An efficient C++ library for sequence analysis

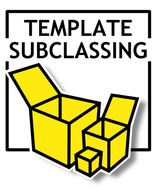
Döring, A., Emde, A.-K., Rausch, T., Reinert, K., Schulz, M., Weese, D.
Algorithmic Bioinformatics, Freie Universität Berlin
www.seqan.de

SeqAn is an open source C++ library of efficient algorithms and data structures for the analysis of biological sequences. Using a template-based library design, SeqAn aims at providing (1) algorithms that are generic, fast and extensible and (2) data structures that allow the rapid design and development of novel sequence analysis methods. The library, documentation and tutorials are available on the web: http://www.seqan.de

DESIGN

"GLOBAL INTERFACES"

- Extensibility: Global functions can be added at any time.
- Flexibility: Algorithms can be specialized for new data types.
- Integration: C++ built-in types can be handled like user-defined types.



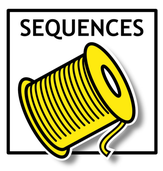
"TEMPLATE SUBCLASSING"

- Class specialization: Use partial specialization of class templates instead of class derivation to refine a given implementation.
- Algorithm specialization: Functions are overloaded for different levels of class template specialization.

"METAFUNCTIONS"

- Metaprogramming: A kind of `code´ that is evaluated at compile time.
- Type traits: Dependent types of a given type can be queried using metafunctions, e.g. the alphabet type of a string.

CONTENT



"SEQUENCES"

- External, memory-mapped, bit-packed, heap and stack allocated strings.
- Sequence modifiers that provide distinct views on a sequence, e.g. an infix or reverse complement view.
- String iterators.

"INDICES"

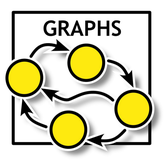
- Enhanced suffix array, lazy suffix tree, in-memory and external indices.
- Suffix tree iterators.
- Gapped and ungapped q-gram indices.
- Algorithms for finding maximal repeats, maximal unique matches and others.

"ALIGNMENTS"

- Alignment data structures.
- Dynamic programming based alignment algorithms.
- Configurable begin-gap and end-gap costs.
- Heuristic, graph-based multiple sequence alignment.
- Fragment chaining.

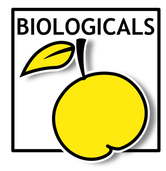
"SEARCHING"

- Exact string matching algorithms, e.g. Horspool, Shift-AND and Shift-OR.
- Approximative string matching algorithms, e.g. Myers bit-vector algorithm and DP algorithms.
- Multiple string matching algorithm, e.g. Wu-Manber and Aho-Corasick.
- Motif search algorithms.



"GRAPHS"

- Directed and undirected graphs, trees, automata and alignment graphs.
- Specialized graphs, e.g. trie, factor oracle or HMM.
- Various graph algorithms, e.g. minimum spanning tree, connected components and many others.



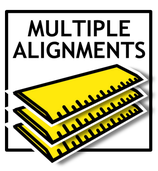
"BIOLOGICALS"

- DNA, RNA and amino acid alphabets.
- PAM, BLOSUM and simple scoring schemes.
- Various file formats, e.g. Fasta and Genbank.
- Modified and profile alphabets.
- Base qualities.

APPLICATIONS

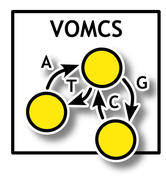
"RAZERS"

- Efficient read mapping with controlled sensitivity.
- Based upon Hamming or edit distance.
- Suitable for both, short-reads from next-generation sequencing and traditional Sanger reads.
- Supports paired-end mapping.



"MULTIPLE ALIGNMENT"

- Segment-based multiple sequence alignment.
- Generic alignments using graphs.
- Suitable for amino acid, DNA, RNA or consensus alignments.
- ReAlignment.



"MARKOV CHAINS"

- Variable order markov chain construction using a lazy suffix tree and an enhanced suffix array.
- Suitable to classify transcription factor binding sites, splice sites and protein families.