Implementing bioinformatics algorithms in TEX — the Gotoh package, a case study

Takuto Asakura

Abstract

TEX is appropriate for implementing many bioinformatics algorithms because they can be programmed with short codes, calculated with a limited range of numbers, and produce visual results. As a case study, I present Gotoh, a LATEX package which implements the Gotoh algorithm, a popular biological sequence alignment algorithm.

1 Motivation

TEX makes for a good programming language to implement many bioinformatics algorithms, such as those for sequence alignment. There are several reasons for this.

First, code for such algorithms tends to be brief. While it is theoretically possible to program even complex algorithms with TEX since it is a Turing machine, it is difficult to write algorithms requiring lengthy source code. Sequence alignment algorithms can be stated with a few lines of recursions.

Secondly, the calculation processes use only a limited range of numbers (usually integers), making it possible to easily store them in TEX's registers. Though the exact limit of what TEX can handle depends on the computing environment, they are usually within the range of what is required by bioinformatics algorithms.

Thirdly, bioinformatics algorithms often build visual output such as charts and strings. These results can be easily incorporated into documents produced by TEX. They can also be utilized as LATEX packages. As LATEX is one of the most widely used front-end systems for typesetting academic papers, it is convenient for researchers if the algorithms that generate contents that go directly into the papers are available as LATEX packages. Users of the packages do not need to execute any commands other than `latex`, and they are freed from the hassles of installing and understanding dedicated tools. It is also possible to link seamlessly with a number of other LATEX packages.

Finally, the stability of the TEX macro language provides for a long-lasting code repository for bioinformatics algorithms. Since the primitives designed by Knuth are extremely stable (Knuth, 1990), an implementation that uses these primitives will continue to function for a long time. However, this might not be necessarily true for implementations using primitives which are available in other engines.

A

1	2	3	4	5
G	A	C	T	A
G	A	.	G	A

B

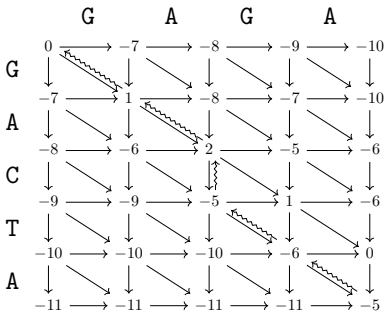

Figure 1: A. An example of pairwise DNA sequence alignment. Here, the third column is a *gap*, the fourth column is a *mismatch* and the others are *matches*.
B. The edit graph corresponding to the matrix H. The squiggly arrows ($\rightsquigarrow$) show the result of trace back.

Here, I consider the Gotoh algorithm, a popular sequence alignment algorithm and implement it in TEX. I also show how to produce publication-ready output by combining my package with other LATEX packages.

2 Sequence alignment

Sequence alignment is often used in bioinformatics to compare the similarity of biological sequences such as DNA, RNA, and amino acid sequences. In the pairwise sequence alignment problem, we are given a pair of sequences

$$A \equiv a_1 a_2 a_3 \dots a_m,\ B \equiv b_1 b_2 b_3 \dots b_n$$

where a_i and b_j are chosen from a finite alphabet, e.g. $\{\mathrm{A}, \mathrm{T}, \mathrm{G}, \mathrm{C}\}$, and the output is a sequence alignment (Figure 1A).

The Longest Common Subsequence (LCS) problem, which is strongly related to the `diff` utility, can be considered as a simple form of sequence alignment in which we score 1 for a *match* and 0 for a *gap*. The optimal score $s_{m,n}$ can be evaluated with the following dynamic programming recursion:

$$s_{i,j} = \max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1. \end{cases}$$

Sequence alignment can be solved by a similar approach, though scoring schemes can be slightly more complex, e.g.

$$\text{match} = c_+,\ \text{mismatch} = c_-,$$
$$g(l) = -d - (l-1)e,$$

where c_+, c_-, d, e are fixed integers, and $g(l)$ is a penalty for an l-length *gap*. One of the most well-known solutions for the problem is the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970; Waterman, Smith, and Beyer, 1976), which calculates the optimal score using the recursion:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + c_{ij} \\ H_{i-k,j} + g(k) \\ H_{i,j-k} + g(k) \end{cases} \quad (1)$$

where

$$c_{ij} = \begin{cases} c_+ & \text{if } a_i = b_j \quad \text{(match)} \\ c_- & \text{otherwise} \quad \text{(mismatch).} \end{cases}$$

After calculating the entries of the dynamic programming matrix H, an optimal alignment can be obtained by trace back of the edit graph (Figure 1B).

3 The Gotoh package

Whereas the Needleman–Wunsch algorithm requires $O(m^2 n)$ time, the Gotoh algorithm (Gotoh, 1982) solves the same problem in $O(mn)$ time. The Gotoh package is an implementation of this algorithm. It is available from CTAN.

3.1 Algorithm

The Gotoh algorithm uses the following formulae transformed from Equation (1):

$$M_{i+1,j+1} = \max\left\{M_{ij}, I^x_{ij}, I^y_{ij}\right\} + c_{ij}$$

where

$$I^x_{i+1,j} = \max\left\{M_{ij} - d, I^x_{ij} - e, I^y_{ij} - d\right\}$$

and

$$I^y_{i,j+1} = \max\left\{M_{ij} - d, I^y_{ij} - e\right\}.$$

An optimal alignment can be obtained by trace back of the three edit graphs corresponding to the matrices M, I^x, I^y. I omit the details.

3.2 Usage and features

The Gotoh package provides two commands: `\Gotoh` for executing the algorithm and `\GotohConfig` for setting parameters with a key–value interface. The package is implemented with only primitives specified by Knuth and some LaTeX macros; it also requires the xkeyval package (Adriaens, 2014).

The usage of `\Gotoh` is simple (Figure 2). This command takes two sequences, assigns the optimal score to `\GotohScore`, and returns the alignment to `\GotohResultA` and `\GotohResultB`. Macros to store the score and results can be changed with the `\GotohConfig` command as follows.

```
\GotohConfig{
  score = \GotohScore,
  result A = \GotohResultA,
  result B = \GotohResultB}
```

A `\Gotoh{`⟨*sequence A*⟩`}{`⟨*sequence B*⟩`}`

B
```
\Gotoh{ATCGGCGCACGGGGGA}{TTCCGCCCACA}
\texttt{\GotohResultA} \\
\texttt{\GotohResultB}
```

C
```
ATCGGCGCACGGGGGA
TTCCGCCCAC.....A
```

Figure 2: Usage of `\Gotoh`. A. Command syntax. B and C. Simple example input and its output. This alignment was calculated with the default parameters of the Gotoh package, which are shown in Equation (2), and the optimal score is −6.

The Gotoh package by default uses the scoring parameters:

$$c_+ = 1,\ c_- = -1,\ d = 7,\ e = 1. \quad (2)$$

They also can be set with `\GotohConfig` as follows.

```
\GotohConfig{
  match = 1, mismatch = -1, d = 7, e = 1}
```

3.3 Collaborating with TEXshade

TEXshade is a LaTeX package designed for typesetting, shading, and labeling preprocessed sequence alignments (Beitz, 2000). This package is also available from CTAN. The Gotoh package can be easily combined with this package.

For example, suppose you define the following macros in the preamble of a LaTeX document.

```
% output file
\newwrite\FASTAfile
\def\writeFASTA#1{%
  \immediate\write\FASTAfile{#1}}

% print alignment
\newcommand{\PrintAlignment}[3][\relax]{%
  \Gotoh{#2}{#3}%
  \immediate\openout\FASTAfile=\jobname.fasta
  \writeFASTA{> Seq 1^^J\GotohResultA}%
  \writeFASTA{> Seq 2^^J\GotohResultB}%
  \immediate\closeout\FASTAfile
  \texshade{\jobname.fasta}#1\endtexshade}
```

At this point, by simply including `\PrintAlignment` in the document, the Gotoh algorithm is executed, the result is formatted by TEXshade, and is output directly in your article (Figure 3).

Note that `\PrintAlignment` communicates between the two packages via a FASTA file, a simple and standard bioinformatics format for recording sequences. This is because TEXshade does not have any user interface to read sequences directly from LaTeX sources (Beitz, 2011). Even so, this macro requires only one `latex` execution.

A `\PrintAlignment[`⟨*TEXshade commands*⟩`]{`⟨*sequence A*⟩`}{`⟨*sequence B*⟩`}`

B

```
seq1       .......GGAGTGAGGGGAGCAGTTGGGCTGAAGATGGTCAACGCCGAGGGAACG    48
seq2       CGCATGCGGAGTGAGGGGAGCAGTTGGG.AACAGATGGTC.CGCCGAGGGACCG    53
consensus  *******!!!!!!!!!!!!!!!!!!!!!!*   !!!!!!!!* !!!!!!!!!! !!


seq1       GTAAAGGCGACGG....AGCTGTGGCAGACCTGGCTTCCTAACCACGTCCCGTGT    99
seq2       GT.GGGCAGACGGGGCCAGCTGTGGCAGACACTGGCTTCTAACCACCGAACGT.T   106
consensus  !!*  !  !!!!!****!!!!!!!!!!!!!   !  ! !!!!!!!!    !!!*!


seq1       TTTGCGGCTCCGCGAGGACTG   120
seq2       CTTTCCGCTCCG.......GG   120
consensus   !! ! !!!!!!******* !
```

Figure 3: Usage of the macro `\PrintAlignment`. A. Command syntax. The first argument ⟨*TEXshade commands*⟩ is optional. B. A sample output.

4 Future directions

It would be conceivable to add to Gotoh a few user interfaces for easier cooperation with other packages that deal with biological sequences. Functional extensions to display more detailed information, such as edit graphs, may also be beneficial. It will be interesting to develop related packages, for instance, one which provides the functionality of multiple-sequence alignments.

Furthermore, it is also interesting to write TEX implementations of algorithms producing visual results to be incorporated into documents. For example, it would be useful if LATEX packages for printing source code such as listings have a `diff` function.

5 Acknowledgements

I would like to thank Shun Sakuraba for his engaging lecture on the Gotoh algorithm which inspired me to develop this package. I am grateful to Anish M. S. Shrestha for helping with the manuscript.

References

Adriaens, Hendri. "The xkeyval package (v2.7a)". `https://ctan.org/pkg/xkeyval`, 2014.

Beitz, Eric. "TEXshade: shading and labeling of multiple sequence alignments using LATEX 2_ε". *Bioinformatics* **16**(2), 135–139, 2000.

Beitz, Eric. "The TEXshade package (v1.24)". `https://ctan.org/pkg/texshade`, 2011.

Gotoh, Osamu. "An improved algorithm for matching biological sequences". *Journal of Molecular Biology* **162**(3), 705–708, 1982.

Knuth, Donald E. "The future of TEX and METAFONT". *TUGboat* **11**(4), 1990. `https://tug.org/TUGboat/tb11-4/tb30knut.pdf`.

Needleman, Saul B., and C. D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of Molecular Biology* **48**(3), 443–453, 1970.

Waterman, Michael S, T. F. Smith, and W. A. Beyer. "Some biological sequence metrics". *Advances in Mathematics* **20**(3), 367–387, 1976.

◇ Takuto Asakura
The University of Tokyo
Department of Bioinformatics and Systems Biology
2-11-16 Yayoi
Bunkyo, Tokyo, 113-0032
Japan
`tkt.asakura (at) gmail dot com`