JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 23, Number 0, 2016

Pp. 1–4
DOI: 10.1089/cmb.2016.0097

Case Study

Cautionary Tales of Inapproximability

DAVID BUDDEN[1,2,*] and MITCHELL JONES[1,3,*]

ABSTRACT

Modeling biology as classical problems in computer science allows researchers to leverage the wealth of theoretical advancements in this field. Despite countless studies presenting heuristics that report improvement on specific benchmarking data, there has been comparatively little focus on exploring the theoretical bounds on the performance of practical (polynomial-time) algorithms. Conversely, theoretical studies tend to overstate the generalizability of their conclusions to physical biological processes. In this article we provide a fresh perspective on the concepts of NP-hardness and inapproximability in the computational biology domain, using popular sequence assembly and alignment (mapping) algorithms as illustrative examples. These algorithms exemplify how computer science theory can both (a) lead to substantial improvement in practical performance and (b) highlight areas ripe for future innovation. Importantly, we discuss caveats that seemingly allow the performance of heuristics to exceed their provable bounds.

Keywords: algorithms, inapproximability, genomics, alignment.

1. SEQUENCE ASSEMBLY: WHERE THEORY MEETS PRACTICE

Given a set of n strings, $S = \{s_1, s_2, \ldots, s_n\}$, the goal of the shortest common superstring problem (SCSP) is to find the minimum length string, s, such that each $s_i \in S$ is a substring of s. The SCSP over the nucleotide alphabet, $\Sigma = \{A, C, G, T\}$, thus provides a simple and convenient model for the sequence assembly problem, whereby we wish to determine the DNA sequence from which a set of reads (or k-mers) are derived. This is a classic example of how decades of research on approximation bounds of **NP**-hard problems can be applied to improve the practical performance of algorithms in the computational biology domain.

A detailed review on the development of approximation and hardness bounds for SCSP is provided by Golovnev et al. (2013). Despite these advancements, the power of theoretical computer science abstractions is limited by how closely they represent the true biological problem (as we discuss later)—in this case, reversing the DNA fragmentation process inherent to high-throughput sequencing experiments. SCSP and its sequence assembly derivatives (Sweedyk, 2000; Kaplan and Shafrir, 2005) have thus been criticized for their assumptions regarding parsimony and tandem repeats (Nagarajan and Pop, 2009), motivating the application of graph theoretic models that make more appropriate sets of assumptions.

[1]Google, Inc., Pyrmont, Australia.
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.
[3]Department of Computer Science, University of Illinois at Urbana-Champaign.
*These authors contributed equally to this work.

1.1. NP-hardness of exact solutions

Most modern sequence assembly algorithms correct for the issues inherent to the SCSP by modeling the assembly problem as one of finding suitable paths in graphs constructed from reads (or *k*-mers). The simplest of these approaches involves the construction of an overlap graph, $OG(S)$ (over a set of strings $S=\{s_1, s_2, \ldots, s_n\}$), which is defined on a set of vertices, $V=\{1, 2, \ldots, n\}$, and weighted directed edges (Kececioglu and Myers, 1995; Myers, 2005). The weight of each edge, (i, j), is defined as the length of the largest string that is both a suffix of s_i and a prefix of s_j.

Similar to the SCSP, the process of sequence assembly involves traversing $OG(S)$ to determine the set of edges that yields the largest overlap. This can be accomplished by visiting each vertex (representing each substring $s_i \in S$) exactly once while maximizing the length of the traversed path (the maximum compression). This problem is precisely the maximum traveling salesman problem (TSP), which is akin to the problem of finding a Hamiltonian path except for weighted edges. Despite the convenience of this representation, the TSP is an **NP**-hard problem that affords approximation bounds of arguable benefit in the context of sequence assembly.

1.2. Approximation algorithms and de Bruijn graphs

A convenient method of circumventing the **NP**-hardness of sequence assembly (when modeled as overlap graphs) is to remodel the problem such that reads are associated with edges rather than vertices. The problem of sequence assembly can thus be modeled as a de Bruijn graph, $DG(S)$, which is a directed unweighted graph such that each vertex is a substring of some length k (k-mer) of the strings S (Compeau et al., 2011) (i.e., each string $s_i \in S$ is represented as an edge rather than a vertex). In the scenario that S contains every k-mer of the optimal assembly, this solution can be found exactly in polynomial time as an instance of the Eulerian path problem. Most modern assembly algorithms use a variant of this graph theoretic ''trick'' (Butler et al., 2008; Chaisson and Pevzner, 2008; Zerbino and Birney, 2008; Simpson et al., 2009; Li et al., 2010).

It is perhaps unsurprising that the exact polynomial-time solution to sequence assembly involves many simplifications to the true problem (Compeau et al., 2011), including that all k-mers present in the genome are contained within the set of reads, that all sequencing reads are error free, that each k-mer appears exactly once within the genome (i.e., no repeats), and that the genome consists of exactly one circular chromosome. In practice, the majority of popular assembly algorithms retain a polynomial-time Eulerian approach while introducing heuristics to address these erroneous assumptions (Butler et al., 2008; Chaisson and Pevzner, 2008; Zerbino and Birney, 2008; Simpson et al., 2009; Li et al., 2010). It is important to realize that no guarantee can be made regarding the quality of these assemblies; by definition of the problem, there is no canonical reference against which performance can be evaluated, and finding an exact validation solution to the more general de Bruijn super path problem is impractical (**NP**-hard (Pevzner et al., 2001; Medvedev et al., 2007; Nagarajan and Pop, 2009)).

2. THE INAPPROXIMABILITY OF SEQUENCE ALIGNMENT

There are many other problems in computational biology that can benefit from theoretical analysis. An excellent example is sequence alignment (i.e., mapping against a reference genome); as opposed to assembly, for which there exists an unambiguous canonical sequence from where a set of reads were produced, the purpose of mapping is typically to identify variations (e.g., insertions, deletions, or mutations) in sequenced reads with respect to an assembled reference genome. This emphasis on inexact matching poses a unique set of theoretical challenges.

2.1. A new theory for genomic deletion

We have recently shown that sequence alignment can be modeled as an instance of the maximum facility location problem (MFLP) (Canzar et al., 2016). In brief, given a bipartite graph of putative mappings from the donor genome (set of *clients*, C) to the reference genome (set of *facilities*, F), the weight $w_{u,v}$ of an edge from a client, $u \in C$, to facility, $v \in F$, is equivalent to our confidence in that deletion. Given that two predicted deletions directly conflict if and only if their genomic intervals overlap, we further associate each

facility, v, with its genomic interval, $I_v \subset \mathcal{R}$. The problem of sequence alignment has now been reformulated as one of selecting a nonoverlapping subset of facilities, $T \subset F$, such that $\sum_{u \in C} \max_{v \in T} w_{u,v}$ is maximized.

Importantly, the hardness of MFLP (and thus sequence alignment in this form) can be demonstrated by reduction from maximum coverage (Canzar et al., 2016). This reduction also preserves approximation bounds. This means that if $\mathbf{P} \neq \mathbf{NP}$, then there is no polynomial-time algorithm that can always find a solution better than $1 - e^{-1} \approx 0.63$ times the optimal solution.* The best known polynomial-time approximation algorithm for this problem yields 0.25-approximation (Feldman, 2013), highlighting a large gap for future innovation in this domain.

3. IMPOSSIBLY GOOD ALGORITHMS?

The size of high-throughput sequencing data sets necessitates the development and application of polynomial-time algorithms. Indeed, principled reductions of these biological scenarios to classical **NP**-hard problems provide a convenient framework for identifying theoretical bounds on the expected performance of these algorithms. However, it is important to recognize that these reductions are often imperfect. One example of this is sequence assembly; as discussed in this article, there have been many attempts at identifying appropriate abstractions for modeling this problem (e.g., SCSP, overlap, and de Bruijn graphs), each of which involves a set of assumptions and simplifications that limit its applicability.

A practical limitation of theoretical modeling is highlighted by our analysis of the sequence alignment problem (Canzar et al., 2016). As the hardness result for this problem is at least ≈ 0.63, one might think all hope is lost. It is then perhaps surprising that the empirical performance of this algorithm (when evaluated on the Illumina NA12878 genome) is far closer to the optimal solution in terms of precision and recall. Combined with the promising performance of other published heuristics (Butler et al., 2008; Chaisson and Pevzner, 2008; Zerbino and Birney, 2008; Simpson et al., 2009; Li et al., 2010), these results likely indicate that these hardness results generalize across a set of worst-case instances that are either unlikely or unable to be encountered in actual biological data. It is equally unlikely that one would draw this important conclusion from theory alone.

The benefits of computer science abstractions extend well beyond the class of problems touched in this article. Literature is brimming with successful heuristics for gene regulatory modeling (Budden et al., 2014a,b; Budden et al., 2015) and network inference (Hurley et al., 2014; Le Novère, 2015), whereas a theoretical study by Krishnan et al. (2007) claims that the latter is impossible to solve for nontrivial network topologies. We encourage researchers to continue exploring the applicability of rigorous theoretical analysis in new domains, but to adopt caution when drawing biological conclusions from their findings.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Budden, D.M., Hurley, D.G., and Crampin, E.J. 2014a. Predictive modelling of gene expression from transcriptional regulatory elements. *Brief. Bioinform.* 16, 616–628.

Budden, D.M., Hurley, D.G., and Crampin, E.J. 2015. Modelling the conditional regulatory activity of methylated and bivalent promoters. *Epigenetics Chromatin.* 8, 21.

Budden, D.M., Hurley, D.G., Cursons, J., et al. 2014b. Predicting expression: The complementary power of histone modification and transcription factor binding data. *Epigenetics Chromatin*. 7, 36.

Butler, J., MacCallum, I., Kleber, M., et al. 2008. ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Res*. 18, 810–820.

*A slightly easier reduction is from MAX-3SAT; however, this only gives us a hardness result of 7/8 (Håstad, 2001), which is not as tight as $1 - e^{-1}$.

Canzar, S., Elbassioni, K., Jones, M., et al. 2016. Resolving conflicting predictions from multimapping reads. *J. Comput. Biol*. 23, 203–217.

Chaisson, M.J., and Pevzner, P.A. 2008. Short read fragment assembly of bacterial genomes. *Genome Res*. 18, 324–330.

Compeau, P.E., Pevzner, P.A., and Tesler, G. 2011. How to apply de Bruijn graphs to genome assembly. *Nat. Biotechnol*. 29, 987–991.

Feldman, M. 2013. Maximization problems with submodular objective functions. Ph.D. thesis.

Golovnev, A., Kulikov, A.S., and Mihajlin, I. 2013. Approximating shortest superstring problem using de bruijn graphs, 120–129. Eds: J. Fischer and P. Sanders. *In Combinatorial Pattern Matching*. Springer, Berlin-Heidelberg.

Håstad, J. 2001. Some optimal inapproximability results. *J. ACM)*. 48, 798–859.

Hurley, D.G., Cursons, J., Wang, Y.K., et al. 2014. NAIL, a software toolset for inferring, analyzing and visualizing regulatory networks. *Bioinformatics*. 31, 277–278.

Kaplan, H., and Shafrir, N. 2005. The greedy algorithm for shortest superstrings. *Inf. Process. Lett*. 93, 13–17.

Kececioglu, J.D., and Myers, E.W. 1995. Combinatorial algorithms for DNA sequence assembly. *Algorithmica*. 13, 7–51.

Krishnan, A., Giuliani, A., and Tomita, M. 2007. Indeterminacy of reverse engineering of gene regulatory networks: The curse of gene elasticity. *PLoS One*. 2, e562.

Le Novère, N. 2015. Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet*. 16, 146–158.

Li, R., Zhu, H., Ruan, J., et al. 2010. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res*. 20, 265–272.

Medvedev, P., Georgiou, K., Myers, G., et al. 2007. Computability of models for sequence assembly, 289–301. Eds: R. Giancarlo and S. Hannenhall. *In Algorithms in Bioinformatics*. Springer, Berlin-Heidelberg.

Myers, E.W. 2005. The fragment assembly string graph. *Bioinformatics*. 21(Suppl 2), ii79–ii85.

Nagarajan, N., and Pop, M. 2009. Parametric complexity of sequence assembly: Theory and applications to next generation sequencing. *J. Comput. Biol*. 16, 897–908.

Pevzner, P.A., Tang, H., and Waterman, M.S. 2001. An eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci.* U. S. A. 98, 9748–9753.

Simpson, J.T., Wong, K., Jackman, S.D., et al. 2009. ABySS: A parallel assembler for short read sequence data. *Genome Res*. 19, 1117–1123.

Sweedyk, Z. 2000. A $2\frac{1}{2}$-approximation algorithm for shortest superstring. *SIAM J. Comput*. 29, 954–986.

Zerbino, D.R. and Birney, E. 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*. 18, 821–829.

Address correspondence to:
Mr. David Budden
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
32 Vassar Street
Cambridge, MA 02139-4307

E-mail: budden@csail.mit.edu