Kapun and Tsarev *BMC Bioinformatics* 2013, **14**(Suppl 5):S7
http://www.biomedcentral.com/1471-2105/14/S5/S7

BMC Bioinformatics

PROCEEDINGS Open Access

De Bruijn Superwalk with Multiplicities Problem is NP-hard

Evgeny Kapun, Fedor Tsarev*

From RECOMB-seq: Third Annual Recomb Satellite Workshop on Massively Parallel Sequencing
Beijing, China. 11-12 April 2013

Abstract

De Bruijn Superwalk with Multiplicities Problem is the problem of finding a walk in the de Bruijn graph containing several walks as subwalks and passing through each edge the exactly predefined number of times (equal to the multiplicity of this edge). This problem has been stated in the talk by Paul Medvedev and Michael Brudno on the first RECOMB Satellite Conference on Open Problems in Algorithmic Biology in August 2012. In this paper we show that this problem is NP-hard. Combined with results of previous works it means that all known models for genome assembly are NP-hard.

Introduction

The majority of current genome sequencing technologies are based on the shotgun method – the genome is split into several small fragments which are read directly. The problem of reconstructing the initial genome from these small fragments (reads) is known as the genome assembly problem. It is one of the fundamental problems of bioinformatics. Several models for genome assembly were studied by researchers. If reads are assumed to be error-free, the assumption made in all models is that every read from the input must be a substring of the genome.

One of the models is based on maximum parsimony principle – the original genome should be the shortest string containing all reads as substrings. This leads to the Shortest Common Superstring (SCS) problem which is NP-hard [1]. Modeling genome assembly as the SCS problem has a sufficient drawback: the majority of genomes have repeats – multiple similar (or even equal) fragments, while the SCS solution would under-represent these repeats.

The de Bruijn graph model proposed in [2] deals with repeats much better. It is based on generating a set of all $(k + 1)$-character substrings (called $(k + 1)$-mers) of reads and constructing a de Bruijn graph in which the vertices are k-mers and edges are $(k + 1)$-mers. Each read is represented by a walk in this graph. Any walk containing all the reads as subwalks represents a valid assembly. Consequently, the genome assembly problem is formulated as finding the shortest superwalk, which is closely related to the polynomial time Eulerian tour problem (which was previously used to solve the problem of sequencing by hybridization [3]). Despite that, the Shortest De Bruijn Superwalk problem (SDBS) was shown to be NP-hard [4]. Note also that SDBS has a special case solvable in polynomial time – if each subwalk contains only one edge, this problem can be reduced to Chinese Postman Problem [5].

In [6] an algorithm for reads' copy counts estimation based on maximum likelihood principle was proposed. A similar algorithm can be applied to find multiplicities of edges in the de Bruijn graph, so, the following problem was formulated in the talk [7]. Given a de Bruijn graph with vertices of size k constructed from a set of reads and multiplicities (in unary notation) of all edges of this graph find a superwalk consistent with edge multiplicities and containing all reads as subwalks. This problem is named De Bruijn Superwalk with Multiplicities problem (DBSM) and its computational complexity remained unknown.

In this paper we prove that this problem is NP-hard.

* Correspondence: tsarev@rain.ifmo.ru
St. Petersburg National Research University of Information Technologies, Mechanics and Optics Genome Assembly Algorithms Laboratory 197101, Kronverksky pr., 49, St. Petersburg, Russia

NP-hardness proof

The proof has the following structure. First, the Common Superstring with Multiplicities (CSM) problem is formulated. This problem is shown to be NP-hard by reducing SCS to it. Then CSM is reduced to de Bruijn Superwalk with Multiplicities problem.

Let S be a string over alphabet Σ. Let $L_c(S)$ denote the number of occurrences of character $c \in \Sigma$ in S. Then, let Common Superstring with Multiplicities problem be the problem, given strings $S_1, S_2, ..., S_n$ and nonnegative integers l_c for all $c \in \Sigma$ (given in unary notation), to find out if there exists a string S such that:

- all strings $S_1, S_2, ..., S_n$ are substrings of S,
- $L_c(S) = l_c$ for each $c \in \Sigma$.

Theorem 1. *Common Superstring with Multiplicities problem is NP-hard for* $|\Sigma| = 2$.

Proof. To prove this, we take an instance of Shortest Common Superstring problem with $\Sigma = \{0, 1\}$, which is NP-hard [8], and transform it into an instance of Common Superstring with Multiplicities problem with the same answer. Let the original instance of SCS problem be $\{S'_1, S'_2, ..., S'_n\}$, l' (this instance means that we need to find if there exists a superstring of $S'_1, S'_2, ..., S'_n$ having length at most l').

Let us define $T_0 = 000111$ and $T_1 = 001011$. These strings have been selected in such a way that each of them contains the same number of zeroes and ones and they do not overlap – no proper suffix of any of the T_c ($c \in \{0, 1\}$) is equal to any of the proper prefixes of any of the T_c($c \in \{0, 1\}$).

Then, let $S_k = T(S'_k)$ and $l_0 = l_1 = 3l'$, where $T(c_1c_2 \ldots c_k) = T_{c_1}T_{c_2} \ldots T_{c_k}$. The following lemmas formulate several properties of these instances of SCS and CSM problems. Equivalence of these instances is shown in lemmas 3 and 7.

Lemma 1. $L_0(T(S')) = L_1(T(S')) = 3|S'|$.

Proof. It follows directly from the definition of T.

Lemma 2. *If* S'_1 *is a substring of* S'_2, *then* $T(S'_1)$ *is a substring of* $T(S'_2)$.

Proof. It follows directly from the definition of T.

Lemma 3. *If the answer for the original instance of SCS problem is positive, then the answer for the instance of CSM problem is also positive*.

Proof. If the answer for the instance of SCS problem is positive, then there exists a string S' of length $l'' \le l'$ such that S' is a superstring of $S'_1, S'_2, \ldots, S'_n$. Then, let $S = T(S'0^{l'-l''})$. Because $|S'0^{l'-l''}| = |S'| + |0^{l'-l''}| = l'' + (l'-l'') = l'$, $L_0(S) = L_1(S) = 3l'$ (see lemma 1) and all $S_i$are substrings of $T(S')$ (see lemma 2) the answer to the instance of CSM is indeed positive.

Lemma 4. *Let* S'_1*and* S'_2*be two strings such that there is a suffix of* $T(S'_1)$ *equal to a prefix of* $T(S'_2)$. *Then the following holds:*

- the length of that suffix is a multiple of 6,
- if the length of the suffix is 6*k, then the suffix of length k of* S'_1 *is equal to the prefix of length k of* S'_2.

Proof. Suppose that the length of the suffix is equal to $6k + i$, $0 < i < 6$. Let c_1 be the last character of S'_1 and c_2 be the character at the $(k + 1)$-th position of S'_2 (positions are numbered starting from one). Then, the suffix of T_{c_1} of length i would be equal to the prefix of T_{c_2} of the same length.

As mentioned before, no proper suffix of any of the T_c ($c \in \{0, 1\}$) is equal to any of the proper prefixes of any of the T_c($c \in \{0, 1\}$). Therefore, the length of the suffix is a multiple of 6. The second follows from T_0 and T_1 both having length 6 and $T_0 \ne T_1$.

Lemma 5. *Let* S'_1*and* S'_2 *be two strings such that* $T(S'_1)$ *is a substring of* $T(S'_2)$.
Then following statements hold:

- each occurrence of $T(S'_1)$ *in* $T(S'_2)$ *starts at a position which is congruent to* 1 *modulo* 6,
- if $T(S'_1)$ *occurs at position* $6k + 1$ *in* $T(S'_2)$, *then* S'_1 *occurs as a substring of* S'_2 *at position* $k + 1$.

Proof. The proof is analogous to lemma 4.

Lemma 6. *Let* $S'_1, S'_2, ..., S'_n$ *be a set of strings, and let S be a superstring of* $T(S'_1), T(S'_2), \ldots, T(S'_n)$ *such that* $T(S'_2), ..., T(S'_n), T(S'_2), ..., T(S'_n)$ *occur in S at positions* $i_1, i_2, ..., i_n$ *respectively (if some* $T(S'_k)$ *occurs in S in multiple positions only one position is recorded) and every character of S is covered by at least one of those occurrences. Then the following statements hold:*

- $i_1, i_2, ..., i_n$ *are all congruent to* 1 *modulo* 6,
- length of S is a multiple of 6,
- There exists a string S' such that S = T(S'). Strings $S'_1, S'_2, ..., S'_n$ *occur in S' at positions* $i'_1, i'_2, ..., i'_n$*where* $i_k = 6i'_k - 5$ *for* $k = 1, 2, ..., n$.

Proof. Suppose the contrary. Let $i_k$be the smallest of $i_1, i_2, ..., i_n$which is not congruent to 1 modulo 6. Then, if i_k-th character of S is covered by some $T(S'_{k'})$ such that $i_{k'} < i_k$, we have a contradiction because $i_{k'}$ is not congruent with i_k modulo 6, but either $T(S'_k)$ and $T(S'_{k'})$ overlap, or $T(S'_k)$ is a substring of $T(S'_{k'})$, which would violate either lemma 4 or lemma 5. If i_k-th character of S is not covered by any $T(S'_{k'})$, such that, $i_{k'} < i_k$, than $(i_k - 1)$-th character of S must be covered by the last character of some $T(S'_{k'})$.

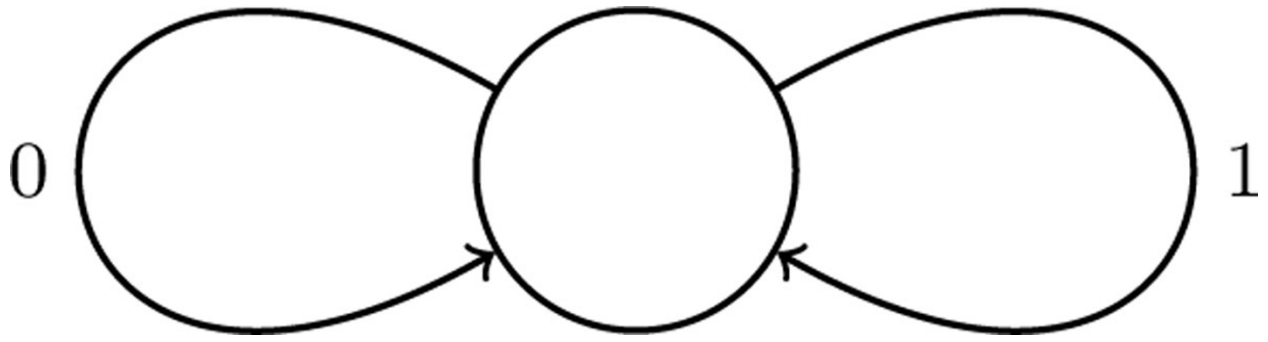

Figure 1 A graph on which Common Superwalk with Multiplicities problem is NP-hard.

But length of $T(S'_{k'})$ is a multiple of 6, so $i_k$must be congruent to $i_{k'}$ modulo 6, which leads to a contradiction.

The last character of S is also covered by the last character of some $T(S'_k)$. Because $i_k$is congruent to 1 modulo 6 and the length of $T(S'_k)$ is a multiple of 6, the length of S is also a multiple of 6.

To prove the last point, it is enough to notice that for j = 1, 7, ..., $|S|$ - 5, the substring of S starting at position j and having length 6 is either T_0 or T_1. This follows from the fact that the j-th character of S is covered by an occurrence of $T(S'_k)$ for some k, but restrictions on lengths of $T(S'_k)$ and on i_k mean that the whole substring of length 6 would be covered by $T(S'_k)$. Moreover, the position at which the substring of length 6 occurs in $T(S'_k)$ is congruent to 1 modulo 6, therefore that substring is either T_0 or T_1 by definition of T .

Lemma 7. *If the answer for the instance of CSM problem is positive, then the answer for the original instance of SCS problem is also positive.*

Proof. If the answer for the instance of CSM problem is positive, then there exists a string S of length 6l' which is a superstring of S_1, S_2, ..., S_n. Let S'' be the shortest common superstring of these strings. Then $|S''| \leq 6l'$ and each character of S'' is covered by an occurrence of one of S_1, S_2, ...,S_n. Recall that $S_k = T(S'_k)$. By lemma 6, there exists a string S' such that $S'' = T(S')$ and $S'_1, S'_2, ..., S'_n$ are

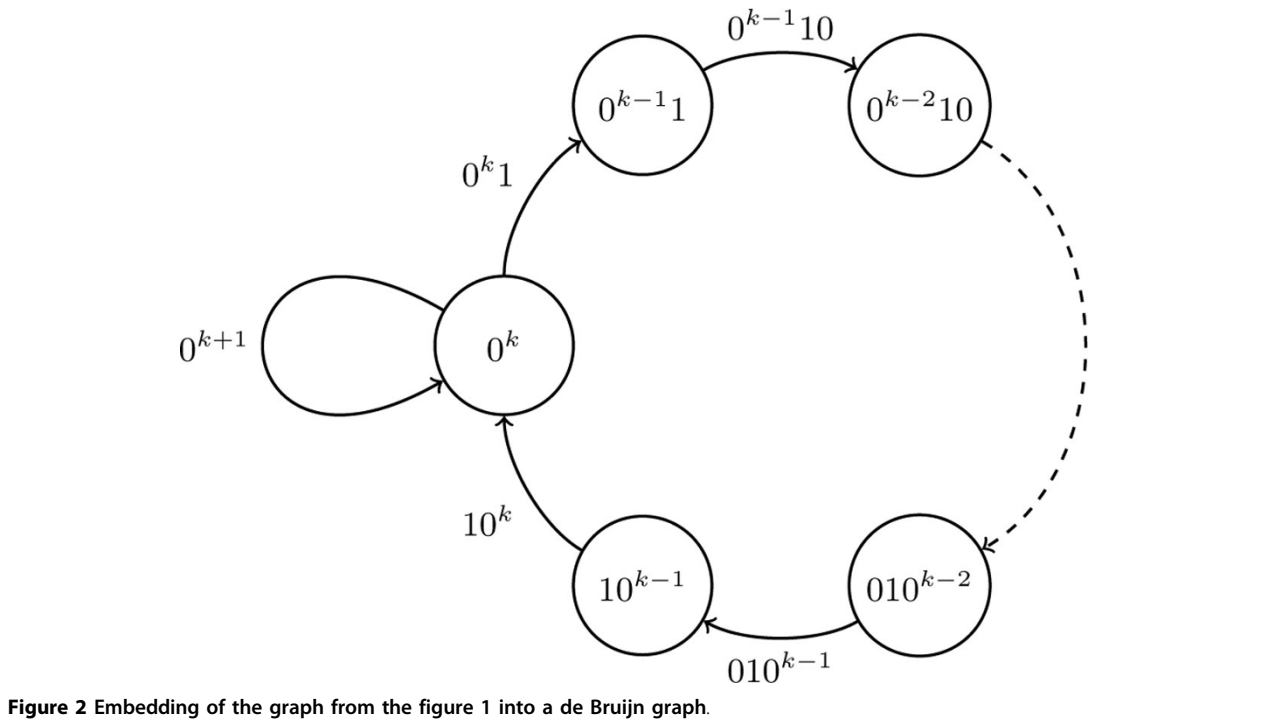

Figure 2 Embedding of the graph from the figure 1 into a de Bruijn graph.

substrings of S'. Also the equation $|S'| = \frac{|S''|}{6} \leq \frac{6l'}{6} = l'$ holds. Therefore, the answer for the original instance of SCS problem is also positive.

Theorem 2. *The de Bruijn Superwalk with Multiplicities Problem is NP-hard for any fixed* $|\Sigma| \geq 2$ *and any positive integer k.*

Proof. Consider the graph with one vertex and two loops (see Figure 1). An instance of Common Superstring with Multiplicities problem with $\Sigma = \{0, 1\}$ can be translated into an instance of Superwalk with Multiplicities problem on this graph in the following way:

- each S_k is directly translated into a walk, by representing 0 as occurrence of edge 0 and 1 as occurrence of edge 1 in the walk,
- the multiplicity of edge 0 is set to l_0, and the multiplicity of edge 1 is set to l_1.

To complete the proof we need to embed this graph into a de Bruijn graph with given k.

This can be done in straightforward manner (see Figure 2). Edge 0 is mapped to a loop, while edge 1 is mapped to a cycle of length $k + 1$.

Conclusion

We have proved that the de Bruijn Superwalk with Multiplicities Problem is NP-hard. Results of this work combined with [4] show that all known models for genome assembly are NP-hard.

However, both de Bruijn Shortest Superwalk and de Bruijn Superwalk with Multiplicities problems have a special case (if subwalks consist of one edge) solvable in polynomial time. A reasonable direction for future research is to find if there exist other polynomially solvable special cases of these problems.

Authors' contributions

The work presented here was carried out in collaboration between all authors. All authors have contributed to, seen and approved the manuscript.

Acknowledgements

Research was supported by the Ministry of Education and Science of Russian Federation in the framework of the federal program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" (contract 16.740.11.0495, agreement 14.B37.21.0562).

Declarations

Publication of this article was supported by the Ministry of Education and Science of Russian Federation in the framework of the federal program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and by the University ITMO.

This article has been published as part of *BMC Bioinformatics* Volume 14 Supplement 5, 2013: Proceedings of the Third Annual RECOMB Satellite Workshop on Massively Parallel Sequencing (RECOMB-seq 2013). The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/14/S5.

Published: 10 April 2013

References

1. Gallant J, Maier D, Storer J: **On finding minimal length superstrings.** *J Comput Syst Sci* 1980, **20(1)**:50-58.
2. Pevzner P, Tang H, Waterman M: **An Eulerian path approach to DNA fragment assembly.** *Proceedings of the National Academy of Sciences* 2001, **98**:9748-9753.
3. Pevzner P: **1-Tuple DNA sequencing: computer analysis.** *J Biomol Struct Dyn* 1989, **7(1)**:63-73.
4. Medvedev P, Georgiou K, Myers G, Brudno M: **Computability of Models for Sequence Assembly, Algorithms in Bioinformatics, 7th International Workshop, WABI 2007, LNCS.4645**:289-301.
5. Edmonds J, Johnson E: **Matching, Euler tours and the Chinese postman.** *Mathematical Programming* 1973, **5**:88-124.
6. Medvedev P, Brudno M: **Maximum Likelihood Genome Assembly.** *Journal of Computational Biology* 2009, **16(8)**:1101-1116.
7. Medvedev P, Brudno M: **De Bruijn Superwalk with Multiplicities Problem.** *Talk at RECOMB Satellite Conference on Open Problems in Algorithmic Biology* St. Petersburg, Russia; 2012.
8. Garey M, Johnson D: **Computers and Intractability: A Guide to the Theory of NP-Completeness.** W. H. Freeman; 1979.

doi:10.1186/1471-2105-14-S5-S7
Cite this article as: Kapun and Tsarev: De Bruijn Superwalk with Multiplicities Problem is NP-hard. *BMC Bioinformatics* 2013 **14**(Suppl 5):S7.