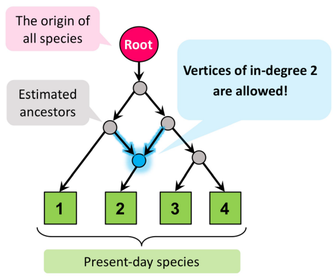
Tree-based phylogenetic network (TBN) の構造定理が導く色々な高速アルゴリズム

早水 桃子　モデリング研究系　助教

Glossary (informal)

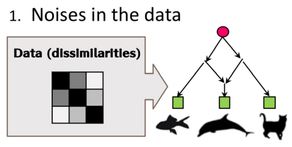
'Phylogenetic network'

✓ **Generalisation of phylogenetic trees**

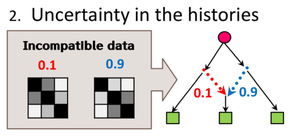


✓ **Useful to represent...**

1. Noises in the data



2. Uncertainty in the histories



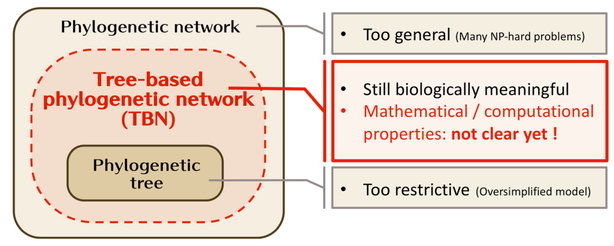
✓ **Drawback**: Flexible, but there are many NP-hard problems!

Glossary (informal)

'Tree-based phylogenetic network (TBN)'

✓ Subclass of phylogenetic networks [Francis & Steel, 2015]

✓ Studied by many combinatorialists and theoretical computer scientists



Abstract

■ **Goal of this study**

- To develop a **framework** for solving various important problems on TBNs
- To derive many fast algorithms **in a unified manner**

■ **Results**

Main theorem

Structural characterisation of TBNs

Algorithms for various problems

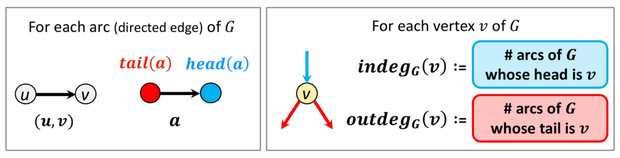
Decision/search	⬅ Linear time
Counting	⬅ Linear time
Enumeration	⬅ Linear time delay
Optimisation	⬅ Linear time

Preprint M. Hayamizu: 'A structure theorem for tree-based phylogenetic networks', arXiv:1811.05849 [math.CO]

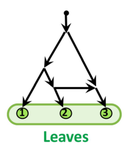
Notation & definitions

■ **Terminology**

- **Graph / Network** ⋯ All refer to **finite**, **simple**, **directed** **acyclic** graphs.
- For a graph G, $V(G)$: its vertex-set, $V(G)$: its arc-set
- A graph with a set V of vertices and a set A of arcs is denoted by (V, A).



- A vertex v of a graph G is called a **leaf** of G if $(\boldsymbol{indeg_G(v), outdeg_G(v)}) = (\mathbf{1}, \mathbf{0})$ holds.
- A graph G is said to be **binary** if for each internal vertex v of G, $(\boldsymbol{indeg_G(v), outdeg_G(v)}) =$ $(\mathbf{1}, \mathbf{2})$ or $(\mathbf{2}, \mathbf{1})$ holds.



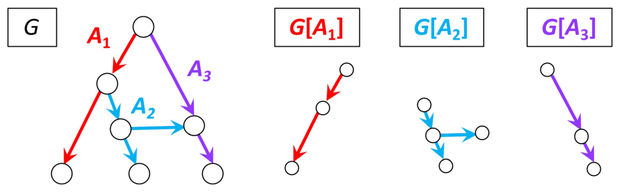
Notation & definitions

■ **Arc-induced subgraph**

- For a graph G, $A' \subseteq A(G)$ is said to **induce the subgraph $G[A']$:= $(V(A'), A')$ of G**, where $V(A')$ denotes the set of all ends of the arcs in A'.

■ **Decomposition of a graph**

- Given a graph G and a partition $\{A_1, \ldots, A_l\}$ of $A(G)$, the collection $\{G[A_1], \ldots, G[A_l]\}$ of arc-induced subgraphs of G is called a **decomposition** of G.

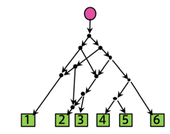


Notation & definitions

■ **X: a non-empty finite set** {1,..., n}

■ **A rooted binary phylogenetic X-network**

is defined to be a DAG that satisfies:



✓ ∃! root (or)

✓ each internal vertex is or

✓ X = the set of leaves

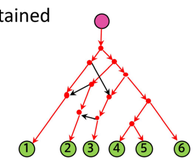
- A rooted binary phylogenetic X-tree is the special case when $\nexists$.

➢ For short...
- $\mathcal{T}_X$:= a collection of all rooted binary phylogenetic X-trees
- $\mathcal{N}_X$:= a collection of all rooted binary phylogenetic X-networks

Notation & definitions

■ **Tree-based phylogenetic network (TBN)** [Francis & Steel, 2015]

- **Intuition:** a phylogenetic tree with additional arcs
- **Definition:** $N \in \mathcal{N}_X$ is called a **TBN** if N can be obtained by the following procedure:
  1. Start with an arbitrary element of $\mathcal{T}_X$
  2. Subdivide each arc zero or more times
  3. Place vertex-disjoint arcs between new vertices
  4. Smooth all v with indeg(v)= outdeg(v) = 1



■ **Equivalent definition** [H, 2018]

- A **subdivision tree of N** is defined to be a spanning tree T of N s.t. T is a subdivision of some element of $\mathcal{T}_X$
- $N \in \mathcal{N}_X$ is a **TBN** if N has one or more subdivision trees.

Problem description
[Francis and Steel (2015)]

■ **① Decision/search problem** [Francis & Steel, 2015]

- Given a rooted binary phylogenetic X-network $N \in \mathcal{N}_X$, find a subdivision tree of N if N is a TBN, and report that 'N is not a TBN' otherwise.

■ **Motivation**

- Biologist often wish to discover an underlying tree T of N; however, such a tree T might not exist for some N. (*i.e.*, TBNs form a 'proper' subclass of $\mathcal{N}_X$)

■ **Known results**

- ∃ a linear time algorithm for this decision/search problem
  (∵ It can be formulated as the 2-SAT problem)
  (∵ Alternatively, from Hall's marriage theorem [Zhang, 2016])

Problem description
[Francis and Steel (2015)]

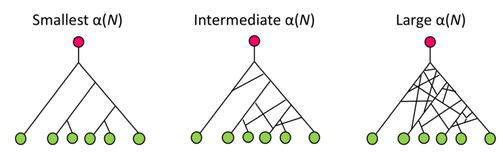
■ **② Counting problem**

- Given a rooted binary phylogenetic X-network $N \in \mathcal{N}_X$, determine the number $\alpha(N)$ of subdivision trees of N.

✓ Francis & Steel (2015) conjectured that counting $\alpha(N)$ 'might be hard'. (because counting # of solutions of 2-SAT is #P-complete.)

■ **Motivation**

- $\alpha(N)$ may be useful to quantify the **structural complexity** of N.



Problem description
[Francis and Steel (2015)]

■ **③ Enumeration problem**

- Given a rooted binary phylogenetic X-network $N \in \mathcal{N}_X$, list all subdivision trees $T_1, \ldots, T_{\alpha(N)}$ of N.

■ **Motivation**

- Useful for **uniform sampling** of subdivision trees of N

■ **Known results**

- The number $\alpha(N)$ of solutions can be exponential in size of N (but details are unclear)

Problem description
[H (2018)]

■ **④ Optimisation problem**

- Given a rooted binary phylogenetic X-network $N \in \mathcal{N}_X$ associated with a weighting function w: $A(T) \to \mathbb{R}_{>0}$, find a subdivision tree T^* of N that maximises the value of $f = \sum_{a \in A(N)} w(a)$.

■ **Motivation**

- Estimation of the 'most likely' tree within the input N
  - ✓ $w(a)$: probability of each arc $a \in A(N)$
  - ✓ $f(T)$: the likelihood of a subdivision tree T

■ **Known results**

- Recall: the number $\alpha(N)$ of subdivision trees can be exponential.
  (➡ Exhaustive search takes exponential time! Any efficient method?)

Useful result
[Francis & Steel (2015)]

Definition

Given a rooted binary phylogenetic X-network N and a subset S ⊆ A(N), S is called an **admissible subset of A(N)** if S satisfies the following conditions:

- **Condition ①** $\text{indeg}_N(v) = 1 \vee \text{outdeg}_N(u) = 1 \Rightarrow$ S contains (u,v)
- **Condition ②** $\text{head}(a_1) = \text{head}(a_2) \Rightarrow$ S contains exactly one arc in $\{a_1, a_2\}$
- **Condition ③** $\text{tail}(a_1) = \text{tail}(a_2) \Rightarrow$ S contains at least one arc in $\{a_1, a_2\}$.

Theorem

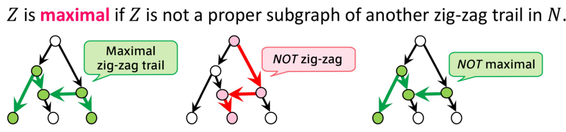
Given a rooted binary phylogenetic X-network N and its (spanning) subtree T, T is a subdivision tree of N if and only if $A(T)$ is an **admissible subset of $A(N)$**.

The collection of subdivision trees of N $\mathcal{T} := \{T_1, \ldots, T_{\alpha(N)}\}$ ⟷ (bijection) The family of admissible subsets of $A(N)$ $\mathcal{A} = \{A(T_1), \ldots, A(T_{\alpha(N)})\}$

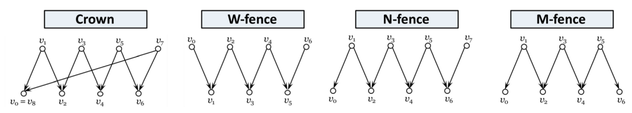
Key ideas

■ **Def. (Maximal zig-zag trail)**

- A subgraph Z of N with $m \geq 1$ arcs is called a **zig-zag trail in N** if there exists a permutation $(a_1, \ldots, a_m)$ of $A(Z)$ such that for any $i \in [1, m-1]$, either $head(a_i) = head(a_{i+1})$ or $tail(a_i) = tail(a_{i+1})$ holds.
  - ➢ Z is **maximal** if Z is not a proper subgraph of another zig-zag trail in N.



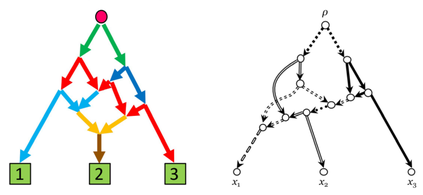
■ **The four types of maximal zig-zag trails**



Key ideas

■ **Decomposition lemma** [H, 2018]

- For any rooted binary phylogenetic X-network N, there exists a unique decomposition $\{Z_1, \ldots, Z_l\}$ of N s.t. each Z_i is a maximal zig-zag trail in N.



■ **Structural analogue of Francis & Steel's theorem**

- S ⊆ A(N) is an admissible subset of A(N) ⇔ ∀ maximal zig-zag trail Z_i in N, S ∩ $A(Z_i)$ is an admissible subset of $A(Z_i)$.

Main result

■ **Structure theorem for TBNs** [H, 2018]

N : a rooted binary phylogenetic X-network

$\mathcal{Z} = (Z_1, \ldots, Z_l)$: an (arbitrarily) ordered set of the maximal zig-zag trails of N.

- N is a TBN ⇔ No element Z_i of $\mathcal{Z}$ is a W-fence.
- N is a TBN ⇒ The family of admissible subsets of A(N) is characterised by the direct product of $\mathscr{S}(Z_i)$, where

$$\mathscr{S}(Z_i) := \begin{cases} \{\langle (10)^{m_i/2} \rangle, \langle (01)^{m_i/2} \rangle\} & \text{if } Z_i \text{ is a crown;} \\ \{\langle (1(01)^{(m_i-1)/2} \rangle\} & \text{if } Z_i \text{ is an N-fence;} \\ \{\langle (1(01)^p (10)^q 1 \rangle \mid p, q \in \mathbb{Z}_{\geq 0}, p+q = (m_i-2)/2\} & \text{if } Z_i \text{ is an M-fence.} \end{cases}$$

(or)

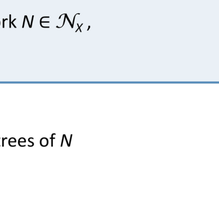
There exist $m_i := |A(Z_i)|$ admissible subsets.

Corollaries

■ **Algorithms** [H, 2018]

- α(N) = α(Z_1)× ・・・ ×α(Z_l), where

$$\alpha(Z_i) = \begin{cases} 0 & \text{if } Z_i \text{ is a W-fence;} \\ 1 & \text{if } Z_i \text{ is an N-fence;} \\ 2 & \text{if } Z_i \text{ is a crown;} \\ |A(Z_i)|/2 & \text{if } Z_i \text{ is an M-fence.} \end{cases}$$

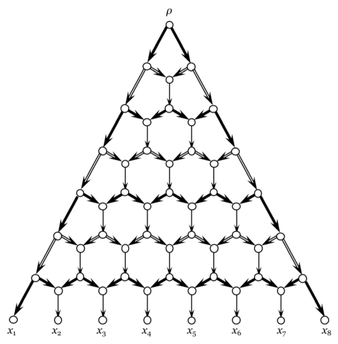


✓ ∴ The **counting problem** can be solved in O(|A(N)|) time.
(We can solve the **decision/search problem** simultaneously.)

- Similarly, the **enumeration problem** can be solved in O(k|A(N)|) time, where k is the number of solutions we want to list.
- Moreover, the **optimisation problem** can be solved in O(|A(N)|) time.
(∵ we can automatically get the global optima by gathering local optima.)

Numerical example

■ **Quantifying the complexity of N by counting α(N)**



- **Maximal zig-zag trail decomposition**
  - ✓ # maximal N-fences: 21
  - ✓ # maximal M-fences: 7

  α(N) = 7*6*5*4*3*2*1 = 5040
- **Advantage of exact computation**
  - Compare the above number with a trivial upper bound 2^{21}=2097152.
- **How complex is N?**
  - The above number is still smaller than # of the rooted binary phylogenetic X-trees, given by (2|X|-3)!! =13*11*...*5*3*1 =135135.
    (∴ N is not complex enough to cover all possible evolutionary scenarios.)

Summary

Problem	Input	Output	Time complexity	Potential applications
Decision/ Search	N: phylogenetic network	■ Whether or not N is tree-based ■ If N is tree-based, find a tree inside N	Linear [Francis & Steel, 2015] (Proof based on 2-SAT) [Zhang, 2016] (Proof based on Hall's marriage theorem) **[H, 2018]** (Proof based on a structure theorem for TBNs)	Checking whether the data can be explained by a phylogenetic tree + additional arcs
Counting		■ # of all possible trees	**Linear [H, 2018]** (Proof based on a structure theorem for TBNs)	Quantifying the structural complexity of N
Enumeration (Listing)		■ Set of all possible trees	**Linear-delay [H, 2018]** (Proof based on a structure theorem for TBNs)	Uniform sampling of trees inside N
Optimisation	N: phylogenetic network w: associated weighting (>0) (e.g., probability)	■ Tree to maximise a prescribed objective function f (e.g., likelihood)	**Linear [H, 2018]** (Proof via a structure theorem for TBNs)	Estimating the 'best-fit' tree inside N

大学共同利用機関法人 情報・システム研究機構
統計数理研究所
The Institute of Statistical Mathematics