様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　６　月　２２　日現在

機関番号：　１４３０１
研究種目：　基盤研究(C)（一般）
研究期間：　2014～2017
課題番号：　２６３３００１４

研究課題名（和文）Fast Graph Algorithms for Phylogenetics

研究課題名（英文）Fast Graph Algorithms for Phylogenetics

研究代表者
ジャンソン　ジェスパー（Jansson, Jesper）

京都大学・白眉センター・特定准教授

研究者番号：６０５３６１００

交付決定額（研究期間全体）：（直接経費）　　3,500,000円

研究成果の概要（和文）：本研究では、同意木の中でもよく使われる、過半数に基づく同意木、緩やかな同意木、貪欲同意木、頻度差同意木、アダムス同意木、R*同意木と２種類の局地同意木に関して高速アルゴリズムを構築した。上記のうちのいくつかは、数十年ぶりに改善に成功したものである。
次に、いくつかの基礎的な合成木問題の計算複雑さを示し、NP困難性を示す派生問題に対し近似アルゴリズムを設計した。ふたつの進化系統樹に対する根付き三つ組を計算するための、高速かつメモリ効率的な重心道に基づくアルゴリズムを示した。
最後に、閉路が互いに素な進化系統ネットワークであるゴールド木間の根付き三つ組距離を高速に計算する手法を示した。

研究成果の概要（英文）：We developed fast algorithms for constructing several popular consensus trees: The majority rule consensus tree, the loose consensus tree, the greedy consensus tree, the frequency difference consensus tree, the Adams consensus tree, the R* consensus tree, and two kinds of local consensus trees. Some of them were the first improvements in a long time; for example, the previously fastest algorithm for the Adams consensus tree was from 1972.
Next, we characterized the computational complexity of some fundamental supertree problems and designed approximation algorithms for NP-hard problem variants. We also gave a fast and memory-efficient centroid paths-based algorithm for computing the rooted triplet distance between two phylogenetic trees.
Finally, we presented a fast method for the rooted triplet distance between galled trees (phylogenetic networks whose cycles are disjoint). Our strategy was to transform the galled trees into pairs of trees and apply the fast algorithm for trees.

研究分野：　Theory of informatics

キーワード：　Algorithm theory　Computational complexity　Implementations　Graph algorithms　Phylogenetic tree　Consensus tree　Supertree

１．研究開始当初の背景

The *phylogenetic tree* is very old data structure commonly used by scientists and scholars to describe evolutionary history. Enormous amounts of genomic data are being collected every day in laboratories all over the world, and it has become easy to share all this data via Internet and public databases. A fundamental algorithmic problem is how to build an accurate phylogenetic tree from such data.

Many alternative algorithms for constructing phylogenetic trees were invented in the 20th century [Felsenstein; 'Inferring Phylogenies', Sinauer Associates, Inc., 2004], but these classical algorithms are sometimes unable to handle *huge* datasets efficiently. One potential remedy is divide-and-conquer: first apply a computationally expensive method such as maximum likelihood to construct reliable trees for small, overlapping subsets of the leaf label set, and then use a computationally cheaper combinatorial method to merge the small trees into one large tree called a *phylogenetic supertree*. An illustrative example of a supertree of the mammals can be found in [Bininda-Emonds et al.; Nature (2007)]. In the presence of errors, to build a reliable phylogenetic tree with a large number of leaves is also challenging because of the computational complexity of the underlying optimization problems, many of which are NP-hard even to approximate.

The phylogenetic *network* model is a powerful extension of the phylogenetic *tree* model that can be used to describe non-treelike evolutionary events such as lateral gene transfer, hybridization, etc. by allowing certain internal nodes to have more than just one parent [Huson et al.; 'Phylogenetic Networks', Cambridge University Press, 2010]. When developing and assessing the accuracy of new methods for reconstructing phylogenetic networks, it is essential to have some way of measuring the structural similarity between two given phylogenetic networks. Unfortunately, not many such methods currently exist. On the negative side, graph comparison problems are computationally expensive in general, and one of the major inherent difficulties is that when a graph contains many cycles, we do not know where to start looking. On the positive side, classes of phylogenetic networks whose cycles interact in restricted ways sometimes have a deep combinatorial structure that can be exploited to obtain efficient algorithms for certain problems [Jansson et al.; SIAM Journal on Computing (2006)].

２．研究の目的

The purpose of this project was to develop fast algorithms for working with huge phylogenetic trees and phylogenetic networks. It focused on three topics:
(A) Constructing different types of consensus trees;
(B) Identifying common sub- and superstructures of a set of phylogenetic trees;
(C) The combinatorics of phylogenetic networks.

Topic (A):
When faced with two or more identically leaf-labeled phylogenetic trees having contradicting branching structures, scientists often use a *consensus tree* to resolve the conflicts according to some well-defined mathematical criteria. Each type of consensus tree has some advantages and some disadvantages; see [Bryant; DIMACS Series in DMTCS (2003)] for a survey. Two of the most widely used consensus trees in practice are the *majority rule consensus tree* [Margush, McMorris; Bulletin of Mathematical Biology (1981)] and the *loose (semi-strict) consensus tree* [Bremer; Cladistics (1990)]. We previously developed deterministic algorithms [Jansson et al.; Proceedings of SODA 2013] and [Jansson et al.; Proceedings of RECOMB 2013] that achieved optimal running times for computing these two consensus trees, thus resolving two long-standing open problems. The goal of topic (A) was to continue this line of research to create fast algorithms for many other types of consensus trees as well.

Topic (B):
In topic (B), we wanted to investigate how the computational complexity of some basic supertree problems changes according to the problem definitions, e.g., by allowing non-binary trees or forbidden substructures to be specified in the input, and to design approximation algorithms for NP-hard problem variants. Furthermore, we wanted to design an exponential-time algorithm for the related optimization problem of inferring a *minimally resolved*

phylogenetic supertree (a phylogenetic tree having as few internal nodes as possible) from an input set of consistent *resolved triplets* (binary phylogenetic trees with exactly three leaves each). Finally, topic (B) addressed the issue of measuring the similarity between two phylogenetic trees with identical leaf label sets (but different branching structures) by computing the *rooted triplet distance* [Dobson; Lecture Notes in Mathematics (1975)]. This measure counts the number of *rooted triplets* (binary as well as non-binary phylogenetic trees with exactly three leaves each) that are embedded subtrees in either one of the input trees, but not the other; intuitively, two trees with a lot of branching structure in common will typically share many such subtrees.

Topic (C):

Here, our goal was to make a fast method for measuring the similarity between two input phylogenetic networks. We considered the important special case where the input networks are *galled trees*, which means that all their underlying cycles are disjoint. (Galled trees are sufficient in cases where a phylogenetic tree is not good enough but it is known that only a few reticulation events have happened [Gusfield et al.; Journal of Bioinformatics and Computational Biology (2004)].) As the measure of similarity, we assumed the extension of the rooted triplet distance from the phylogenetic tree setting to the phylogenetic network setting by [Gambette and Huber; Journal of Mathematical Biology (2012)]. The previously fastest known algorithm for computing this number relies on triangle counting and runs in O(n**2.687) time [Jansson and Lingas; Journal of Discrete Algorithms (2014)]. This is too slow when n is large, so in topic (C), we wanted to see if a combinatorial approach would lead to a faster algorithm.

３．研究の方法

For topic (A), we used techniques such as recursion, Day's algorithm, Boyer-Moore's majority algorithm, one-compatible clusters combined with filtering, radix sort, the principle of inclusion-exclusion, Apresjan clusters, linear-time preprocessing for answering level-ancestor queries in a tree in constant time, balanced binary search trees, and range minimum query data structures. In particular, to obtain our fast algorithm for the Adams consensus tree, we extended a wavelet tree-based technique for orthogonal range counting on a grid by [Bose et al.; Proceedings of WADS 2009] that may be of independent interest.

For topic (B), we used measure-preserving reductions between combinatorial problems, the BUILD algorithm [Aho et al.; SIAM Journal on Computing (1981)], smooth polynomial integer programs, label-to-bin assignments [Jiang et al.; SIAM Journal on Computing (2001)], decomposition trees, linearity of expectation, indicator variables, Semple's characterization [Semple; Discrete Applied Mathematics (2003)], and dynamic programming in trees. For computing the rooted triplet distance between two phylogenetic trees, we developed a fast and memory-efficient algorithm using the recent framework in [Brodal et al.; Proceedings of SODA 2013] but replaced the hierarchical decomposition tree by a simpler centroid paths-based solution.

For topic (C), our strategy was to transform the input into a constant number of pairs of trees so that the rooted triplet distance can be obtained by applying any existing algorithm for the simpler case of two phylogenetic trees a constant number of times. Basically, in any galled tree, removing one of the two edges leading to an indegree-2 vertex in every cycle yields a tree which still contains most of the branching information, and we showed how to compensate for what is lost by doing so while avoiding double-counting.

４．研究成果

Part (A):

We obtained new, deterministic algorithms for constructing several popular consensus trees. Given an input consisting of k phylogenetic trees with n leaves each and with identical leaf label sets, the worst-case running times of our algorithms are:

- O(k n) time [majority rule consensus tree],

- O(k n) time [loose consensus tree],

- O(k n**2) time [greedy consensus tree],

- min{O(k n**2), O(k n (k + log**2 n))} time

[frequency difference consensus tree],

- O(k n log n) [Adams consensus tree],

- O(n**2) [R* consensus tree with k=2],

- O(n**2 log**(4/3) n) [R* consensus tree with k=3],

- O(n**2 log**(k+2) n) [R* consensus tree with k>3],

- O(k n**3 + 2.733**n) [minimally resolved local consensus tree], and

- O(k n**3 + 4**n poly(n)) [minimally rooted-triplet-inducing consensus tree].

Some of our theoretical results provide the first improvements in a long time; for example, the previously fastest algorithm for the Adams consensus tree was from 1972. On the practical side, most of our new algorithms have been implemented and included in the FACT package.

Part (B):
For the generalized maximum rooted triplets consistency problem, we obtained a polynomial-time 1/4-approximation algorithm, an exact exponential-time algorithm whose running time depends on the degree of the output tree, and an exponential-time approximation scheme. We also provided a polynomial-time approximation scheme for the problem restricted to complete instances. For determining generalized consistency of rooted triplets, we characterized how the computational complexity changes under various restrictions, and presented a linear-time algorithm for dense inputs with no forbidden resolved triplets. For the minimally resolved supertree problem, we obtained an exact algorithm with O(2.733**n) time complexity (available in the FACT package), where n is the number of leaf labels. Finally, for computing the rooted triplet distance between two phylogenetic trees with n leaves, we obtained an O(n log**3 n)-time algorithm named 'CPDT-dist' that, although slower in theory than Brodal et al.'s state-of-the-art O(n log n)-time algorithm, was faster in practice for n <= 4,000,000 as well as less memory-consuming.

Part (C):
The idea described above led to a new algorithm named 'Galled-CPDT-dist' for computing the rooted triplet distance between two input galled trees with O(n log n) time complexity, where n is the size of the leaf label set. We implemented our algorithm, and applying it to pairs of randomly generated galled trees with up to 500,000 leaves confirmed that it is fast in practice.

５．主な発表論文等

〔雑誌論文〕（計 14 件）

1) J. Jansson, W.-K. Sung, H. Vu, and S.-M. Yiu:
Faster Algorithms for Computing the R* Consensus Tree (extended abstract),
in Proceedings of the 25th International Symposium on Algorithms and Computation (ISAAC 2014), *Lecture Notes in Computer Science*, Vol. 8889, pp. 414-425, 2014.
DOI: 10.1007/978-3-319-13075-0_33

2) J. Jansson, W.-K. Sung, H. Vu, and S.-M. Yiu:
Faster Algorithms for Computing the R* Consensus Tree,
Algorithmica, Vol. 76, No. 4, pp. 1224-1244, 2016.
DOI: 10.1007/s00453-016-0122-2

3) J. Jansson, Z. Li, and W.-K. Sung:
On Finding the Adams Consensus Tree (extended abstract),
in Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015), *LIPIcs*, Vol. 30, pp. 487-499, 2015.
DOI: 10.4230/LIPIcs.STACS.2015.487

4) J. Jansson, Z. Li, and W.-K. Sung:
On Finding the Adams Consensus Tree,
Information and Computation, Vol. 256, pp. 334-347, 2017.
DOI: 10.1016/j.ic.2017.08.002

5) J. Jansson and W.-K. Sung:
Minimal Phylogenetic Supertrees and Local Consensus Trees (extended abstract),
in Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016), *LIPIcs*, Vol. 58, pp. 53:1-53:14, 2016.
DOI: 10.4230/LIPIcs.MFCS.2016.53

6) J. Jansson, R. Rajaby, and W.-K. Sung:
Minimal Phylogenetic Supertrees and Local Consensus Trees,
AIMS Medical Science, Vol. 5, No. 2, pp.

181-203, 2018.
DOI: 10.3934/medsci.2018.2.181

7) J. Jansson, C. Shen, and W.-K. Sung:
Improved Algorithms for Constructing Consensus Trees,
Journal of the ACM, Vol. 63, No. 3, Article 28, 2016.
DOI: 10.1145/2925985

8) J. Jansson, R. Rajaby, C. Shen, and W.-K. Sung:
Algorithms for the Majority Rule (+) Consensus Tree and the Frequency Difference Consensus Tree,
IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 15, No. 1, pp. 15-26, 2018.
DOI: 10.1109/TCBB.2016.2609923

9) J. Jansson, A. Lingas, and E.-M. Lundell:
The Approximability of Maximum Rooted Triplets Consistency with Fan Triplets and Forbidden Triplets (extended abstract),
in Proceedings of the 26th Annual Symposium on Combinatorial Pattern Matching (CPM 2015), *Lecture Notes in Computer Science*, Vol. 9133, pp. 272-283, 2015.
DOI: 10.1007/978-3-319-19929-0_23

10) J. Jansson, A. Lingas, R. Rajaby, and W.-K. Sung:
Determining the Consistency of Resolved Triplets and Fan Triplets (extended abstract),
in Proceedings of the 21st Annual International Conference on Research in Computational Molecular Biology (RECOMB 2017), *Lecture Notes in Computer Science*, Vol. 10229, pp. 82-98, 2017.
DOI: 10.1007/978-3-319-56970-3_6

11) J. Jansson, A. Lingas, R. Rajaby, and W.-K. Sung:
Determining the Consistency of Resolved Triplets and Fan Triplets,
Journal of Computational Biology, Vol. 25, 2018.
DOI: 10.1089/cmb.2017.0256

12) J. Jansson and R. Rajaby:
A More Practical Algorithm for the Rooted Triplet Distance (extended abstract),
in Proceedings of the 2nd International Conference on Algorithms for Computational Biology (AlCoB 2015), *Lecture Notes in Computer Science*, Vol. 9199, pp. 109-125, 2015.

13) J. Jansson and R. Rajaby:
A More Practical Algorithm for the Rooted Triplet Distance,
Journal of Computational Biology, Vol. 24, No. 2, pp. 106-126, 2017.
DOI: 10.1089/cmb.2016.0185

14) J. Jansson, R. Rajaby, and W.-K. Sung:
An Efficient Algorithm for the Rooted Triplet Distance between Galled Trees (extended abstract),
in Proceedings of the 4th International Conference on Algorithms for Computational Biology (AlCoB 2017), *Lecture Notes in Computer Science*, Vol. 10252, pp. 115-126, 2017.
DOI: 10.1007/978-3-319-58163-7_8

〔学会発表〕（計 6 件）

Faster Algorithms for Computing the R* Consensus Tree
2014-12-16
Jeonju, South Korea

On Finding the Adams Consensus Tree
2015-03-05
Munich, Germany

The Approximability of Maximum Rooted Triplets Consistency with Fan Triplets and Forbidden Triplets
CPM 2015
2015-06-30
Ischia Island, Italy

A More Practical Algorithm for the Rooted Triplet Distance
AlCoB 2015
2015-08-04
Mexico City, Mexico

Comparing Phylogenetic Networks by Counting Triangles
NUS workshop (not refereed)
2015-07-27
Singapore
[Invited lecture]

Minimal Phylogenetic Supertrees and Local Consensus Trees
MFCS 2016
2016-08-22
Krakow, Poland

〔図書〕（計 5 件）

1) J. Jansson and W.-K. Sung:
Algorithms for Combining Rooted Triplets into a Galled Phylogenetic Network,
in M.-Y. Kao (editor), *Encyclopedia of Algorithms (Second Edition)*, pp. 48-52, Springer Science+Business Media New York, 2016.
DOI: 10.1007/978-3-642-27848-8_92-2

2) J. Jansson:
Directed Perfect Phylogeny (Binary Characters),
in M.-Y. Kao (editor), *Encyclopedia of Algorithms (Second Edition)*, pp. 553-556, Springer Science+Business Media New York, 2016.
DOI: 10.1007/978-3-642-27848-8_112-2

3) J. Jansson and W.-K. Sung:
Maximum Agreement Supertree,
in M.-Y. Kao (editor), *Encyclopedia of Algorithms (Second Edition)*, pp. 1224-1227, Springer Science+Business Media New York, 2016.
DOI: 10.1007/978-3-642-27848-8_222-2

4) J. Jansson:
Perfect Phylogeny (Bounded Number of States),
in M.-Y. Kao (editor), *Encyclopedia of Algorithms (Second Edition)*, pp. 1550-1553, Springer Science+Business Media New York, 2016.
DOI: 10.1007/978-3-642-27848-8_288-2

5) J. Jansson:
Phylogenetic Tree Construction from a Distance Matrix,
in M.-Y. Kao (editor), *Encyclopedia of Algorithms (Second Edition)*, pp. 1564-1567, Springer Science+Business Media New York, 2016.
DOI: 10.1007/978-3-642-27848-8_292-2

〔産業財産権〕

None.

〔その他〕
ホームページ等

Jesper Jansson's webpage:
http://www.comp.polyu.edu.hk/~csjj/

FACT: Fast Algorithms for Consensus Trees:
http://compbio.ddns.comp.nus.edu.sg/~consensus.tree/

CPDT-dist:
http://sunflower.kuicr.kyoto-u.ac.jp/~jj/Software/CPDT-dist.html

Galled-CPDT-dist:
https://github.com/Mesh89/Galled-CPDT-dist

６．研究組織
(1)研究代表者

Jesper Jansson
Specific Associate Professor,
The Hakubi Project,
Kyoto University.

Researcher Number: 60536100