A note on the shortest common superstring of NGS reads

Tristan Braquelaire∗ Marie Gasparoux∗ Mathieu Raffinot†
Raluca Uricaru∗

May 19, 2016

Abstract

The Shortest Superstring Problem (SSP) consists, for a set of strings \( S = \{s_1, \cdots, s_n\} \), to find a minimum length string that contains all \( s_i, 1 \leq i \leq k \), as substrings.

This problem is proved to be \( NP-Complete \) and \( APX-hard \). Guaranteed approximation algorithms have been proposed, the current best ratio being \( \frac{211}{23} \), which has been achieved following a long and difficult quest. However, SSP is highly used in practice on next generation sequencing (NGS) data, which plays an increasingly important role in sequencing. In this note, we show that the SSP approximation ratio can be improved on NGS reads by assuming specific characteristics of NGS data that are experimentally verified on a very large sampling set.

1 Introduction

The Shortest Superstring Problem (SSP) consists, for a set of strings \( S = \{s_1, \cdots, s_n\} \), in constructing a string \( s \) such that any element of \( S \) is a substring of \( s \) and \( s \) is of minimal length. For an arbitrary number of sequences \( n \), the problem is known to be \( NP-Complete \) \[8, 9\] and \( APX-hard \) [2]. Lower bounds for the achievable approximation ratios on a binary alphabet have been given by Ott [15]. The best known approximation ratio so far is \( \frac{211}{23} \approx 2.478 \) [14] after a long series of improvements [13, 2, 12, 5, 6, 17, 19, 11, 16].

In the meantime, SSP compression algorithms have been designed as sub-routines of the previous ones. The idea is to ensure a fixed compression ratio between the sum of the lengths of the sequences of the set and the optimal superstring on this set. The greedy algorithm is such a compression algorithm that is proven to achieve a compression ratio of at least \( \frac{1}{2} \), while the best compression algorithm achieves a ratio of \( \frac{58}{63} \) [12].

In this note, we focus on practical applications of SSP, like assembling biological sequences, mostly DNA sequences with an alphabet of \( \{A,C,G,T\} \) named bases, but also on proteome sequences with a 26 letter alphabet corresponding to amino acids. SSP is used in contig reconstruction step, contigs that subsequently need to be organised.

Over the past decade, the landscape of sequencing and assembly deeply changed, with the increasing development of Next Generation Sequencing (NGS) devices. These relatively cheap devices produce, from a “soup” of cells, millions of randomly read, short, equal length DNA sequences in a single run. Each sequence is typically 32 to 1000 bases long, with a small and still decreasing cost per base. Such sequences are named reads. NGS technology allows to tackle new challenges in biology and medicine; the exponential increase of sequencing demands leads to the creation of more and more sequencing platforms, dealing with NGS data at 99%.

∗LaBRI and CBiB, University of Bordeaux, France.
†CNRS, LaBRI and CBiB, University of Bordeaux, France.
Considering the specificity of read sequences, is it possible to propose better approximation algorithms for this type of data? This research, similar to the one targeting better algorithms for small-world graphs in social networks, aims to better suit the actual data.

This note is a first step in this direction. We first model the read sequences more finely thus, according to our examples, better matching the experimental data. Then, we derive a better approximation ratio algorithm by using the properties of the reads. For instance, on the set SRR069579, we reach a 2.0738 approximation ratio (see Table 1). To our knowledge, the only related work is \[\text{[10]}\], where the sequences have the same length. Up to 7 bases, they propose a better approximation ratio based on De Bruijn graphs. However, these sequences are way shorter than real-world reads.

Note that some theoretical variations of SSP have also been studied \[\text{[21]}\]. Here we do not dwell on these studies since their focus is far from ours, neither do we detail the greedy algorithm approximation conjecture, which is a subject by itself \[\text{[18]}\].

2 Modeling of reads

NGS reads have some specific properties that we model and exhibit on real sets of reads.

For a string \(s\) of length \(n\), any integer \(1 \leq p \leq m\) is a period of \(s\) if \(s[i] = s[i + p]\) for all \(1 \leq i \leq m - p\). Note that \(s\) always has at least one period, corresponding to its length. The smallest period of \(s\) is called the period of \(s\), and denoted period(s).

We consider SSP on \(n\) reads \(S = \{s_1, s_2, \ldots, s_n\}\) of length \(m > 0\), where \(m \ll n\). We now consider the period of each read. We denote \(n(i), 1 \leq i \leq m\), the number of reads of period \(i\).

Let \(0 \leq \alpha \leq 1\) be a parameter and let \(sp = \sum_{i=1}^{m} n(i) / m\). We express \(sp\) (for small period) as a percentage of \(\frac{n}{m}\) relatively to the value of \(\alpha\) and we denote \(sp = \text{perc}_{\alpha} \frac{n}{m}\).

A strong characteristic of a set of reads is that even for ratios \(0.8 < \alpha < 1\), \(sp\) is very small compared to \(n\). The order of magnitude is \(sp\) being a few per hundred of \(\frac{n}{m}\). For a large panel of sets of reads on which we tested our approach, we found for \(0.8 < \alpha < 1\) a \(sp\) value inferior to \(0.02 \frac{n}{m}\). In Figure 1 we show such four sets of reads with lengths of 32, 36, 98 and 200.

3 Approximation algorithm

For two strings \(u, v\) we define the overlap of \(u\) and \(v\), denoted \(\text{ov}(u, v)\), as the longest suffix of \(u\) that is also a prefix of \(v\). Also, we define the prefix of \(u\) relatively to \(v\), denoted \(\text{pref}(u, v)\), as the string \(x\) such that \(u = x \text{ov}(u, v)\), i.e., the prefix of \(u\) that does not overlap \(v\).

The prefix graph (also called the distance graph) of \(S\) is a complete directed graph with the vertex set \(S\) and the edges \((s_i, s_j)\) of weight equal to the length \(\text{perc}(s_i, s_j)\).

We consider the classical algorithm of \[\text{[2]}\text{[20]}\], which gives a general framework. This algorithm is proved to be a 3 approximation algorithm in the general case. We prove below that applied on NGS data the approximation factor can be improved. The scheme of the algorithm is the following:

1. Compute a maximal cycle decomposition on the prefix graph
2. For each cycle \(c_i\) choose one of the strings in \(c_i\) as a representative string \(r_i\).
3. \(\sigma_i = (\text{pref}(r_i, r_{i+1}) \cdot \ldots \cdot \text{pref}(r_k, r_i)) \cdot r_i\) (cycle from \(r_i\) concatenated with \(r_i\)).
4. Let \(S_\sigma = \{\sigma_i\}\) and \(w_\sigma\) as a concatenation of all \(\sigma_i\).
5. Compress \(w_\sigma\) using an SSP compression algorithm.

The cycle decomposition produces cycles of several lengths. The period of a cycle is given by its length. We split the set of cycles, in two parts, the small cycles of period less than or equal to
Figure 1: Sets of reads from left to right, top to bottom: SRR069579 (human), ERR000009 (yeast), SRR211279 (human), SRR959239 (human). The x-axis is the period and the y-axis is in log_{10} scale. The circles represent $n(x)$ and the crosses $\sum_{i=1}^{x} \frac{n(x)}{x}$. The dash vertical line corresponds to the final $ma$ (computed in the experimental results in Section 4) and the horizontal to 0.02 $\frac{n}{m}$.

$m\alpha$, and the larger ones, denoted large. We now focus on the number of small cycles. The weight of a cycle is the sum of the weights of its edges and let $wt(C)$ be the sum of the weights of all cycles.

**Lemma 1** Let $c \in C$ be a cycle and $s$ a sequence in the cycle, then $\text{period}(c) \geq \text{period}(s)$

*Proof.* Each sequence in the cycle can be expressed by turning around the cycle. If $\text{period}(c) < \text{period}(s)$, then $\text{period}(c)$ is also a period of $s$, which is smaller than its smallest period, contradiction. \(\square\)

**Corollary 1** Let $c \in C$ be a cycle and $s_1 \ldots s_k$ the sequences in $C$. Then $\text{period}(c) \geq \max\{\text{period}(s_i)\}$.

*Proof.* Directly derives from lemma 1. \(\square\)

**Lemma 2** Let $c \in C$ be a cycle and $s_1 \ldots s_k$ the sequences in $C$. If $\text{period}(c) \leq m\alpha$, $\text{period}(s_i) \leq m\alpha$.

*Proof.* By corollary 1 the periods of all sequences in a cycle are smaller than or equal to the period of the cycle. \(\square\)

**Corollary 2** Let $1 \leq i \leq m$, the maximal number of cycles of period less or equal to $i$ is bounded by $\frac{1}{2} \sum_{k=1}^{i} n(i)$.

*Proof.* A cycle contains at least two sequences. By Lemma 2 all the sequences in $c$ of period $i$ must have a period less than or equal to $i$ and there are only $\frac{1}{2} \sum_{k=1}^{i} n(i)$ such sequences. \(\square\)
3.1 Analysis of the algorithm

We bounded the number of small cycles relatively to $\alpha$. Let us now take this into account while analysing the approximation algorithm. Obviously, $w_\sigma$ is a superstring of $S$. Let us bound its size.

Lemma 3

$$|w_\sigma| = \sum |\sigma_i| \leq wt(C) + wt(C)\frac{1}{\alpha} + \frac{sp \cdot m}{2} \leq (1 + \frac{1}{\alpha})OPT + \frac{sp \cdot m}{2}$$

Proof. A $\sigma_i$ is formed on a cycle as $(\text{pref}(r_i, r_{i+1}) \cdot ... \cdot \text{pref}(r_k, r_i)) \cdot r_i$. We first sum over all the $\sigma_i$ the prefixes of each $\sigma_i$ corresponding to the cycle. This leads to a first global $wt(C)$. Then we consider the sizes of the $r_i$ for the large cycles. The point is that all $r_i$ have the same length $m$, and that each $r_i$ can be represented (or expressed) by turning around the cycle it corresponds to (see Figure 2). As large cycles have a period of at least $ma$, turning $\frac{1}{\alpha}$ period($c_i$) around the cycle $c_i$ is enough to read $r_i$. Thus, the sum over all large cycles of $|r_i|$ is bounded by $wt(C)\frac{1}{\alpha}$.

![Figure 2: Expressing a representative $r$ over the cycle it belongs to. The period of cycle (a) is larger than $ma$ and thus the expression of $r$ requires only $1/\alpha$ cycles. The period of cycle (b) is $2 \leq i < ma$ and the expression of $r$ thus requires $m/i$ cycles, which can be at maximum $m/2$](image)

The remaining step is counting the sum of the $r_i$ corresponding to $n_\alpha$ small cycles.

As, by corollary 2 there are at most $sp/2$ such cycles, the sum of the corresponding $r_i$ is bounded by $sp \cdot m/2$. This would already be an acceptable bound since $sp$ is small relatively to $m/n$. But this implies counting $m$ for all small cycles, independently of the periods of the cycles, which can vary from 2 to $ma$. The larger the period of the cycle, the less we need to turn on the cycle to read the representative $r_i$. Thus our worst case for counting the small cycles from period 1 to $ma$ is when there is a maximum of smaller cycles at each step $2 \leq k \leq ma$ and, by corollary 2 this maximum from $k - 1$ to $k$ can only be increased by $n(k)$. Expressing the representative of each such $n(k)$ additional cycles of period $k$, requires $n(k)\frac{m}{k}$. Thus, the expression of the representatives of all the small cycles is bounded by $\frac{1}{2} \sum_{i=1}^{ma} \frac{n(i) \cdot m}{i} = \frac{sp \cdot m}{2}$.

Eventually, as $wt(C) \leq OPT$, the result follows. □

We then compress $S_\sigma$ using the guaranteed compression algorithm of [12], similarly to the classical approaches related to the superstring approximation. We define $OPT_\sigma$ as an optimal minimal superstring on $S_\sigma$ and $\tau$ as the result of the compression algorithm on $S_\sigma$. The next lemma [2] allows us to link $OPT_\sigma$ and $OPT$.

Lemma 4 $OPT_\sigma < OPT + wt(C)$

By applying the compression algorithm on $S_\sigma$, we thus derive the following result:

Lemma 5 $|\tau| \leq 2OPT + \frac{38}{63} \left(\frac{1-\alpha}{\alpha}\right)OPT + \frac{38}{129}sp \cdot m$

Proof. Lemma 1 gives $OPT_\sigma < OPT + wt(C) \leq 2OPT$ (see Figure 3). The distance from $OPT_\sigma$ to $|w_\sigma|$ is greater than or equal to $|w_\sigma| - 2OPT$. In the worst case it is equal, then the compression algorithm applies a compression factor $38/63$ to this distance, which leads to the result. □
Figure 3: Compressing $w_\sigma$ using the 38/63 algorithm [2]

An important point is that $OPT \geq n$, since any superstring contains at least one base of each sequence. As $sp = perc_\alpha \frac{n \cdot 38}{63} \frac{m}{2} \leq \frac{38}{126} perc_\alpha OPT$

**Theorem 1**

$$|\tau| \leq 2OPT + \frac{38}{63} \left(1 - \frac{1}{\alpha}\right) OPT + \frac{38}{126} perc_\alpha OPT$$

| period | nbseq | cum. nbseq | $\alpha$ | $1 + \frac{1}{\alpha}$ | $2 + \frac{38}{63} \left(1 - \frac{\alpha}{\alpha}\right)$ | $\frac{38}{126} sp \frac{m}{n}$ | $\beta$ |
|--------|-------|------------|----------|----------------|-------------------------------------------------|----------------------------|--------|
| 1      | 4     | 4          | 0.0277778| 37            | 23.1111                                          | 1.74749e-05                | 23.111 |
| 2      | 6     | 10         | 0.055556 | 19            | 12.254                                           | 3.0581e-05                 | 12.254 |
| ...    | ...   | ...        | ...      | ...           | ...                                              | ...                        | ...    |
| 32     | 23746 | 41326      | 0.888889 | 2.125         | 2.0754                                           | 0.00588868                 | 2.08129 |
| 33     | 98795 | 140121     | 0.916667 | 2.09091       | 2.05483                                          | 0.0189677                  | 2.0738 |
| 34     | 247451| 387572     | 0.944444 | 2.05882       | 2.03548                                          | 0.0507631                  | 2.0824 |
| 35     | 829535| 1217107    | 0.972222 | 2.02857       | 2.01723                                          | 0.154306                   | 2.17154|
| 36     | 2485202| 3702309   | 1       | 2             | 2                                                 | 0.455893                   | 2.45589|

Table 1: SRR069579 read set, 3702309 reads of size 36. $\beta = 2 + \frac{38}{63} \left(1 - \frac{1}{\alpha}\right) + \frac{38}{126} sp \frac{m}{n}$

| period | nbseq | cum. nbseq | $\alpha$ | $1 + \frac{1}{\alpha}$ | $2 + \frac{38}{63} \left(1 - \frac{\alpha}{\alpha}\right)$ | $\frac{38}{126} sp \frac{m}{n}$ | $\beta$ |
|--------|-------|------------|----------|----------------|-------------------------------------------------|----------------------------|--------|
| 1      | 4     | 4          | 0.03125  | 33            | 20.6984                                          | 8.38862e-06                | 20.6984 |
| 2      | 8     | 12         | 0.0625   | 17            | 11.0476                                          | 1.76772e-05                | 11.0476 |
| ...    | ...   | ...        | ...      | ...           | ...                                              | ...                        | ...    |
| 28     | 30474 | 61366      | 0.875    | 2.14286       | 2.08617                                          | 0.00518227                 | 2.09135 |
| 29     | 89474 | 150840     | 0.90625  | 2.10345       | 2.0624                                          | 0.0119997                  | 2.0744 |
| 30     | 341160| 492000     | 0.9375   | 2.06667       | 2.04021                                          | 0.0371279                  | 2.0734 |
| 31     | 953389| 1445389    | 0.96875  | 2.03226       | 2.01946                                          | 0.105085                   | 2.12454|
| 32     | 2606944| 4052333 | 1       | 2             | 2                                                 | 0.285099                   | 2.2851 |

Table 2: ERR000009 read set, 4052333 reads of size 32

| period | nbseq | cum. nbseq | $\alpha$ | $1 + \frac{1}{\alpha}$ | $2 + \frac{38}{63} \left(1 - \frac{\alpha}{\alpha}\right)$ | $\frac{38}{126} sp \frac{m}{n}$ | $\beta$ |
|--------|-------|------------|----------|----------------|-------------------------------------------------|----------------------------|--------|
| 1      | 2     | 2          | 0.005    | 201           | 122.032                                          | 4.80545e-06                | 122.032 |
| 100    | 23    | 25         | 0.5      | 3             | 2.60317                                          | 5.35808e-06                | 2.60318 |
| ...    | ...   | ...        | ...      | ...           | ...                                              | ...                        | ...    |
| 195    | 38013 | 54574      | 0.975    | 2.02564       | 2.01547                                          | 0.00067922                 | 2.01615 |
| 196    | 134284| 188858     | 0.98     | 2.02041       | 2.01231                                          | 0.00232588                  | 2.01464 |
| 197    | 473686| 662544     | 0.985    | 2.01523       | 2.00919                                          | 0.00810323                 | 2.01729 |
| 198    | 1685038| 2347582 | 0.99     | 2.0101        | 2.00699                                          | 0.0285511                  | 2.03464 |
| 199    | 5811666| 8159248 | 0.995    | 2.00503       | 2.00303                                          | 0.0987212                  | 2.10175 |
| 200    | 16944518| 25103766 | 1       | 2             | 2                                                 | 0.302286                   | 2.30229 |

Table 3: SRR211279 read set, 25103766 reads of size 200
4 Experimental results

We present experimental results for the sets of reads SRR069579 (Table 1), ERR000009 (Table 2), SRR211279 (Table 3), and SRR959239 (Table 4).

In each table, for each period \( i \) from 1 to \( m \) we show: (a) \( n(i) \), (b) the cumulative number of sequences, (c) the value of \( \alpha \) corresponding to \( i/m \), (d) the value of \( 1 + \frac{1}{\alpha} \), (e) \( 2 + \frac{38}{63} (\frac{1-\alpha}{\alpha}) \) which corresponds to the term of equation 1 due to the large cycles, (f) \( \frac{38}{126} sm \) which is the part of the final ratio brought by the small cycles, and eventually (g) \( \beta = 2 + \frac{38}{63} (\frac{1-\alpha}{\alpha}) + \frac{38}{126} sm \), the final ratio that can be reached by using the value of \( \alpha \) from the previous line in the table.

The resulting approximation ratios on the read sets cited above are respectively 2.0738, 2.09, 2.01464 and 2.02623.

Table 4: SRR959239 read set, 4143243 reads of size 98

| period | nbseq | cum. nbseq | \( \alpha \) | \( 1 + \frac{1}{\alpha} \) | \( 2 + \frac{38}{63} (\frac{1-\alpha}{\alpha}) \) | \( \frac{38}{126} sm \) | \( \beta \) |
|--------|-------|------------|-------------|-----------------|-------------------------------|----------------|--------|
| 1      | 1     | 1          | 0.0102041   | 99              | 60.5079                       | 7.13344e-06    | 60.5079 |
| 50     | 4     | 5          | 0.510204    | 2.96            | 2.57905                       | 7.70411e-06    | 2.57906 |
| ...    | ...   | ...        | ...         | ...             | ...                          | ...            | ...    |
| 95     | 65228 | 93719      | 0.969388    | 2.03158         | 2.01905                       | 0.00708125     | 2.02613 |
| 96     | 302973| 396692     | 0.979592    | 2.02083         | 2.01257                       | 0.0295941      | 2.04216 |
| 97     | 942267| 1338959    | 0.989796    | 2.01031         | 2.00622                       | 0.088889       | 2.10511 |
| 98     | 2804284| 4143243   | 1           | 2               | 2                            | 0.303013       | 2.30301 |

References

[1] C. Armen and C. Stein. A 2 \( \frac{2}{2} \) superstring approximation algorithm. Discrete Applied Mathematics, 88(1–3):29 – 57, 1998. Computational Molecular Biology \{DAM\} - \{CMB\} Series.

[2] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. J. ACM, 41(4):630–647, July 1994.

[3] D. Breslauer, T. Jiang, and Z. Jiang. Rotations of periodic strings and short superstrings. Journal of Algorithms, 24(2):340 – 353, 1997.

[4] A. Chris and S. Clifford. Algorithms and Data Structures: 4th International Workshop, WADS ‘95 Kingston, Canada, August 16–18, 1995 Proceedings, chapter Improved length bounds for the shortest superstring problem, pages 494–505. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[5] M. Crochemore, M. Cygan, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, and T. Walen. Algorithms for three versions of the shortest common superstring problem. In CPM, volume 6129 of Lecture Notes in Computer Science, pages 299–309. Springer, 2010.

[6] A. Czumaj, L. Gasieniec, M. Piotrów, and W. Rytter. Sequential and parallel approximation of shortest superstrings. Journal of Algorithms, 23(1):74 – 100, 1997.

[7] G. Fici, T. Kociumaka, J. Radoszewski, W. Rytter, and T. Walen. On the greedy algorithm for the shortest common superstring problem with reversals. Inf. Process. Lett., 116(3):245–251, 2016.

[8] J. Gallant, D. Maier, and J. Astorer. On finding minimal length superstrings. Journal of Computer and System Sciences, 20(1):50 – 58, 1980.

[9] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990.

[10] A. Golovnev, A. S. Kulikov, and I. Mihajlin. Approximating shortest superstring problem using de Bruijin graphs. In CPM, volume 7922 of Lecture Notes in Computer Science, pages 120–129. Springer, 2013.

[11] H. Kaplan and N. Shafrir. The greedy algorithm for shortest superstrings. Inf. Process. Lett., 93(1):13–17, 2005.

[12] S. R. Kosaraju, J. K. Park, and C. Stein. Long tours and short superstrings. In Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on, pages 166–177, Nov 1994.
[13] M. Li. Towards a DNA sequencing theory (learning a string). In Proc. of the 31st Symposium on the Foundations of Comp. Sci., pages 125–134. IEEE Computer Society Press, Los Alamitos, CA, 1990.

[14] M. Mucha. Lyndon words and short superstrings. In SODA, pages 958–972. SIAM, 2013.

[15] Sascha Ott. Graph-Theoretic Concepts in Computer Science: 25th International Workshop, WG’99 Ascona, Switzerland, June 17–19, 1999 Proceedings, chapter Lower Bounds for Approximating Shortest Superstrings over an Alphabet of Size 2, pages 55–64. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[16] K. E. Paluch, K. M. Elbassioni, and A. van Zuylen. Simpler approximation of the maximum asymmetric traveling salesman problem. In STACS, volume 14 of LIPIcs, pages 501–506. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[17] Z. Sweedyk. a $2\frac{1}{2}$-approximation algorithm for shortest superstring. SIAM J. Comput., 29(3):954–986, December 1999.

[18] J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. Theoretical Computer Science, 57(1):131 – 145, 1988.

[19] S.-H. Teng and F. F. Yao. Approximating shortest superstrings. SIAM J. Comput., 26(2):410–417, 1997.

[20] V. V. Vazirani. Approximation Algorithms. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[21] Y. W. Yu. Approximation hardness of shortest common superstring variants. CoRR, abs/1602.08648, 2016.