying alphabet, occurs as a subsequence rather than a string of consecutive symbols in a text. We say that $\mathcal{W}$ is hidden in the text. For example, `date` occurs as a subsequence in the text `hidden pattern`, four times in fact but not even once as a string. The gaps between the occurrences of $\mathcal{W}$ may be bounded or unrestricted. The extreme cases are: the *fully unconstrained* problem, in

which all gaps are unbounded, and the *fully constrained* problem, in which all gaps are bounded. We analyze these and mixed cases. Also, in Chapter 5 we present a general model that contains all the previously discussed pattern matchings. In short, we analyze the so-called *generalized subsequence problem.* In this case the pattern is $\mathcal{W} = (\mathcal{W}_1, \ldots, \mathcal{W}_d)$, where $\mathcal{W}_i$ is a collection of strings (a language). We say that the generalized pattern $\mathcal{W}$ occurs in the text X if X contains $\mathcal{W}$ as a *subsequence* $(w_1, w_2, \ldots, w_d)$, where $w_i \in \mathcal{W}_i$. Clearly, the generalized subsequence problem includes all the problems discussed so far. We analyze this generalized pattern matching for general probabilistic dynamic sources, which include Markov sources and mixing sources as recently proposed by Vallée (2001). The novelty of the analysis lies in the translation of probabilities into compositions of operators. Under a mild decomposability assumption, these operators possess spectral representations that allow us to derive precise asymptotic behavior for the quantities of interest.

Part II of the book starts with Chapter 6, in which we describe some data structures on words. In particular, we discuss digital trees, suffix trees, and the two most popular data compression schemes, namely Lempel–Ziv'77 (LZ'77) and Lempel–Ziv'78 (LZ'78).

In Chapter 7 we analyze tries and digital search trees built from *independent* strings. These basic digital trees owing to their simplicity and efficiency, find widespread use in diverse applications ranging from document taxonomy to IP address lookup, from data compression to dynamic hashing, from partial-match queries to speech recognition, from leader election algorithms to distributed hashing tables. We study analytically several tries and digital search tree parameters such as depth, path length, size, and average profile. The motivation for studying these parameters is multifold. First, they are efficient shape measures characterizing these trees. Second, they are asymptotically close to the parameters of suffix trees discussed in Chapter 8. Third, not only are the analytical problems mathematically challenging, but the diverse new phenomena they exhibit are highly interesting and unusual.

In Chapter 8 we continue analyzing digital trees but now those built from correlated strings. Namely we study suffix trees, which are tries constructed from the suffixes of a string. In particular we focus on characterizing mathematically the length of the longest substring of the text occurring at a given position that has another copy in the text. This length, when averaged over all possible positions of the text, is actually the typical *depth* in a suffix trie built over (randomly generated) text. We analyze it using analytic techniques such as generating functions and the Mellin transform. More importantly, we reduce its analysis to the exact pattern matching discussed in depth in Chapter 2. In fact, we prove that the probability generating function of the depth in a suffix trie is asymptotically close to the probability generating function of the depth in a

trie that is built over n *independently* generated texts analyzed in Chapter 7, so we have a pretty good understanding of its probabilistic behavior. This allows us to conclude that the depth in a suffix trie is asymptotically normal. Finally, we turn our attention to an application of suffix trees to the analysis of the Lempel–Ziv'77 scheme. We ask the question how many LZ'77 phrases there are in a randomly generated string. This number is known as the multiplicity parameter and we establish its asymptotic distribution.

In Chapter 9 we study a data structure that is the most popular and the hardest to analyze, namely the Lempel–Ziv'78 scheme. Our goal is to characterize probabilistically the number of LZ'78 phrases and its redundancy. Both these tasks drew a lot of attention as being open and difficult until Aldous and Shields (1988) and Jacquet and Szpankowski (1995) solved them for memoryless sources. We present here a simplified proof for extended results: the central limit theorem for the number of phrases and the redundancy as well as the moderate and large deviation findings. We study this problem by reducing it to an analysis of the associated digital search tree, already discussed in part in Chapter 7. In particular, we establish the central limit theorem and large deviations for the total path length in the digital search tree.

Finally, in Chapter 10 we study the string complexity and also the joint string complexity, which is defined as the cardinality of distinct subwords of a string or strings. The string complexity captures the "richness of the language" used in a sequence, and it has been studied quite extensively from the worst case point of view. It has also turned out that the joint string complexity can be used quite successfully for `twitter` classification (see Jacquet, Milioris, and Szpankowski (2013)). In this chapter we focus on an average case analysis. The joint string complexity is particularly interesting and challenging from the analysis point of view. It requires novel analytic tools such as the two-dimensional Mellin transform, depoissonization, and the saddle point method.

Nearly every chapter is accompanied by a set of problems and related bibliography. In the problem sections we ask the reader to complete a sketchy proof, to solve a similar problem, or to actually work on an open problem. In the bibliographical sections we briefly describe some related literature.

Finally, to ease the reading of this book, we illustrate each chapter with an original comic sketch. Each sketch is somewhat related to the topic of the corresponding chapter, as discussed below.

Acknowledgments

This book is dedicated to **Philippe Flajolet**, the father of analytic combinatorics, our friend and mentor who passed away suddenly on March 22, 2011. An obituary – from which we freely borrow here – was recently published by Salvy, Sedgewick, Soria, Szpankowski, and Vallee (2011).

We believe that this book was possible thanks to the tireless efforts of Philippe Flajolet, his extensive and far-reaching body of work, and his scientific approach to the study of algorithms, including the development of the requisite mathematical and computational tools. Philippe is best known for his fundamental advances in mathematical methods for the analysis of algorithms; his research also opened new avenues in various areas of applied computer science, including streaming algorithms, communication protocols, database access methods, data mining, symbolic manipulation, text-processing algorithms, and random generation. He exulted in sharing his passion: his papers had more than a hundred different co-authors (including the present authors) and he was a regular presence at scientific meetings all over the world.

Philippe Flajolet's research laid the foundation of a subfield of mathematics now known as analytic combinatorics. His lifework *Analytic Combinatorics* (Cambridge University Press, 2009, co-authored with R. Sedgewick) is a prodigious achievement, which now defines the field and is already recognized as an authoritative reference. Analytic combinatorics is a modern basis for the quantitative study of combinatorial structures (such as words, trees, mappings, and graphs), with applications to probabilistic study of algorithms based on these structures. It also strongly influences other scientific areas, such as statistical physics, computational biology, and information theory. With deep historic roots in classical analysis, the basis of the field lies in the work of Knuth, who put the study of algorithms onto a firm scientific basis, starting in the late 1960s with his classic series of books. Philippe Flajolet's work took the field forward by introducing original approaches into combinatorics based on two types of methods: symbolic and analytic. The symbolic side is based on the automation of decision procedures in combinatorial enumeration to derive characterizations

of generating functions. The analytic side treats those generating functions as functions in the complex plane and leads to a precise characterization of limit distributions.

Finally, Philippe Flajolet was the leading figure in the development of a large international community (which again includes the present authors) devoted to research on probabilistic, combinatorial, and asymptotic methods in the analysis of algorithms. His legacy is alive through this community. We are still trying to cope with the loss of our friend and mentor.

While putting the final touches to this book, the tragic shooting occurred in Paris of the famous French cartoonists at Charlie Hebdo. We are sure that this event would shock Philippe Flajolet, who had an inimitable sense of humour. He mocked himself and his friends. We were often on the receiving end of his humour. We took it, as most did, as a proof of affection. Cartoonists and scientists have something in common: offending the apparent truth is a necessary step toward better knowledge.

There is a long list of other colleagues and friends from whom we benefited through their encouragement and constructive comments. They have helped us in various ways during our work on the analysis of algorithms and information theory. We mention here only a few: Alberto Apostolico, Yongwook Choi, Luc Devroye, Michael Drmota, Ananth Grama, H-K. Hwang, Svante Janson, John Kieffer, Chuck Knessl, Yiannis Kontoyiannis, Mehmet Koyuturk, Guy Louchard, Stefano Lonardi, Pierre Nicodeme, Ralph Neininger, Gahyuan Park, Mireille Régnier, Yuriy Reznik, Bob Sedgewick, Gadiel Seroussi, Brigitte Vallée, Sergio Verdu, Mark Ward, Marcelo Weinberger. We thank Bob Sedgewick for writing the foreword.

Finally, no big project like this can be completed without help from our families. We thank Véronique, Manou, Lili, Mariola, Lukasz and Paulina from the bottom of our hearts.

This book has been written over the last five years, while we have been traveling around the world carrying (electronically) numerous copies of various drafts. We have received invaluable help from the staff and faculty of INRIA, France and the Department of Computer Sciences at Purdue University. The second author is grateful to the National Science Foundation, which has supported his work over the last 30 years. We have also received support from institutions around the world that have hosted us and our book: INRIA, Rocquencourt; Alcatel-Lucent, Paris; LINCS, Paris; the Gdańsk University of Technology, Gdańsk; the University of Hawaii; and ETH, Zurich. We thank them very much.

Philippe Jacquet
Wojtek Szpankowski