

Computational Biology 33

Hannes Hauswedell

Sequence Analysis and Modern C++

The Creation of the SeqAn3
Bioinformatics Library



Computational Biology

Volume 33

Advisory Editors:

Gordon Crippen, University of Michigan, Ann Arbor, MI, USA
Joseph Felsenstein, University of Washington, Seattle, WA, USA
Dan Gusfield, University of California, Davis, CA, USA
Sorin Istrail, Brown University, Providence, RI, USA
Thomas Lengauer, Max Planck Institute for Computer Science, Saarbrücken, Germany
Marcella McClure, Montana State University, Bozeman, MT, USA
Martin Nowak, Harvard University, Cambridge, MA, USA
David Sankoff, University of Ottawa, Ottawa, ON, Canada
Ron Shamir, Tel Aviv University, Tel Aviv, Israel
Mike Steel, University of Canterbury, Christchurch, New Zealand
Gary Stormo, Washington University in St. Louis, St. Louis, MO, USA
Simon Tavaré, University of Cambridge, Cambridge, UK
Tandy Warnow, University of Illinois at Urbana-Champaign, Urbana, IL, USA
Lonnie Welch, Ohio University, Athens, OH, USA

Editors-in-Chief:

Andreas Dress, CAS-MPG Partner Institute for Computational Biology, Shanghai, China
Michal Linial, Hebrew University of Jerusalem, Jerusalem, Israel
Olga Troyanskaya, Princeton University, Princeton, NJ, USA
Martin Vingron, Max Planck Institute for Molecular Genetics, Berlin, Germany

Editorial Board Members:

Robert Giegerich, University of Bielefeld, Bielefeld, Germany
Janet Kelso, Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany
Gene Myers, Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
Pavel Pevzner, University of California, San Diego, CA, USA

Endorsed by the *International Society for Computational Biology*, the *Computational Biology* series publishes the very latest, high-quality research devoted to specific issues in computer-assisted analysis of biological data. The main emphasis is on current scientific developments and innovative techniques in computational biology (bioinformatics), bringing to light methods from mathematics, statistics and computer science that directly address biological problems currently under investigation.

The series offers publications that present the state-of-the-art regarding the problems in question; show computational biology/bioinformatics methods at work; and finally discuss anticipated demands regarding developments in future methodology. Titles can range from focused monographs, to undergraduate and graduate textbooks, and professional text/reference works.

More information about this series at <https://link.springer.com/bookseries/5769>

Hannes Hauswedell

Sequence Analysis and Modern C++

The Creation of the SeqAn3 Bioinformatics
Library



Hannes Hauswedell 
Reykjavik
Iceland

ISSN 1568-2684

ISSN 2662-2432 (electronic)

Computational Biology

ISBN 978-3-030-90989-5

ISBN 978-3-030-90990-1 (eBook)

<https://doi.org/10.1007/978-3-030-90990-1>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022, corrected publication 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This is a book about software engineering, bioinformatics, the C++ programming language and the SeqAn library. In the broadest sense, it will help the reader create better, faster and more reliable software by deepening their understanding of available tools, language features, techniques and design patterns.

Every developer who previously worked with C++ will enjoy the in-depth chapter on important changes in the language from C++ 11 up to and including C++ 20. In contrast to many resources on Modern C++ that present new features only in small isolated examples, this book represents a more holistic approach: readers will understand the relevance of new features and how they interact in the context of a large software project and not just within a “toy example”. Previous experience in creating software with C++ is highly recommended to fully appreciate these aspects.

SeqAn3 is a new, re-designed software library. The conception and implementation process is detailed in this book, including a critical reflection on the previous versions of the library. This is particularly helpful to readers who are about to create a large software project themselves, or who are planning a major overhaul of an existing library or framework. While the focus of the book is clearly on software development and design, it also touches on various organisational and administrative aspects like licensing, dependency management and quality control.

The field that SeqAn3 provides solutions for is sequence analysis or, in a broader sense, bioinformatics. Readers working in this domain will recognise many of the discussed problems. However, almost all content is useful to software engineers in general and research software engineers in particular; no background in biology or previous experience with the SeqAn library is required.

This book is based on a dissertation, so the general style is more reminiscent of a “story” than might be typical for a computer science book. Some readers will enjoy reading it cover to cover while others will want to jump to sections of interest directly. The original preface of the dissertation is given on the following page as

the acknowledgements section. In addition to the persons mentioned there, I would like to thank Martin Vingron who was part of my defence committee and suggested this book project. I would also like to thank Susan Evans and the team at Springer Nature for helping it become reality.

Reykjavik, Iceland

Hannes Hauswedell

Acknowledgements

The SeqAn library is a very active project with a long history. Over the last more than 10 years, it has had different core developers and many people who contributed features and fixes. Although SeqAn3 contains almost no code from SeqAn1/2, the experience of working on and with previous versions was invaluable in the development of SeqAn3. I feel that it is therefore only proper to mention Andreas Gogol-Döring, David Weese, Enrico Siragasu and Manuel Holtgrewe at this point, all of whom contributed significantly to SeqAn1/2. Of course Knut Reinert has always guided and does until today lead the project. His experience is the main pillar of its continued success.

This thesis introduces a new and radically different version of the SeqAn library. The scope of this project is huge, and it certainly would not have been possible to create the library single-handedly in this time. I do, however, credit myself with its inception, the vision behind the project and the endurance to pursue a complete rewrite of the library when most people called it infeasible. The design process, the overarching goals and the technical decisions are overwhelmingly my work—that is the foundation of this thesis. On the practical side, I have also written and changed more code than the next most important contributors combined, but I want to state clearly that relevant parts of SeqAn3 have also been implemented by people other than myself.

René Rahn has shared the responsibility of leading the project with me on a social and administrative level. Since the early beginnings of SeqAn3, I relied strongly on his counsel. Later, we assembled the SeqAn *core team* to discuss design and strategy matters on a regular basis. This included Svenja Mehringer, Marcel Ehrhardt and Enrico Seiler. All members of the core team have left their mark in some way on the library, and I am confident that SeqAn3 is in good hands after I leave the project.

I would like to thank everyone who contributed to SeqAn3, but more generally I want to also thank everyone for the great time at Freie Universität and the unforgettable SeqAn retreats! Special thanks go to Sara Hetzel and Felix Heeger who provided very helpful comments on a draft of this dissertation. Sara will also continue work on Lambda, an application presented later in this thesis.

On a professional and personal level, my sincere gratitude goes to Knut Reinert who has been my mentor now for so many years. None of this would have been possible without him. I would also like to express my sincere gratitude to Stefan Kurtz who agreed to co-supervise this (quite comprehensive) thesis although we had not worked together previously.

Attending the meetings of and contributing to the ISO C++ committee has had the most profound influence on my understanding of C++ and has thus helped greatly with creating SeqAn3. I would like to thank Fabio Fracassi and Nico Josuttis from the DIN Arbeitskreis Programmiersprachen as well as Corentin Jabot and JeanHeyd Meneide for helping me find my way around WG21.

Before working at Freie Universität, my studies were funded through a stipend of the Max-Planck-Gesellschaft. I additionally received a fellowship by the Hans-Böckler-Stiftung which allowed me to attend various extracurricular activities, for which I am very grateful.

Finally, I would like to thank my parents for supporting me during my youth and my early university studies. I am privileged to have had access to computers as a child and to grow up in an environment that fostered my curiosity in science and technology. I am grateful for the support of my friends and especially Romy and Betti. I look forward to spending more time with everyone again!

Contents

Part I Background

1	Sequence Analysis	3
2	The SeqAn Library (Versions 1 and 2)	7
2.1	History	7
2.2	Design Goals	8
2.3	Programming Techniques	8
2.3.1	Generic Programming	9
2.3.2	Template Subclassing	9
2.3.3	Global Function Interfaces	11
2.3.4	Metafunctions	12
2.4	Discussion	14
2.4.1	Performance	17
2.4.2	Simplicity	19
2.4.3	Generality, Refineability and Extensibility	24
2.4.4	Integration	25
2.4.5	Summary	31
3	Modern C++	33
3.1	Type Deduction	35
3.1.1	The <code>auto</code> Specifier	36
3.1.2	Class Template Argument Deduction (CTAD)	39
3.2	Move Semantics and Perfect Forwarding	39
3.2.1	Move Semantics	40
3.2.2	Reference Types and Perfect Forwarding	42
3.2.3	Out-Parameters and Returning by Value	43
3.3	Metaprogramming and Compile-Time Computations	45
3.3.1	Metafunctions and Type Traits	45
3.3.2	Traits Classes	46
3.3.3	Compile-Time Computations	48

3.3.4	Conditional Instantiation	50
3.3.5	Standard Library Traits	51
3.4	C++ Concepts	52
3.4.1	Introduction	53
3.4.2	Defining Concepts	54
3.4.3	Using Concepts	55
3.4.4	Concepts-Based Polymorphism	57
3.4.5	Standard Library Concepts	59
3.5	Code Reuse	59
3.5.1	The Curiously Recurring Template Pattern (CRTP)	60
3.5.2	Metaclasses	62
3.6	C++ Ranges	63
3.6.1	Introduction	63
3.6.2	Range Traits and Concepts	65
3.6.3	The View Concept	69
3.6.4	Range Adaptor Objects	71
3.6.5	Standard Library Views	74
3.7	Customisation Points	76
3.7.1	Excursus: Calling Conventions	76
3.7.2	Introduction	77
3.7.3	“Niebloids”	79
3.7.4	Future Standardisation	82
3.8	Concurrency & Parallelism	82
3.9	C++ Modules	83
3.10	Utility Types	84
3.11	Discussion	85

Part II SeqAn3

4	The Design of SeqAn3	89
4.1	Design Goals	89
4.1.1	Performance	90
4.1.2	Simplicity	90
4.1.3	Integration	91
4.1.4	Adaptability	92
4.1.5	Compactness	92
4.2	Programming Techniques	93
4.2.1	Modern C++	93
4.2.2	Programming Paradigms	94
4.2.3	Polymorphism and Customisation	95
4.2.4	Aspects of Object-Orientation	96
4.2.5	Ranges and Views	97
4.2.6	“Natural” Function Interfaces	98
4.2.7	constexpr if Possible	98

4.3	Administrative Aspects	99
4.3.1	Header-Only Library	99
4.3.2	Licence	100
4.3.3	Platform Support	100
4.3.4	Stability	103
4.3.5	Availability	106
4.3.6	Combining SeqAn2 and SeqAn3	107
4.4	Dependencies and Tooling	107
4.4.1	Library Dependencies	108
4.4.2	Documentation	113
4.4.3	Testing	116
4.5	Project Management and Social Aspects	122
5	Library Structure and Small Modules	125
5.1	Library Structure	126
5.1.1	Files and Directories	126
5.1.2	Modules and Submodules	126
5.1.3	Names and Namespaces	128
5.2	“Small” Modules	129
5.2.1	Argument Parser	129
5.2.2	The Core Module	131
5.2.3	The Utility Module	133
5.2.4	The STD Module	137
5.2.5	The Contrib Module	138
5.3	Discussion	139
5.3.1	Performance	139
5.3.2	Simplicity	140
5.3.3	Integration	142
5.3.4	Adaptability	142
5.3.5	Compactness	142
6	The Alphabet Module	145
6.1	General Design	146
6.1.1	Character and Rank Representation	147
6.1.2	Function Objects and Traits	149
6.1.3	Concepts	152
6.2	User-Defined Alphabets and Adaptations	155
6.2.1	User-Defined Alphabets	157
6.2.2	Adapting Existing Types as Alphabets	159
6.3	The Nucleotide Submodule	161
6.3.1	General Design	162
6.3.2	Canonical DNA Alphabets	164
6.3.3	Canonical RNA Alphabets	165
6.3.4	Other Nucleotide Alphabets	166

- 6.4 The Amino Acid Submodule 166
 - 6.4.1 General Design 167
 - 6.4.2 Amino Acid Alphabets 168
 - 6.4.3 Translation 169
- 6.5 Composite Alphabets 170
 - 6.5.1 Alphabet Variants 170
 - 6.5.2 Alphabet Tuples 173
 - 6.5.3 Alphabet “any” Types 176
- 6.6 The Quality Submodule 178
 - 6.6.1 General Design 179
 - 6.6.2 Quality Alphabets 179
 - 6.6.3 Quality Tuples 180
- 6.7 Discussion 182
 - 6.7.1 Performance 183
 - 6.7.2 Simplicity 184
 - 6.7.3 Integration 185
 - 6.7.4 Adaptability 185
 - 6.7.5 Compactness 185
- 7 The Range Module 187**
 - 7.1 General Design 187
 - 7.2 Container 189
 - 7.2.1 Concepts 189
 - 7.2.2 Bit-Compressed Container 191
 - 7.2.3 Containers of Containers 191
 - 7.2.4 Fixed-Capacity Containers 192
 - 7.3 Views 194
 - 7.3.1 General Design 194
 - 7.3.2 Alphabet-Specific Views 200
 - 7.3.3 Some General-Purpose Views 202
 - 7.3.4 Implementation Notes 203
 - 7.4 Discussion 207
 - 7.4.1 Performance 208
 - 7.4.2 Simplicity 214
 - 7.4.3 Integration 216
 - 7.4.4 Adaptability 217
 - 7.4.5 Compactness 217
- 8 The Input/Output Module 219**
 - 8.1 The Stream Submodule 220
 - 8.2 Serialisation 221
 - 8.3 Formatted Files 222
 - 8.3.1 Files and Formats 223
 - 8.3.2 Records and Fields 224
 - 8.4 The Sequence File Submodule 225
 - 8.4.1 Input 226
 - 8.4.2 Output 228

8.4.3	Combined Input and Output	229
8.4.4	Asynchronous Input/Output	230
8.5	Discussion	232
8.5.1	Performance	232
8.5.2	Simplicity	235
8.5.3	Integration	238
8.5.4	Adaptability	239
8.5.5	Compactness	240
9	The Search Module	243
9.1	The FM-Index Submodule	244
9.1.1	Unidirectional FM-Index	245
9.1.2	Bidirectional FM-Index	248
9.2	The k -Mer-Index Submodule	248
9.2.1	Shapes in SeqAn3	249
9.3	General Algorithm Design	252
9.4	The (Search) Algorithm Submodule	253
9.4.1	Search Strategies	255
9.5	The Configuration Submodule	256
9.5.1	Excursus: Aggregate Initialisation and Designated Initialisers	257
9.5.2	Search Config Elements	258
9.6	Discussion	261
9.6.1	Performance	261
9.6.2	Simplicity	264
9.6.3	Integration and Adaptability	268
9.6.4	Compactness	269
10	The Alignment Module	271
10.1	The Aligned Range Submodule	272
10.1.1	Concepts and Function Objects	273
10.1.2	Gap Decorators	274
10.2	The Scoring Submodule	277
10.2.1	Alphabet Scoring Schemes	278
10.2.2	The Gap (Scoring) Scheme	279
10.3	The Pairwise (Alignment) Submodule	280
10.3.1	Algorithm Interface	280
10.3.2	Alignment Result Type	282
10.3.3	Theoretical Background and Implementation Details	283
10.4	The Configuration Submodule	284
10.5	Discussion	287
10.5.1	Performance	287
10.5.2	Simplicity	289
10.5.3	Integration	292
10.5.4	Adaptability	293
10.5.5	Compactness	294

Part III Lambda

11 Lambda: An Application Built with SeqAn	299
11.1 Introduction	299
11.1.1 Previous Work	301
11.1.2 History of LAMBDA	302
11.2 Implementation	303
11.2.1 Index Creation	304
11.2.2 Search	306
11.3 Results	307
11.3.1 Notable Features	308
11.3.2 Performance	309
11.4 Discussion	313
11.4.1 From SeqAn2 to SeqAn3	314
11.4.2 Algorithmic Choices	316

Part IV Conclusion and Appendix

12 Conclusion	321
Correction to: Sequence Analysis and Modern C++	C1
Appendix A	325
A.1 Notes on Reading This Book	325
A.1.1 References and Hyperlinks	325
A.1.2 How to Read Code Snippets	325
A.2 Software and Hardware Details	328
A.2.1 Benchmarking Environment	328
A.2.2 Helpful Software	329
A.3 Copyright	329
A.3.1 SeqAn Copyright	329
A.4 Longer Code Snippets	331
A.5 Detailed Benchmark Results (Local Aligners)	337
References	339