2023 年度数学特別セミナー
組合せ論的系統学入門

(Introduction to Combinatorial Phylogenetics)

述: 早水 桃子 [*]/ 記: 穂坂 秀昭 [†]

2024 年 3 月

概要

開成学園では毎年，現役の数学者に講演していただく「数学特別セミナー」を開催しています．2023 年度の 1 回目は，早稲田大学理工学術院の早水桃子先生をお招きし，「組合せ論的系統学入門」というテーマでセミナーをしていただきました．このノートは，その講演の内容をまとめたものです．

なお，脚注はいずれも穂坂が付け足したものです．早水先生にお話していただいた内容は全て本文中に含めています．また，このノートの文責は，誤りも含め，全て穂坂にあります．

目次

[*] 早稲田大学理工学術院 准教授

[†] 開成中学校・高等学校 数学科 教諭

0　今回のセミナーの概要

「系統樹」は生物たちが共通の祖先からどのように分岐してきたかという進化史を記述する最も基本的で重要なモデルだが，実際のデータや現象は一つの樹形図で説明できるほど単純ではないため，系統樹を一般化した「系統ネットワーク」を活用して高度な系統解析を行う新技術も必要とされている．しかし，分岐しか表せない系統樹とは異なり，系統ネットワークは非常に複雑なグラフを含む広大なグラフのクラスであるため，その数学的・計算機科学的な性質についてはまだ多くのことが解明されていない．

この講義では，講師自身による組合せ論的系統学の研究成果の一例として，複雑な系統ネットワークに含まれる系統樹の数え上げ，列挙，最適化といった多様な計算問題を統一的に取り扱うことを可能にし，各問題に対する高速アルゴリズムを導出して幾つかの未解決問題を解決した「系統ネットワークの構造定理」を分かりやすく解説する．

1　はじめに

■座長より早水先生のご紹介　それでは本日の数学特別セミナーを開催したいと思います．本日は早稲田大学より早水先生にお越しいただいて，「組合せ論的系統学入門」というタイトルでお話いただきます．

早水先生と僕（穂坂）はもうだいぶ昔からの古い仲で，多分，学部 3 年ぐらいのときからの長い付き合いですよね．当時僕は理学部数学科にいて，早水先生は医学部医学科にいらしたんですけど，その時から統計にすごく興味お持ちなんだなという印象がありました．『統計学を拓いた異才たち』をはじめ，何冊か本を早水先生にご紹介していただいた記憶があります．その早水先生はお医者さんになられた後，今こうして数学者になられ，現在では数学と医学の境界領域で活躍なさっている第一人者です．ぜひ，生物の方にも数学の方にも話を楽しんで聞いていただけたらと思います．

それでは早水先生，どうぞよろしくお願いいたします．[会場より拍手]

■ご挨拶　ご紹介ありがとうございます．みなさん初めましてこんにちは．早稲田大学基幹理工学部の応用数理学科というところで教員をしております早水と申します．今紹介してくださった通り，穂坂先生とは学部こそ違えど大学の同級生という感じで古い仲ですので，今回こういう場所，こういう形でお招きいただいて皆さんの前でレクチャーをさせてもらえる機会をいただけて嬉しく思っております．

私は数学の中でも離散数学，いわゆる組合せ論という分野を専門にしています．元々のバックグラウンドが医学ということもあり，とりわけ「生命科学に関する組合せ論」を専門にしております．今日は「組合せ論的系統学」という私の研究分野の話をします．皆さん全然聞いたことがなく「何それ？」という感じだと思うので，ごくごく初歩的なことから始めて，「私がこういう研究をしている」ところまで説明をします．「こんな分野があるんだ」とか「面白いな」って思ってもらえたらいいなと思っております．気楽に聞いてください．

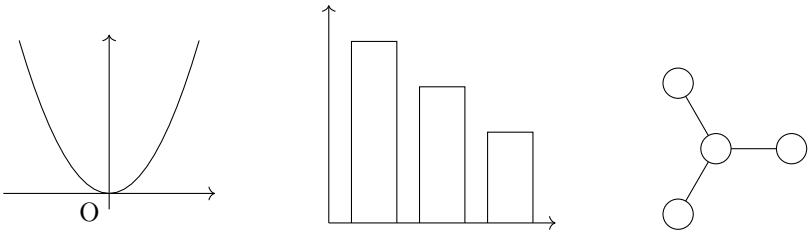
■系統学とは　組合せ論的系統学とは何なのかと言いますと，生物の進化の系統解析に関する組合せ論とかアルゴリズムを研究する分野です．ただ，一応「生物の進化に関する話」と言ったんですけれども，別に生物に限った話じゃないです．皆さんは日常生活でも「系統」っていう言葉を割と使うと思うんですよ．「赤とオレンジは同じ系統の色だ」とか「なんか似た系統の服ばっかり着てる」とか．ですので系統の解析は，生物の進化に全然限らず，結構いろんな場面で出てくる話です．

組合せ論的系統学の応用は，特に生物以外ですと，例えば文化の系統樹とか言語の系統解析とかですね．皆さん歴史とか勉強すると多分出てくるんですけど，日本の文化には中国をはじめ，海外から伝わってきたことがいろいろあり，純粋にジャパニーズだけでできてきたわけじゃないんです．そんなわけで，日本の文化がどういうところから流入して影響を受けて今に至ってるかという系統の解析に組合せ論的系統学が使われます．あと言語の語族というのがあります．ドイツ語とかフランス語とか英語とかについて「なんとか語族」っていう，この言語は元々どこから来たかという言語の解析にも使われていたりします．

そういう系統解析の理論的な研究，方法論を作るっていう研究には数学的にいろんなアプローチがあります．代数でアプローチする人もいれば，幾何学のアプローチで研究してる人もいたりとか，統計学的な研究もあったりとか，すごくいろいろあるんですよ．私は専門にしてるのは組合せ論という分野でして，組合せ論的系統学，英語だと combinatorial phylogenetics って呼ばれています．

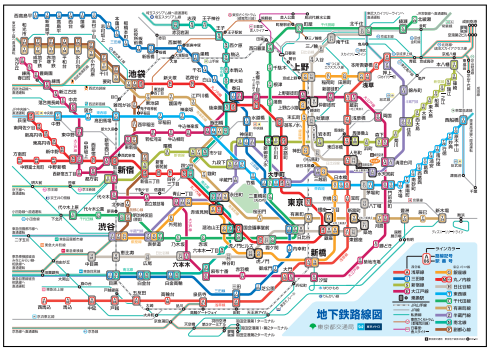
■組合せ論とグラフ理論　そもそも組合せ論が何かというと，ざっくり言うと有限個のもの，離散的なものとか有限個のものを扱う数学の分野です．有限のものは，皆さんには結構お馴染みの話なんですよね．一番馴染みがある例は場合の数ですね．有限個の物事が何通りかあって，それを数えなさいっていう数える問題だったりとか．あとは高校でやるかどうかあまりわからないんですけど，いくつかの有限個のものがある中でベストなものを選びなさいみたいな問題，最適化っていうような分野があったりとか．そういうのも組合せ論の一部となっています．

例えばグラフ理論と呼ばれるものが，組合せ論の一つの分野と言えます．ここで言ってる「グラフ」は，「関数のグラフを書きなさい」みたいなグラフのことじゃありません．点と線の集まりみたいなもののことを，グラフって言ってます．高校の段階だと多分，グラフというと放物線みたいなものをイメージすると思うんですけど，今日はもっぱらこっちの意味で使っています．



左から順に「2 次関数 $y = x^2$ のグラフ」「棒グラフ」「今日考えるグラフ」

典型的には地下鉄の路線図ですよね．本当はすごく複雑な地図の中にいろんな地点があるわけだけれども，駅を点とみなして，路線が走ってるところを線でつないでいる．路線図は典型的なグラフの例といえます．



東京メトロの路線図

グラフに関する組合せの問題っていうのはどういうものがあるのかというと，たとえば「西日暮里駅から西早稲田駅までの最短経路を求めなさい」みたいな最適化問題ですね．一番安いやつとかね．有限個の物事を扱ってるので可能な候補の数も有限個しかないわけなので，「じゃあしらみつぶしにあり得る行き方を全部見つけて最短のやつ選べばいいじゃん」と思うかもしれません．ただ，西日暮里から西早稲田ならそれでいいかもしれないんですけど，普通の問題では候補の数がものすごく膨大にある．「有限だからって全部しらみつぶしで数えたり探したりすればいいわけじゃないよ」ということで，効率的なアルゴリズムを作ることが大事になってくる．すごく現実の問題があるわけです．西日暮里からどこそこまで最短で行きたいとか，一番安く行きたいとか，そういうのを実際にグラフの言葉で定式化して，その「最短で行きたい」というのを最適化問題の形で書き直して，それから問題を解くアルゴリズムを考える．そういうのがグラフ理論とその応用の基本的な研究のスタンスってことになります．

今言ったことはですね，私がすごく大好きな言葉なんです．グラフ理論の父と言われる Frank Harary という人がいます．Harary 先生自身は応用とかそういうことを全くやってない，なんというか純粋グラフ理論とでも言うべき純粋数学の研究者なんです．その Harary 先生が “Distance in Graphs”，『グラフの中の距離』っていう本[*1] の前書きでですね「グラフ理論っていうのはこういう分野だよ」というのを記述しているところがあります．

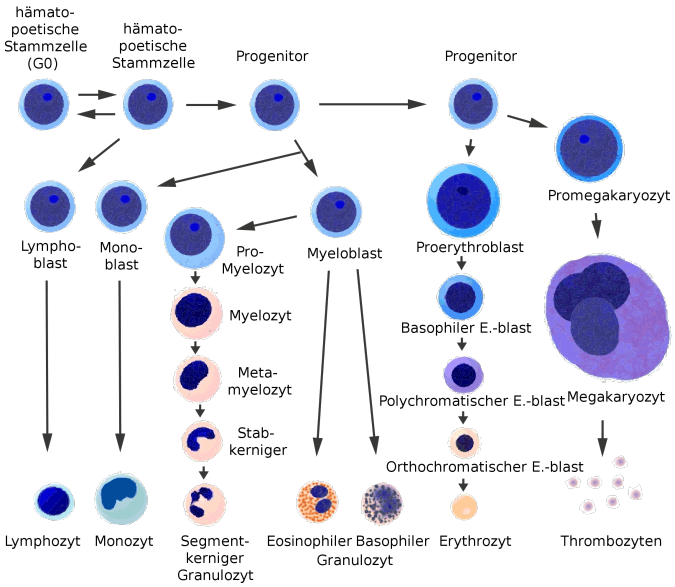
> Graph theory is a branch of mathematics which has applications in many areas: anthropology, architecture, biology, chemistry, computer science, economics, environmental conservation, psychology, and telecommunications, to name a few. The list goes on and on. In a typical situation, a problem arises in a real-world subject area that can be modeled using graphs. Then existing theorems or algorithms are used or new ones are developed to solve the original problem.
>
> （グラフ理論は数学の一分野であり，人類学，建築学，生物学，化学，コンピュータサイエンス，経済学，環境保全，心理学，通信学などをはじめとして，非常に多くの分野に応用されている．そういった現実の諸分野では，グラフを使ってモデル化できる問題が現れることがよくある．それで，その現実の問題を解決するために，グラフ理論における既存の定理やアルゴリズムが活用されたり，あるいは新たな定理やアルゴリズムが生まれたりするのだ．）

現実のいろんな分野では，さっきの乗り換え検索みたいなグラフを使ってモデル化できる問題が現れることがよくある．グラフを使って問題を抽象化し，グラフ理論の既存の定理とかアルゴリズムを使ったり，あるいは，ない場合はそれを自分で作り出したりとかする．それがグラフ理論の研究の営みなんだっていうようなことを書いている．まさに私の研究のスタンスでもあるかなって思います．

[*1] F. Buckley & F. Harary, “Ditstance in Graphs,” Addison–Wesley Publishing Company, Advanced Book Program, Redwood City, CA, 1990.

■生物学に現れるグラフ　じゃあそうすると「乗り換え検索との話はわかったんだけど，生命科学，医学とか生物学においてグラフには何の関係があるのか」って思うかもしれません．もしかしたらもう既に生化学とか分子生物学とか細胞生物学とか勉強している人がいるかもしれないんですけど，実は大学に入っていろんな生物学の本を開くと，びっくりするぐらい，さっきの路線図みたいなグラフで溢れてるんですよ．

たとえば私は今日は進化の話をしますけど，もう 1 個やってるテーマとして，細胞の分化があります．細胞の分化のツリーって，細胞生物学の中ですごくたくさん出てくる図なんです．

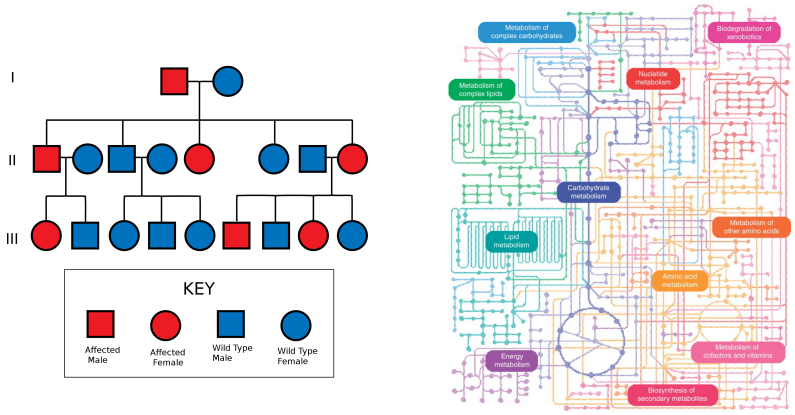


細胞分化のツリー[*2]

細胞の分化って言うんですけど，一番最初に何にでもなれる受精卵みたいな細胞，この絵だと Stammzelle と書かれた幹細胞があって，そこから枝分かれして何かになっていくわけですね．皆さんの身体ってものすごくいろんな細胞からできてるじゃないですか．髪の毛の細胞もある，皮膚の細胞もある，脳細胞もある，筋肉の細胞もあるなって．そういういろんな細胞があるんだけど，元々は皆さん 1 個の受精卵だったわけですよね．それが色々に分化して異なる細胞になっていって，変化を遂げていって今の多様性があるわけですよ．動物の細胞の場合，その枝分かれした後に最初に戻ってグルグル循環したりすることはないんで，細胞の分化は分岐する木構造というので表せる．ということでこのツリー構造が生物の教科書にものすごく出てきます．これが植物だと，枝分かれで記述できるとは限らないんです．我々の腕ってちぎったりしたら別に生えてこないですけど，植物って断ち切っても生えてくるじゃないですか．理科で勉強した人がいるかもしれないんですけど，あれは元に戻ってるんですね．脱分化といって，一旦分化した後に逆行して初期状態に戻るということが起きてまた分化をやり直すことができる．ですので植物の場合は話がより複雑になるんですけど，動物のヒトとかの細胞の場合は，こんな風にツリー構造で書けるということが知られているわけです．

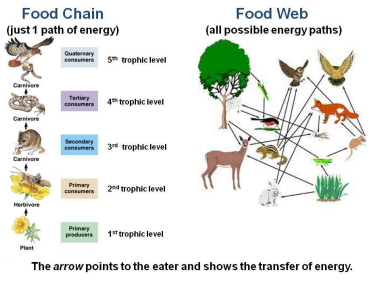
[*2] Wikimedia Commons contributors, “File:Blood cells differentiation chart de.svg,” Wikimedia Commons, https://commons.wikimedia.org/w/index.php?title=File:Blood_cells_differentiation_chart_de.svg&oldid=826611328 （2024/1/21 閲覧）.

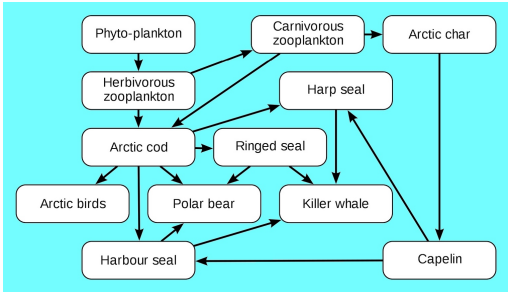
また私自身の専門ではないんですけれども，遺伝性疾患の家系図も医学に出てくるグラフの例です．病気を発症した患者さんの血縁者や先祖の中で誰が発症して誰が発症しなかったかを調べると左のようになります．このような家系図もグラフなので家系図のグラフ理論というのも研究されてたりします．あとは生化学で出てくる，細胞の中の代謝経路のネットワークとか，シグナル伝達回路とか経路とかね．こういうのも，グラフの典型的な例ですね．



左: 遺伝性疾患の家系図[*3] / 右: 代謝経路ネットワーク[*4]

生態学の話で言うと，食物連鎖．食う食われるの関係がありますよね．それは food chain っていわれる鎖，向きのある変化なんですけれども，より一般的な概念として「食物網」というネットワークの話があります．意外なことに，これ，私は数学者に教えてもらったんですよね．「Food chain っていうのはすごく単純化しすぎた話で，食物網 food web というのがより適切なんだ」って．特に，砂漠みたいなものすごく食べ物が枯渇した場所だと，食う食われるのきれいな関係があるわけじゃなくて，お互いがお互いを食べあって生きていくみたいなすごい複雑なネットワークができてたりするらしくて．それで土の中の栄養状態の貧しさ豊かさがわかるとかいう話もあったりするみたいです．こういう生態学に関するグラフも研究されたりします．





左: Simple path of energy from the producers to the consumers...[*5]
右: 北極の Food web[*6]

[*3] Wikimedia Commons contributors, "File:Autosomal Dominant Pedigree Chart.svg," Wikimedia Commons, https://commons.wikimedia.org/w/index.php?title=File:Autosomal_Dominant_Pedigree_Chart.svg&oldid=808746259 (2023/11/20 閲覧).

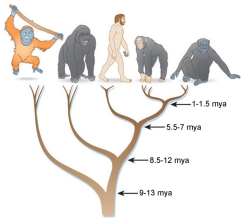
[*4] https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/metabolic-network (2023/11/20 閲覧)

[*5] https://www.thinglink.com/scene/786024112798564354 (2023/11/20 閲覧)

[*6] Wikimedia Commons contributors, "File:Arctic food web.jpg," Wikimedia Commons, https://commons.wikimedia.org/w/index.php?title=File:Arctic_food_web.jpg&oldid=801078827（2023/11/20 閲覧）

■系統樹と系統ネットワーク　こういう色んな例で「生物学の中にいろんなグラフが出てくるよ」という話をしたんですけど，今日の話は系統樹と系統ネットワークっていう，進化に関する話です．

系統ネットワークは多分初耳だと思うんですけど，進化の系統樹に関しては多分皆さん見たことがあるんじゃないかなって思います．明示的に系統樹って言ってるかどうかわかんないですけど，多分中学校の理科の教科書とかで見たことはあると思うんですよね．ヒトとかサルとかチンパンジーの進化の分岐の絵ですね．

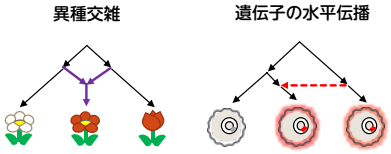


ヒト科の分岐[*7]

でも今回考えていく対象は，系統樹だけじゃなく，系統ネットワークっていうものです．後でちゃんと説明しますけど，系統ネットワークというのは系統樹を一般化した，系統樹だけでは表せないようなものを表すために作られた概念なんですね．系統ネットワークを使うと，いろんなものが記述できる．

一番わかりやすいのは異種交雑というやつです．農作物の品種改良みたいに，違う種類になって分解したものをまた掛け合わせて新しい種ができるみたいな合流は，分岐だけでは表せません．異種交雑による新種の出現という現象の記述には，系統樹では不足してるわけですよ．そういう時には系統ネットワークが使える．

あとは遺伝子の水平伝播ですね．これは細菌，バクテリアで起きたりする話です．例えば皆さん知ってるO157，ベロ毒素というのを作る大腸菌があります．食中毒の原因になるやつですね．普通の大腸菌はその辺にいっぱいいますけど，それらと違って突然変異によって遺伝子を獲得した中で，ベロ毒素を作る悪いやつがいるわけです．O157 って 1 匹いるだけだったら別に大して影響はしないかもしれないけど，他の大腸菌を引きずり込んで O157 にするんですね．我々って自分以外のヒトに遺伝子を伝えようと思うと，やり方としては，子供を作って子供に伝えるしかないですよね．あなたに伝えようと思ってもできないし，兄弟がすごくいい遺伝子を持ってたとしても「なんかいいじゃん」とか言ってもらえたりしないじゃないですか．ところが細菌ってこれができるんですよ．なので我々が思ってるような「家系」って，お父さんお母さんみたいな血統とか血縁関係じゃないんです．バクテリアの世界っていうのはその辺にいるバクテリアから遺伝子をもらったりとか交換したりとか，死んでるやつからもらったりすらできちゃう．そういう時ってグラフが分岐して，流入する・奪うみたいな線が余計に入ることになるので，普通の系統樹では表せない現象ということになります．



こんなわけで自然界には，系統樹で表せないような新しい進化，複雑な進化の現象っていうのがすごくいろいろある．系統樹では不足してるから，その上位互換で記述能力を高めた系統ネットワークというのを使おうという流れがあります．今日はその話をしていきます．

[*7] Mitchell, M. W. & Gonder, M. K. (2013), "Primate speciation: A case study of African apes," *Nature Education Knowledge* **4**(2):1, Figure 1

2　早水先生の経歴・研究室紹介

■医学生・研修医のころ　さっき穂坂先生がざっと紹介してくれたんですけれども，私は別に進化生物学者ではないし，「なんでこの話してるの？」ってちょっと疑問に思うかもしれないんですよね．そもそも，どこから突っ込んでいいかわかんないような珍しい経歴なので，「医学部行ってお医者さんになったのになんで数学やってるの」みたいな興味が多分あるんじゃないかな，って思うんです．そうなんですよ，人は結構 10 年間で変わるなって思うんです！ 10 年あるとお医者さんから数学者になれるみたいな，そういう例を説明します (笑)．10 年前と生活が全然違うんですよ．だから自分でも驚くべきことというか，そういうことあるんだみたいな感じですね．多分意外とよくあるのかもって思ってるんですけど．

ちなみに高校は行ってないんです．今でいう高卒認定，当時は大検ってやつで自分で勉強して大学に入りました．高校の教育を全然受けてないんですね．だからそもそも高校で何やるか知らなかったので「それ違うよ」っていうのがあったら教えてください．

大学入ったのは研究者になりたいとかって思ったからなんです．私は大学入る時，元々ライフサイエンスの研究者になりたかったんですね．人間とかそういうものに関係する基礎研究をやりたいなと思って，大学に入ったんですよ．私は元々薬学の研究とかがいいなと思っていたので，お医者さんになろうとか微塵も考えたことがなくて，薬学部に行きたくて東大の理科二類に入ったんですよ．

皆さん知ってるかもしれないですけど，理科二類に入ると「進学振り分け」ってやつで成績次第でいろんな学部に行けるわけなんですよね．ただ 18 歳か 19 歳かそこらの時にですね，自分がどういう研究したいのかとか，何学部に行ったら自分がやりたいことができるのかとか，そういうの当然わかんないわけですよ．意識が高かった私は，いろんな研究室を巡っていろんな先生の話を聞いて「とりあえず医学部に進学をしよう」と思ったんです．数学と違って，そもそも生物系の研究とかっていうのはめちゃくちゃお金がかかって，すごく予算が潤沢にないとできる研究が限られたりもする．あと人間を使う研究はやっぱり医学部じゃないとできないとか．いろいろあって「行けるんだったら医学部に行った方がいいのかな」と思って医学部に進んだんですよ．

確か大学 2 年の終わりぐらいの頃に，とりあえずいくつかの研究室を見て回って勉強させてもらおうかなと思って，東大医学部の宮下研究室にメールをしてみたんです．脳の研究したいなと思って，脳の fMRI (functional MRI) っていう研究をやらせてもらったんです．脳の fMRI というのは，被験者を MRI に突っ込んで何かタスクをやらして「運動してる」とか「考え事してる」とか，そういう時に脳のどの部分が活動してるのかっていう画像を見る研究なんですよ．活動していそうな場所が，ピカピカ黄色く光るんです．

ただですね，「画像を見る」って言っても当然，脳細胞自身が光を発してるわけじゃないんです．あくまでも「脳のこの領域が活動していそうです」って計算された値，存在しないものを見てる画像っていう感じなので，「本当にそうなのかな？」って思う．写真みたいに目で見えて確かめられた画像じゃないから，当然その推定に使う統計の手法とか数理的な手法が変われば全然違う結果になってしまうし，解釈も変わりうる．統計手法とか数理的手法がなんか非常に重要なんだなってことは，データ解析を通じて当時一貫して自分の中にあったんですよね．

今でこそ多分皆さん機械学習とかディープラーニングとか流行ってるんで知ってると思うんですけど，当時全く流行ってなくて誰も知らなかった．けど，私は当時は全然その流行っていなかったこの分野が大事になるんだなっていう風に思ったんです．まず独学で機械学習とか統計のことを勉強したりとか，東大医学部のカリキュラムの一環で脳画像の統計解析のソフトウェア SPM 開発元 (University College London) に数ヶ月間留

学させてもらったりとか，そういうことをしてきました．その頃になると結構，将来は「統計とか数理とかそっち系で，理論研究で医学に貢献する道っていいな」ぐらいのことを思い始めていたんですよね．ただそうは言っても結構，医学部を卒業すると初期研修をやらなきゃいけなかったりとか，結局何か義務付けられてるので，なんかなんだかんだ言ってしばらくお医者さんをやっていました．画像をずっと扱ってたので，一応大学病院の放射線科医をしばらくやってたんですね．

■数学者への転向　放射線科で働くのも楽しかったんですけど，ある程度，仕事に慣れてくると，フリーランスの医者ができるぐらいにはなってました．そこで「ずっと研究者になりたいって思ってたし，そろそろ」って思って大学病院を退職しました．結局総研大って大学院に入って，統計数理研究所というところの機械学習のラボ，福水健次先生の研究室にお世話になりました．そのときにたまたま京大の CiRA（iPS 細胞研究所）にいた，東大医学部の先輩にあたる細胞生物学の研究者から，さっき言った幹細胞の分化の話，iPS 細胞の枝分かれの話を知りたいというデータ解析の相談を受けました．それを「へえー面白そう！」って思ったので，博士課程のテーマにしようかと思ったんですね．そもそも研究室の先生が機械学習の専門の先生だったし，「細胞の分化に関するデータ解析をできればいいな」ぐらいのことを思ってやり始めてたんですよ．

当時は離散数学とか，数学やりたいとかいう感じ全然じゃなかったんですが，iPS 細胞研究所のその研究者から「我々の業界で最近こういう論文が出てる」とか「ツリーに関するアルゴリズムが使われてるんだけど知ってる？」みたいな話を聞いて，なんか「へー離散数学って組合せの分野があるんだ」みたいなことを知りました．何も知らなかったけど，それで色々手当たり次第勉強したんですよ．周りに専門家がいなかったから手当たり次第読んだんですね．そしたら Mike Steel って人が書いた “Phylogenetics” っていう，組合せ論的系統学の超ド定番の教科書がすごいいい本[*8]で，私にとって，非常にアイオープナーだったんですよ．たまたまそれを開いたらですね，さっき冒頭で述べたような「グラフと生物学ってこんな関係あるんだ」みたいなね，進化の系統樹もそうだしネットワークもそうだし，食物連鎖のネットワークとか家系図とか，「こういうのって全部グラフだったんだ！」って思いました．「私が医学とかの教科書で見た，なんとかパスとかなんとかチェーンとかなんとかサイクルとか，そういうのって全部グラフの言葉だったんだ！」みたいな感じになってですね．それで，初めて見ることばっかりなのに，すごくファミリアというか見慣れたような気持ちになって，「この分野は面白いな！」とか「組合せ論的系統学っていうのか！」ってのめり込んだんですね．離散数学に一気にのめり込んで，すごい勢い余って実験研究者も巻き込んで一緒に離散数学の論文とか書いて，グラフ理論をやってもすごく面白いなっていう風に思い始めていたんですよ．

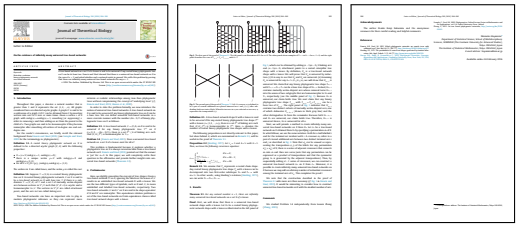
言い出すときりがないんですけど，SIAM (Societies for Industrial and Applied Mathematics) の「代数幾何の応用に関する国際会議」っていう，もはやタイトルとイメージが乖離してる感じの国際会議があるんですけど，そこの組合せ論的系統学のセッションでたまたまその時書いた論文に関して研究成果を話す機会があったんですよ[*9]．私としては，ただのフリーランスのお医者さんが趣味で離散数学やってるぐらいのノリなのに，代数幾何学の国際会議によく分からず連れてこられて，すごいアウェイな感じで，数学の研究発表なんて見たこともないけどこういう感じかなと思いながら喋ってたんですよね．でも「私のこと知ってる人なんかいないだろうしな」って思ってたら，たまたま最前列で聞いてる Andrew Francis って人がいて，その人が「あなたの最近出した論文見たよ」って声かけてくれたんですよ．「こんなアウェイなところで私のこと知ってる人がいるなんて嬉しい」って思って話を聞いたんですね．そしたらその人が，例の教科書を書いた Mike Steel っていう業界のオピニオンリーダーみたいな人の共同研究者で，一緒に論文を書いてる人だよって聞き

[*8] C. Semple and M. Steel, “Phylogenetics,” Oxford University Press, 2003

[*9] mini-symposium Combinatorial Phylogenetics 1, SIAM Conference on Applied Algebraic Geometry (AG’15) .

ました．それで，その Andrew Francis から「自分は系統ネットワークっていう進化のグラフのことを研究してるんだよね」「こういう未解決問題が色々あるんだよ」みたいなことを聞いて，「へー面白そう」みたいな会話をしたんですよ．「日本に帰ったらちょっと考えてみようかな」ぐらいの返事をして日本に帰りました．

日本に帰ったらそれはそれでいつもの日常で，普通のクリニックでアルバイトしてて，患者さんが来たり来なかったりして，そんな感じでなんかアルバイトしながら時間のあるときに考えてたんです．そしたらですね，その系統ネットワークに関する未解決問題の 1 つっていうのが，アルバイトしてる間に解けたんですね．[会場爆笑] 意外と簡単だったんですよ！こういうことあるんだなって思って．



見ての通り，3 ページ目がほぼ白紙なので実質 2 ページですごい終わってるすごい簡単な論文です．高校生でもできる「数学的帰納法で示します」みたいな，本当に簡単な証明だったんですよ．しかも *Journal of Theoretical Biology* っていう理論生物学のめちゃくちゃいいジャーナルに出したら通ったんです．あまりにもあっけない，すごい短いって理由で結構ウケが良かったんです．ほら，短いとみんな読む気がするから，結構いろんな人が読んでくれたんですよ．それで「あの論文の人だ！」みたいな感じで結構評判が良くなって，私はその反応を見て「いける！」って思いました．味をしめて「私はこの分野でやっていく」みたいな感じになってたんですよ．これだけですごいうまくいくわけじゃなく，その後いろいろ紆余曲折があったんですけど，ちょっと人にバカにされても頑張ってやっていたら，事態は段々好転していきました．

■研究者として　結局博士号を取った後には，組合せ論的系統学の研究者の一人としても，だんだん論文が出せたり研究が増えていったりとかして，いろんな研究をしてきました．理論だけではなくて，さっきの細胞の分化に関して iPS 細胞研究所の人たちと一緒に研究したりとか，細胞の分化に関するデータ解析ソフトを作ったりとか，結構いろんなことをやってます．

それから統計数理研究所の助教もしばらくやってたんですけど，研究所なので，学生を育てて増やす，自分の仲間を増やすみたいなことがあんまりできない環境なんですね．だから「早水さんってなんかいつも変わったことしてるよね」みたいなポジションで常に居続ける，よく言えばユニーク，悪く言えばマイナーというのがずっと続いてて「私の研究分野がメジャーになる方法は何かないのかな」と思ったわけですよ．

考えることとしては，当然人を育てて増やさなきゃいけないなっていう風に思って．それが理由で私は「学生がいっぱいいる大学に行って研究室を持って，そこでいろんな学生を育てて教えれば，私の仲間がいっぱいできて，この組合せ論的系統学の研究者がどんどん増えるじゃん」と思ったんですね．学会でも私がたくさん発表してるだけじゃなくて，私の学生がいっぱい増えて「組合せ論的系統学って今流行ってんだなみたいな感じになるじゃん！」て思って，それをやろうと思ったんですよね．私立大学はだいたいそうかもしれないんですけど，見ての通り早稲田大学は学生の人数が多くてですね．学生が 17 人ぐらいうちの研究室にいて，もうファミリーみたいな，すごい人数になってるんです．数学科と応用数理学科っていう 2 つの学科の学生が共存してるような感じなんです．私としては新しい場所というか，数学もやってる人もいるしデータ解析やってる人もいるしみたいな，いろんな人が集まる研究室になればいいなと思っていたんですが，今気づくと結構いい感じに運営できていたりします．

国際的な共同研究拠点というか，いろんな人が集まるような場所を作りたいなと色々思っていたんですけど，思いの他早く叶ってきましたね．学生が結構たくさんいるので学生同士の共同研究も活発だったりとか．数学者も来たりとかバイオインフォマティクスの人が来たりとか，突然幾何学的群論の人たちがなんかセミナーやったりとか，突拍子もない，幅広い感じです．エンジニアの人たちも来たりとかベンチャーの人も来たりとか，自由奔放にやっています．あと東大の医学部の学生のインターン先にもなっていて，うちで研究やると東大医学部の単位になるんですよ．なかなかない場所になってたりしますね．

私は今日話すように，系統の解析に関するグラフとアルゴリズムっていう感じで，組合せ論的系統学の話とかですね，あとは進化の話以外に，さっき言ってた細胞の系統解析みたいな話してます．

学生の研究内容も結構多岐にわたっていて，たとえばさっきの iPS 細胞のとかですね．がん細胞のデータの解析を機械学習の深層学習を使ったりとかっていう感じでやってる学生もいれば，系統学のより応用というか，離散幾何学な手法を使って鳥類のデータ解析をする学生もいたりとか．そういう応用だけじゃなくって，完全に理論系のグラフ理論のことをやってる人もいたりとか，全然関係ない感じの解析的組合せ論って分野の人もいたりとか，自分で考えたパズルゲーム的な離散数学の人もいればっていう感じで，すごい多岐にわたる内容をしています．こういう面白い場所なので，機会があったらぜひ遊びに来てください．

そんなわけで私としてはですね，大分うまく，意図をしていた通りの研究室ができつつあるなというふうに思っているところなんです．今日の講義内容は，私の YouTube チャンネルにあります．もし今日話を聞いて「もう一回聞きたいな」っていうことがありましたら，YouTube で適当に検索してもらえれば出てきます．必死にノート取らなくても大丈夫かなと思います．

■参考文献　ちなみにテキスト，参考文献について．本ってなると，もう基本は英語専門書ばっかりになっちゃいますね．さっき言った私が影響を受けたのが，C. Semple & M. Steel の “Phylogenetics” って本ですね．他にも

- D. H. Huson *et. al.*, “Phylogenetic networks: concepts, algorithms and applications,” Cambridge University Press, 2010.
- M. Steel, “Phylogeny: discrete and random processes in evolution,” SIAM, 2016.

がありますが，英語の本です．日本語で書かれてる本には，三中信宏先生の『生物系統学』（東京大学出版会，1997）って本があります．組合せ論だけじゃなくいろんなトピックを扱っていて面白いと思うんですね．

あと手前味噌で恐縮ですけれども，今回の組合せ論的系統学の講義内容に関する解説記事としては，既に出版されているものに雑誌「数学セミナー」の 2020 年 1 月号・2 月号がございます．2 回だけですけど「進化の系統樹とデータ解析」という連載記事を書かせてもらいました．またサイエンス社の「総理科学」という雑誌で，今度の 9 月号に「情報科学と数理の特集号」というものが出るんですけれど，そこにも私が「系統ネットワークの数理とアルゴリズム」という記事を寄稿させてもらいました[*10]．こちらはごく数日前にようやく原稿書いて，さらに最新の情報を盛り込んでやってる感じになっています．

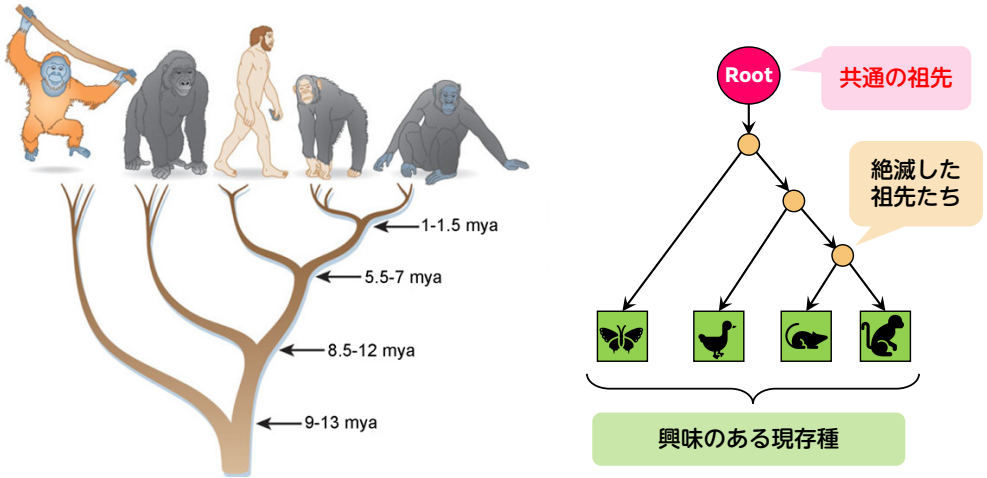
ここまでが私の長い長い自己紹介という感じでした．ここまで見るとだいたい，「そういう感じのいきさつの人が組合せ論的系統学のことを喋るんだな」ということが分かったと思うんですよね．というわけでだんだん内容に，系統学の話に入っていこうと思います．

[*10] このセミナーの後，2023 年 9 月になって，早水先生の記事が雑誌「数理科学」の 2023 年 9 月号に掲載されました．

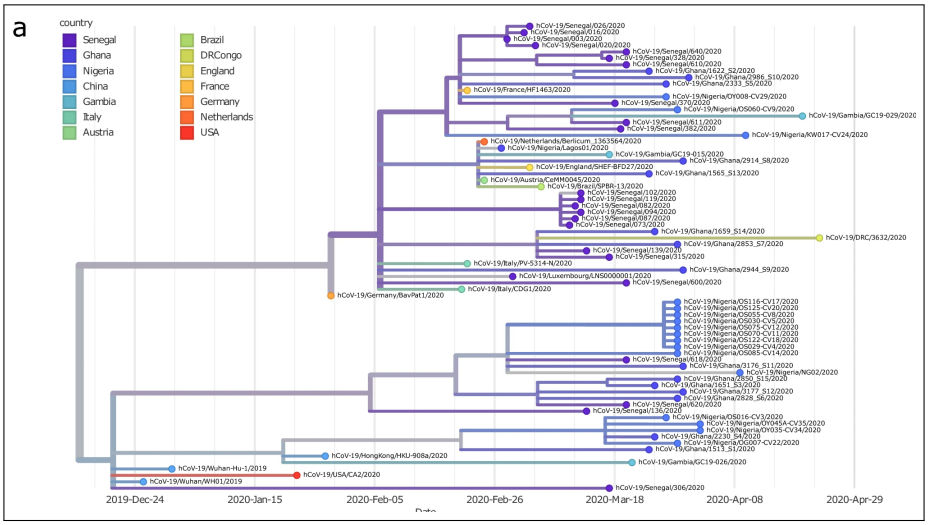
3　系統学の歴史と展望

3.1　20 世紀までの振り返り

■系統学　まず系統学について．進化の系統樹についてはさっき話した通りで，生物の進化を記述するための最も基本的なモデルですよね．共通の祖先がいて，そこから枝分かれして今に至ってるみたいなやつです．



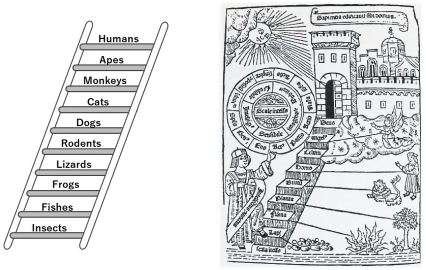
もちろん生物の進化だけじゃなくて，時事ネタでいうと新型コロナウイルスの系統解析とかですね．そういうのでもたくさん使われてるのは，結構目にするんじゃないかなと思います．系統樹はすごく広まってて，みんなが使っているものです．



新型コロナウイルスの系統解析[*11]

[*11] W. Wruck and J. Adjaye, “Detailed phylogenetic analysis tracks transmission of distinct SARS-COV-2 variants from China and Europe to West Africa,” *Scientific reports* **11**(1), 21108 の Figure 1a です．

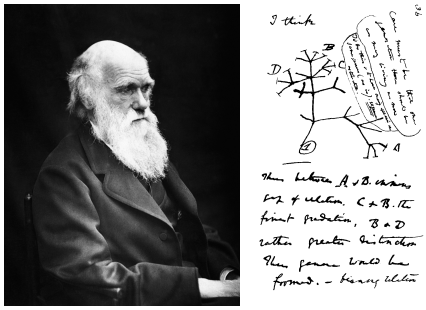
■進化論に至る道のり　ただよくよく調べてみると，進化を分岐図・樹形図で表すっていうこと自体は人類の歴史からすると非常に最近なんですね．というのは，そもそも「生物が進化する」っていう概念が登場したのって 19 世紀なんですよ．それまでは何だったかっていうと，よくあったのがこの階段の図なんです．



左: 古代ギリシャ時代の考え方 / 右: 13–14 世紀頃の考え方[*12]

19 世紀に至るまではですね，生物の序列というのを 1 本の直線とか 1 本の鎖とか，一つの階段とかはしごとかで表すのが常識だったんですね．“The Ladder of life” っていう風に言われるんですけど，Aristotle（アリストテレス）としてはただ単に「生命のはしご」みたいな．何かというと，生物は複雑なもの，高等なものから下等なものに必ず並べられるっていうのが左の図です．その右側にあるのは長年，科学と宗教が密接に関わっていた時期にずっと占めていた考えです．そもそも世界は神様が作ったっていうキリスト教の考えに基づくと，生物が進化するってことは神様が不完全なものを作ったという意味になるので，「それはありえない」というのが定説だったんですよ．だから序列の一番上には神がいて，生物に限定すると人間がいるとか．これが万物の序列を 1 本鎖で表したということです．無生物から神様までが 1 個の階段で表されてたんですね．今からすると笑ってしまうけど，サルはヒトになる途中であって，ほっといたらヒトになるっていう風に思われてた．もっと言うと無生物まで入ってるから，土もちょっとしたら植物になるし，植物もちょっとしたら動物になるし，動物もちょっとしたらヒトになるみたいな，そういう序列があったんですよね．

その定説を覆すターニングポイントになったのが，Darwin の進化論なんですよ．Darwin は「序列が 1 個に定まるんじゃなくて一つの種，起源から分岐したことによって，現在の多様な生き物が現れた」っていう革命的な物の考え方を提唱したんですよ．次の絵は 1837 年の有名な Darwin のスケッチですね．左に “I think” 「そうなんじゃないかと思う」と Darwin の筆跡で書いてあります．



これって結構不思議じゃないですか．Aristotle の時代から 19 世紀までのとんでもなく長い年月の間人類が支持し続けていた仮説から，なんでいきなり脱却して，直線じゃなくて分岐なんだって思い立ったのかなって．不思議だと思うんです．

[*12] Ramon Llull’s Ladder of Ascent and Descent of the Mind, 1305

なんでこれを思ったのか，書いた時どこにいたんだっていう話を調べてみます．Darwin はイギリス，バーミンガムの生物学者・博物学者で，この時船に乗っていて，すごくあちこちでフィールドワークしていろんな生物を探し続けて，調査していたんですね．イギリスから出発して 4 年間，ずっと船に乗って旅をしてたんですよ．イギリスから出て，見ての通り南米をぐるっと回ってあちこちのポイントを回ってました．その訪れる先の中にガラスパゴス諸島があったわけですね．ガラパゴス諸島って皆さん聞いたことあると思うんですけど，こういうような場所です．

左: Darwin の航海経路 / 右: ガラパゴス諸島の風景

ガラパゴス諸島の何が特別かというと，今まで大陸と一度も陸続きになったことがない，正真正銘の孤島なんですよ．他と何もインタラクションもないから，ここにしかいない生物ってのがうじゃうじゃいる．写真のようにその辺歩いてるカメが 2m あるとか，赤道直下の砂漠で生きるペンギンとか，「ペンギンってこんな進化し得たんだ」みたいなそういう例がいろいろゴロゴロ転がっている場所なんですね．Darwin はもちろん，ガラパゴス諸島でいろんな生き物を見たと思います．

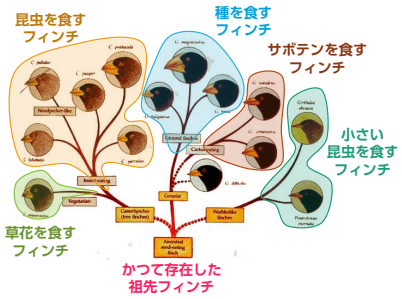
中でも何が Darwin を刺激したのかというと，フィンチっていう非常に有名な鳥です．

左から順にオオガラパゴスフィンチ，ガラパゴスフィンチ，
コダーウィンフィンチ，グレイムシクイフィンチ[*13].

フィンチっていうのは非常にいろんな種類がいて，姿形，特にくちばしの形にかなりバリエーションがあるんですよ．これ全部で 4 枚の写真出してますけど，左の 2 種類のフィンチたちは基本的に地面の方に住んでます．食べ物としては，落ちてきた木の実とかを壊してバリバリ噛んで食べてるようなやつらなんで，当然くちばしもガリって噛めるようなすごく太くでかい，たくましいくちばしを持ってるわけですね．一方で右の 2 種類は木の上に止まっていて，穴に入ってる虫とか何かをほじくり出したりつまんだりとかしてるんで，すごくちっちゃくて細いくちばしを持ってます．今は大まかに 2 系統だけ挙げましたけど，本当はもっと色んなのがいるんです．野菜だけ食べるベジタリアンとかいるんですよ．草だけ食べるフィンチとか，サボテン食べるのに特化したフィンチとか，すごい色々いるんです．

[*13] この図は Wikimedia Commons contributors, “File:Darwin’s finches.png,” Wikimedia Commons, （2023/11/20 閲覧）に並んでいる 4 枚の写真の配列を並べ替えたものです．

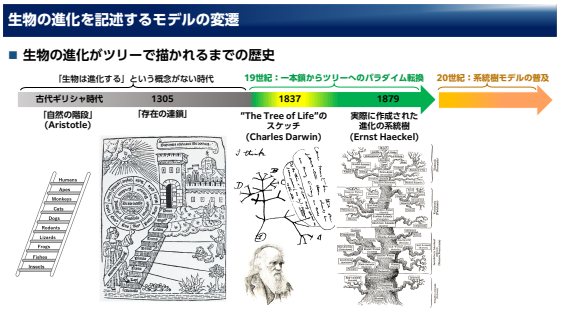
そして住んでる場所，暮らしぶり，それに最適化されたような全然違う姿形をしてるって事に Darwin は気づいて「なんでこんないろんな鳥がいるんだろう？」っていう風に考えたんですよね．最初っからいたのかなっていうふうに思うんですけど，なんかそれはおかしいと．じゃあ「外からやってきたのか？」って，そんなことはないわけですよ．孤島だから外からも来ない．ということは結局，元々そういう風に作られてたわけじゃなくて，最初は 1 種類だったんだけど，ランダムにいろんな突然変異ができていろんなのが生まれた．適者生存，その場所に特化した一番フィットしたやつだけがたまたま生き残った，今我々はその結果を見てるんだって気がついたわけなんですね．



フィンチの進化[*14]

そんなわけで Darwin は “The Tree of Life” っていうのを提案したんです．Darwin はガラパゴス諸島のフィンチの進化の道筋がツリーで表せるんじゃないかと思ったんだけど，“The” って付いてるとおりですね，それはガラパゴス諸島だけじゃなくて地球全体でも言えるんじゃないかなとも思ったんですね．地球って孤立した星の中で多様な生き物がいて，それは外から来たもんじゃないし．元々生命の起源っていうのは 1 種類の種がいて，そこから枝分かれしていろんな種が生まれたんじゃないかなっていう風に考えたわけですよ．だから神様が最初から作って今の種があるんじゃなくて，こういう風に現在までの全生物の進化の歴史はツリーで表せるっていう風に考えたわけですね．

これ神への冒涜みたいに受け取られて袋叩きにされるんですけど，幸いなことに Haeckel っていう Darwin の熱狂的な支持者みたいな生物学者が Darwin の仮説を裏付ける実際の生物の系統樹を作って，Darwin の考え，アイデアを具現化して，広めることに尽力したんです．そのおかげで幸い，Darwin が生きているうちに生物が進化するというのはその辺の人も知ってるような事実になりました．進化の系統鎖モデルが 19 世紀の終わりにはようやく消えたわけですね．それまでは階段の絵が教科書に載ってたんですよ．それが消えたのが 19 世紀の終わりなんです．

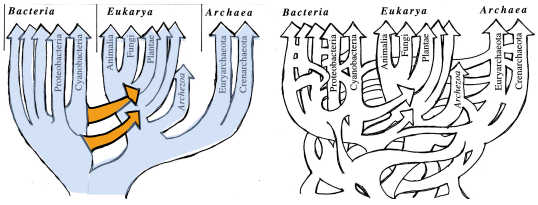


[*14] この図は https://www.pinterest.co.uk/pin/598838081681881492/ にある Sonali Sharma, “Identifying Darwin’s finches - Galapagos Conservation Trust” を早水先生が改変したものです．

3.2　21 世紀の展望

20 世紀に入るとようやく系統鎖の時代，一本鎖モデルが終わって系統樹モデルの時代になりました．だから今ある系統樹を推定する手法とかツール，そういうものは全部 20 世紀の産物なわけですね．

■系統ネットワークの登場　その 20 世紀には「系統樹いいね」って言われる一方で，「Darwin の考えは全部が正しいわけじゃないな」っていう話も出てきました．全ての生き物の進化を一つのツリーで表すの無理があるよねっていう話になった．さっきの交雑とか，水平伝播であるとかですね．「いろいろ考えると地球上に起きたこと全てを枝分かれの図で表すのは無理があるよ」ってことで，20 世紀の代わりにはこういう図がよく知られるようになったんですね．

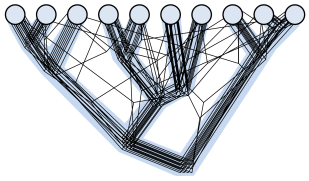


左: “Current Consensus” [Doolittle1999, Fig. 2][*15]
右: 網状進化ネットワーク [Doolittle1999, Fig. 3]

これらの図は，1999 年の *Science* 誌に乗ったものです．まあでも右の「網状進化ネットワーク」は結構過激派の意見で，左側にある “Current consensus” と言われてるやつが「大体こうなってるんじゃないかな」と多くの人が思ってる図らしいですね．地球上の全部の生命たちの進化を表すと「大まかに言えばツリーなんだけど，ちょっと無視できない現象もあるよね」という図です．

一方でこれだけ見ると「なんだじゃあツリーってもういらないんだ，もう忘れていいんだ」と思うかもしれないんですけど，全然そんなことはないです．地球上の全生物を見てるとツリーでは不十分だけど，例えば魚とヒトの進化とかを考えるときに，別に合流とか考える必要はないわけですよ．魚とヒトの間に今更子供とかできないので，その時はツリーで十分なわけですね．もちろんツリーで済む場合はツリーを使う方がいい．

でもツリーで十分な場合においても，いろんな情報源から集めた遺伝子系統樹，“Statistical Tree of Life” という話があります．遺伝子の配列から系統樹を生成するっていうにしてもですね，どこの遺伝子を使うかで推定される系統樹って違うわけなんですよ．だから遺伝子の数の分だけ，使う配列の種類の分だけ結果が変わってしまうので，「じゃあ結局どれを信じればいいのか」って話になるんです．そこで Koonin という人が「それらを全部集めた合いのこで，一番重なってるとこ，一番最もらしいところを推定する」っていうアイディアを提唱したんですね．こんな風に，元々の目標が系統樹だったとしても，その途中過程で系統樹を合わせるために系統ネットワークを使うというような，そういう使い方も考えられる．



統計的な生命の系統樹 [Puigbò–Wolf–Koonin2013, Figure 1]

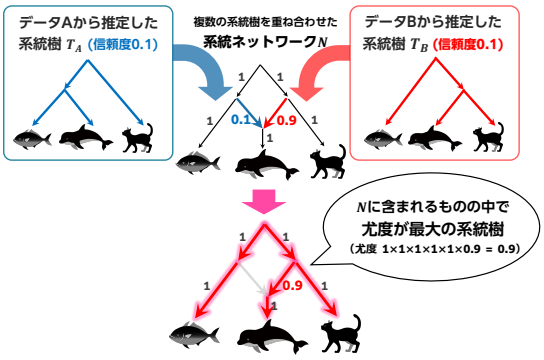
[*15] 原論文の図はモノクロです．着色は早水先生によります．

何にせよ「ツリーだけじゃ無理な場合があるしネットワークをうまく使わないと高度な系統解析はできないな」っていう話になっていったのが，20 世紀です．

■現状と問題意識　でも「今イメージされてたことが 21 世紀に実現できてるか」っていうと，部分的にしかできてないんですよ．すごく難しくて「できたらいいね」で止まっちゃってることが多いんですね．何故かと端的にいうと，アルゴリズムの設計がすごく難しいというのが理由です．系統樹はすごく単純なグラフなので，いろんな計算問題が簡単に解ける．全部が全部じゃないけど解けることが多い．数学的にも考えやすいしアルゴリズムの設計も易しいことが多いし，もちろん実装も簡単だったりとか，そういう意味で扱いやすいんですね．一方で系統ネットワークっていうのは分岐も合流もあり，もう本当何でもありってすごく広い対象なので，これを相手にすると急に物事は難しくなってしまったりするわけですよ．今日はあんまり話さないですけど，系統樹で普通に解けていた問題が系統ネットワークに置き換えると NP 困難になってしまうとか，多項式時間アルゴリズムがなさそうだみたいな話になっちゃったりして，いろいろな計算上の困難があったりするわけですね．仮に計算できたとしても，それの解釈も難しかったりする．どんな問題だったら効率的に解けるのか，どうやって計算するのか，それをどう活用するのか，そういうのが全然見通せていないのが現状だったりします．

というわけで今日，いろいろある系統ネットワークに関する問題の中でもとりわけフォーカスしたい問題は，さっきも出てきた「複雑な系統ネットワークが与えられています．その中に含まれるツリーを探したい」という問題です．これが問題意識です．

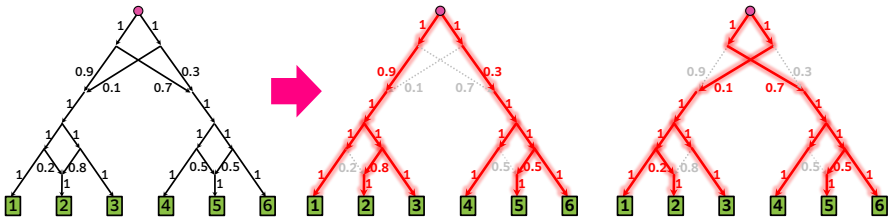
私がこれから話す内容に関してより具体的に言うと，Statistical Tree of Life が近いですね．さっきもちらっと言いましたけども，ゲノムの配列ってめちゃめちゃ長いので，全部使うわけにはいかないんですよ．だから「この配列の中のこの部分を使って生物の配列がどのぐらい違うかを比べて系統樹作りましょう」という風にやるんです．それで系統樹を作る時にどの遺伝子を使うかによって，推定される系統樹が変わってきてしまう．A という配列から推定したら左になる，B という配列から推定したら右になるっていう風に，どこを使うかによって結果が異なって進化のシナリオも変わってしまったりする．



この例だと 2 つしかないですけれども，例えば何らかの意味で片方の方が信頼度が非常に高い，A は信頼度 0.1，B は信頼度 0.9 で B の方が信頼できるとかだったら A を無視してもいいかもしれない．けど「データが何百個もあって，A を支持するデータもそこそこ何十個もあります」ってなると，無視しがたいわけですよね．そういうときは，どれか 1 個しか選ばないんじゃなくて一旦全部合わせてしまって，さっきみたいに複数のものの重ね合わせをして一番いいところを取るっていうアイデアが生きてくるわけですね．だからこんな風にして 2 つの系統樹を合わせる．

高校生の皆さんに尤度って使っちゃいましたけども，尤度は何かというと確率．たとえば確からしさが 0.1, 0.9 とあったとして，尤度ってのはそれらの掛け算だと思ってください．だから全ての辺が 1 だったら当然 1 でめちゃ確かだし，どっかすごい低い値の辺があったら「ちょっとありえないシナリオ」ってことになる．確からしさの度合いのことを，最もらしさの度合いのことを尤度っていう風に言うと．それだけ理解してくれればいいです．

今ツリーの候補が 2 通り，尤度 0.9 の系統樹か尤度 0.1 の系統樹だけになっちゃってますけれども，実際にはちょっとだけ複雑になるだけでも，あの選び方もあるし，これもあるしみたいな感じになってしまって，どれが最大の尤度なのかっていうのはよくわかんないわけですよ．結局「どれがいいんだろう」とかって話になるわけですよね．



というわけなので，後で詳しく説明しますけども「存在するならば一つシナリオを見つける」というのは典型的な問題です．他にも「候補の数が何通りあるのか」という数え上げ問題とか，「全部リストアップしなさい」という列挙問題だったりとか．あとは今みたいに尤度がついてたとしたら「尤度最大のものを見つけなさい」とか．尤度じゃなくてもよくて，何らかの意味でベストなやつ，例えば長さが最小のものを求めるとか，そういうのでもいいんです．あるいは「ナンバーワンだけじゃなくてトップ 5 を求める」とか，そういう自然な計算問題がいくつか出てくるわけですね．こういった問題は，系統ネットワークの中に含まれている系統樹を探るっていう話なんです．数え上げ問題よりも複雑な列挙問題，最適化問題やランキング問題については，多項式時間で解けるかどうか全然わかってなかったんですよ．

だからいろんな人たちが個々の問題を研究して，それぞれ「どう解けばいいのかな」とか考えてたんです．ですけど私の研究では，「全部 1 個，1 つの考え方でできるといいよね」っていうことをまず考えて，系統ネットワークの構造定理っていう基礎理論と言えるものを作りました．そこから全部の問題に対する，存在し得る中で一番早い線型時間アルゴリズムを導出するっていう研究をしました．そういうわけで今までの未解決だった「多項式時間で解けるのかな？」みたいな問いを解決しただけじゃなくて，今までは列挙とかその数えるとかだけだったんだけど，最尤推定っていう尤度を最大にする問題とか，最適化とかランキングっていう，誰も考えてなかったんで私が導入した問題，そういう統計的な応用も開拓することができたっていう研究です．その内容をお話ししようという話ですね．

ここまでで今日の概要は一応終わり．これから数学的な話をやっていこうと思います．

4　数学的準備

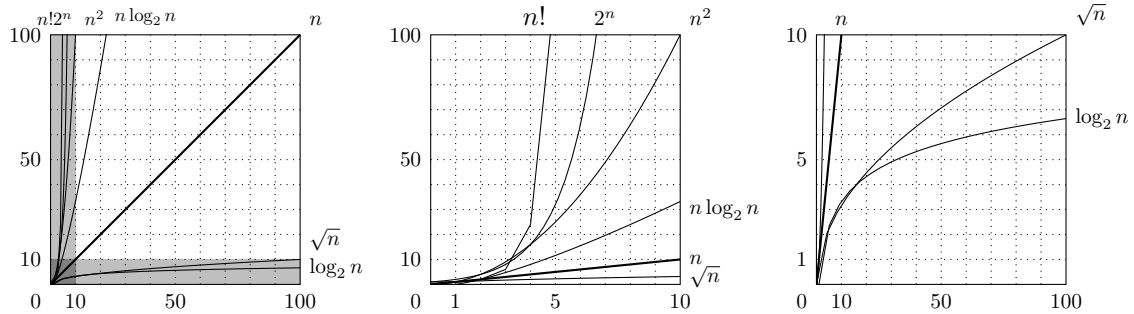
幸いにして高校生とかの生徒さんが多いので，さっき線型時間とか色々書いちゃいましたけど，「アルゴリズムが速いってなんだよ」っていう話に触れておきます．今日はそんなに長くないし難しいことやらないので深入りはしなくて，ごくごく必要な最小限までに留めます．

4.1　数学の一般論から

■アルゴリズムとその計算量　そもそもアルゴリズムというのは，解が定まっているような計算問題に対して，値を正しく求める手続き，やり方，解法を記したもののことをいいます．問題に対して「何が与えられて何を求めなさい」という風に解き方を記述したものなんです．

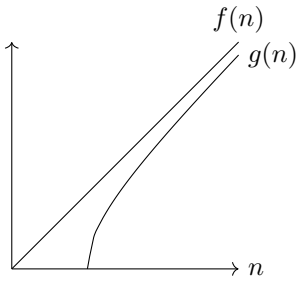
じゃあアルゴリズムの速い遅いをどう評価するかなんですけど，コンピューターで実装してストップウォッチとかで実測するのは良くないわけですね．もちろんそれでいいような状況もあるかもしれないですけど，使うコンピューターによって当然時間が変わっちゃう．それに例えば「n 個の数字が並んでいる状況で最大値を求めなさい」あるいは「n 個の数字の中に 5 っていう数字はあるでしょうか」って計算問題があるとするじゃないですか．そうすると n 個の数字が与えられるので，入力のサイズは n なわけですよ．n が小さければ別にすぐに求まる話ですよね．3 個，5 個とか 10 個とか数字があって最大値はどれですかみたいなのはすぐわかる．けど「n がめちゃくちゃ膨大だったときめちゃくちゃ時間かかるようじゃあ困る」，という話なんですよ．そうすると「ストップウォッチで測る」といっても「n がいくつの時に計測すればいいのか」って全然よくわからないので，理論的に解析されることが多いんですね．より正確に言うと「アルゴリズムを実行するのに要する最悪の場合の計算時間」を入力サイズ n の関数で表します．

皆さんなんとなくご存知なんじゃないかと思うんですけど，関数 $f(n)$ が階乗とか指数関数のオーダーになってると，n がちょっと増えただけですごくバーっと増えちゃって良くないわけですよね．我々が欲しいのは，例えば n が 2 倍，定数倍になるぐらいだったら実行するのにかかる時間も 2 倍くらい，2 倍ずつじゃなくていいけど定数倍で抑えられて済むような，そういうものだといいわけですよね．爆発しないといいなっていう話．さっきの 2^n とかだと爆発してしまうわけですけれども，n とか n^2 とか n^3 とか多項式の形をしてれば，n を定数倍したとき値も定数倍増えるだけなんで，今言った望み通りの性質を満たしてくれるわけですね．ということで多項式の形をしてると嬉しいなと．ざっくり言うとそういうことです．



計算量の議論でよく出る $\sqrt{n}$, n, $n\log_2 n$, n^2, 2^n, $n!$ の図．左の図におけるグレーの長方形を拡大したものが，右 2 つの図である．

■計算量のオーダー　多項式っていうと項が一つじゃなくて定数倍とか足し算とか出てくるなんて風に思うかもしれないんですけど，計算量を評価する時にはですね O（オーダー）という記法があって，定数倍は無視するとか，足し算があったら一番大きいところを拾うとかいう考え方をします．n が無限大に増えた時，要は n がすごい大きくなった時にアルゴリズムの実行時間がどうなるかっていうのを表すための手法なんです．



$g(n) = O(f(n))$

いろいろ書いてありますけど，要するにこの横軸の n がすごく増えてる時に，その実行時間の関数 $f(n)$ がどういう関数の下にあって上から押さえられるかっていうのを表しているわけです．結構どこにも書いてある話なので，興味のある人は後で調べてください．この定義に従うとですね，次のようになります．

例 1　適当な定数は 1 の定数倍で書けますので $O(1)$ です．例えば 100 は 1 の 100 倍なので $100 = O(1)$ っていう風に書けます．

例 2　$3n^2 - 100n + 6$ はですね，n が一定よりも大きい時には大体 n^2 乗の定数倍で書けることがすぐ確かめられると思います．なので $3n^2 - 100n + 6 = O(n^2)$ と書けます．3 っていう係数を無視していいし，2 乗よりも小さいところは捨てていいわけですね．

だだですねちょっと注意したいんですけど，今 $3n^2 - 100n + 6 = O(n^2)$ って言ったのはあくまでも上からの評価なんで，右辺を例えば $O(n^3)$ とか $O(n^4)$ と書いても別にいいんです．甘い評価をしてる分には別に構わないわけですよ．だからすごいガバガバ，ゆるゆるの評価で相当大きく見積っている可能性もあるわけですね．だから「これさえあればこの時間あれば十分だよ」ってこと言うオーダーっていう big O 記法[*16]とは逆に「この時間は必要だよ」，十分じゃなくて必要だよって下から抑えた評価をしたいってこともあるわけですよね．それが Ω っていうやつです．

O 記号とは逆に「n が十分大きい時は $f(n)$ はこの関数 $g(n)$ の上にありますよ」っていうのを $f(n) = \Omega\big(g(n)\big)$ と書きます．そして「O なんとか，かつ Ω なんとか」つまり上からも下からもこれで抑えられますよって，そういう時は Θ っていうのを使います．今日はあまり詳しいことは理解してなくてもいいんですけど，O と Θ がちょっと出てくるので，一応触れておきました．

次の性質は証明しようと思ったら使いますけど，今日はほとんど使わないです．

命題 1. $O\big(f(n)\big) + O\big(g(n)\big) = O\big(\max\{f(n), g(n)\}\big)$, $O\big(c \cdot f(n)\big) = O\big(f(n)\big)$.

[*16] big O 記法と似て非なる「small o 記法」というものも定義されるので，O のことをいちいち big O と言っています．big O と small o の関係は，大体「以下と未満の違い」のようなものです．

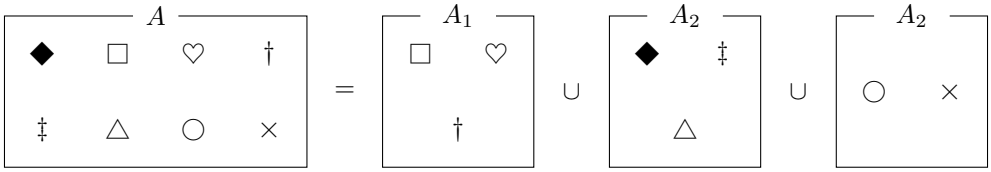
さっきの例 2 で見た通り，重要な性質として「足し算や定数倍の場合は簡単になるよ」ってことだけ覚えてれば十分です．積の形，掛け算の場合はちょっと物事が難しくなりますけど，足し算と定数倍だけ簡単ってことは覚えておいてください．

■集合・写像に関する用語と記法　あとは集合に関する用語と記法をちょっとさらっと話しておきます．和集合とかと積集合とか部分集合は知ってると思うんですけど，集合の分割だけ，一応説明しておきますね．

定義 2. A の部分集合族 $\mathcal{F} = \{A_1, \ldots, A_n\}$ が分割であるとは，以下の全ての条件を満たすこと．

① どの A_i も空集合でない
② $A_1, \ldots, A_n$ が互いに素（どの 2 つも共通部分を持たない）
③ $A = A_1 \cup \cdots \cup A_n$

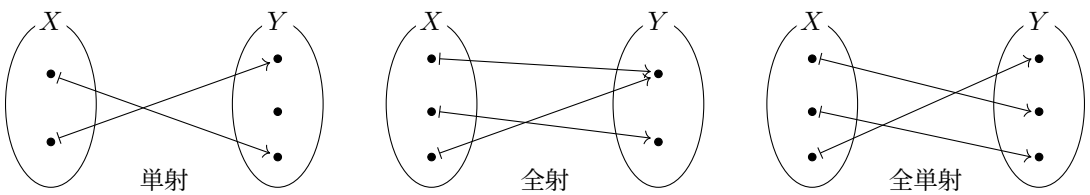
集合の要素を空でないような部分集合に区切ったもののことを分割といいます．これは後で出てくるので一応言っておきます．「空ではないこと」「重なってないこと」「合わせたら元の集合になってます」っていうところが重要なので強調しておきます．



写像に関しても，今日は全単射だけ知ってればよいです．2 つ集合があった時に，要素の間にもれなくダブりなく 1 対 1 な対応関係がある時に，そのままなんですけど一対一対応とか全単射とかいいます．

定義 3. 集合 X の任意の要素 x に集合 Y のちょうど 1 個の要素 $f(x) = y$ を対応させる対応 f のことを「X から Y への写像」という．

- 写像 $f\colon X \to Y$ が単射（全射）であるとは，集合 Y の任意の要素 y に対して $f(x) = y$ となる X の要素 x が高々 1 個（少なくとも 1 個）であることをいう．
- 全射かつ単射である写像を全単射または一対一対応という．



あとは直積というのが定義されます．子音と母音を例にとってみました．

	a	i	u	e	o
k	(k, a)	(k, i)	(k, u)	(k, e)	(k, o)
s	(s, a)	(s, i)	(s, u)	(s, e)	(s, o)
t	(t, a)	(t, i)	(t, u)	(t, e)	(t, o)

この表を見ればわかるからもういいかなっていうふうに思うんですけど，A が $\{\mathrm{k}, \mathrm{s}, \mathrm{t}\}$ っていう集合で B が母音の集合 $\{\mathrm{a}, \mathrm{i}, \mathrm{u}, \mathrm{e}, \mathrm{o}\}$ だとしたら，$A \times B$ という風に書くんですね．これ順番に意味があります．$B \times A$ じゃなくて $A \times B$ って書くと，先に子音が来て次に母音が来るような，この形をした要素の全部の集合のことを表します．

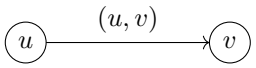
$A \times B \times C$ とか複数あっても，同じことです．こういうものを直積といいます．

4.2 グラフ理論の基礎知識

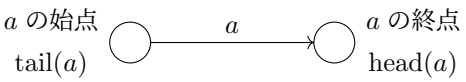
これで集合に関してさらっと説明したので，次にグラフに関する用語と記法の話をしようと思います．

■グラフに関する用語と記法 皆さん知らないと思うんですけれども，グラフには，無向グラフと有向グラフがあります．進化はちゃんと向きのある現象ですから，今日は無向グラフがたまにしか出てきません．有向グラフをメインに話します．矢印で遷移が表されるようなグラフだと思ってください．

向きのある辺のことアークといいます．書き方は人によりますけど，ここでは頂点 u から v に向かっているアークのことを，丸括弧で (u, v) という風に書きます[*17]．順番に意味がありますから，u から v に向かうのを (u, v) と書きます．端点 u, v が与えられてアークに名前を付けたいって状況とは逆に，始点とか終点の名前が u, v みたいな風に与えられず a というアークだけが与えられて始点と終点を名指ししたいっていうときは，a の終点のことを矢印の頭なんで $\mathrm{head}(a)$，a の始点側を $\mathrm{tail}(a)$ という風に書いたりします．



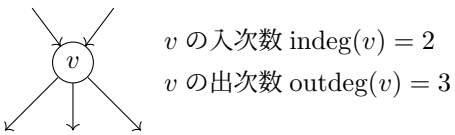
(i) 頂点に名前がある場合



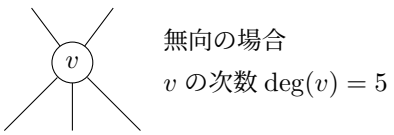
(ii) アークに名前がある場合

有向グラフ G の頂点集合を $V(G)$，アーク集合を $A(G)$ と表します．この辺グラフ理論の基本的なお約束事です．

■次数 グラフでは，頂点における次数というものが定義されます．有向グラフだと辺に向きがあるので，頂点の次数は 2 種類，入次数（いりじすう）と出次数（でじすう）というものを定義して欲しいんですね．訓読みで気持ち悪いんですけど「にゅうじすう」じゃなくて「いりじすう」って読むんですよ．



有向グラフの場合



無向グラフの場合

[*17] 今回の講義でターゲットにしている有向グラフでは，「頂点 u から v へ向かうアークが複数本ある」という状況が現れません．そのため始点と終点を指定した (u, v) という書き方で，きちんとアークが 1 本に定まります．

もし図のような頂点があったとき，v に 2 本辺が入って 3 本辺が出ていきますね．v の入次数 $\mathrm{indeg}(v)$ がこの場合に 2 ってことですね．出る方，出次数は $\mathrm{outdeg}(v) = 3$ という風に書きます．

あと「どこのグラフで測るか」っていうのが結構重要になることもあります．グラフ G の頂点の次数を見る，つまり「G の中で見た G の辺が何本入ってきてるか」を示すために，右下に添字 G をつけて indeg_G のように書くこともあります．省略することもしばしばあります．

これ無向グラフだったら向き関係ないので，ただ単に何本辺が接続されてるかっていう話なんで，この場合は単に $\deg(V) = 5$ という風に書きますね．これを先ほどと同じように「グラフ G で測ったときに次数 5 だよ」っていうには $\deg_G(V) = 5$ っていう風に書いたりします．これも省略することもあります．

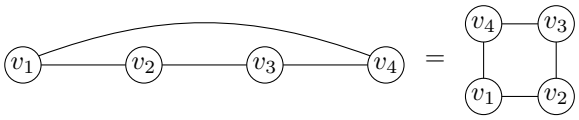
■パスとサイクル　今日はツリーの話をしていきますのでツリーを定義しないといけないんですけど，ツリーを定義するのに必要なのでしょうがなく，パス path も定義しますね．まず無向グラフの道 path の定義．

定義 4. 頂点集合が $V = \{v_1, \ldots, v_m\}$，辺集合が $E = \{\{v_1, v_2\}, \ldots, \{v_i, v_{i+1}\}, \ldots, \{v_{n-1}, v_n\}\}$ となっているグラフ V をパスという[*18].

見ての通り，絵で描けばなんてことはないです．頂点が v_1 とか v_2 とか v_3 とかあって，隣り合う点の間に辺があるっていう感じですね．要素は当然異なるもので，この列のパスの中に頂点とか辺の重複はないわけですね．これが無向パス P_4 です．

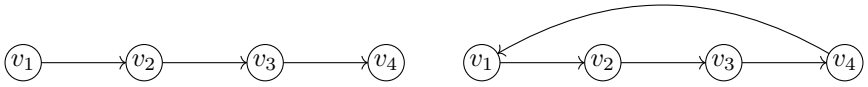
P_4: 

サイクルは，パスに 1 本余分なやつを付け足したやつですね．道 P_4 の最初の点と最後の点，始点と終点の間に辺を引っ張ったやつが C_4 です．

C_4: 

グラフは頂点集合と辺集合の順序対として定義されるので，描き方，見た目は別にどうでもいいんですよ．なので左は，あんまり輪っかに見えないですけど，ちょっと綺麗に整えていくと右のように，v_1, v_2, v_3, v_4 が正方形に並びます．これが C_4 っていうサイクルです．

有向の場合も同じで，これは向きありにすればいいだけですね．そうすると，辺に v_1 から v_2 のように向きがあるわけです．これを有向パスといいます．有向サイクルも全く同じで，有向パスに 1 本加えたやつのことですね．これが有向サイクルです．



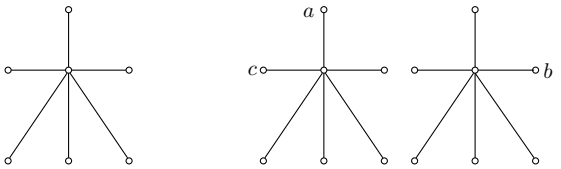
有向パス　　有向サイクル

[*18] 無向グラフの辺には向きがないので，「u と v を結ぶ辺」を「v と u を結ぶ辺」といっても構いません．そして，順序対の記号を使ってしまうと $u \neq v$ のとき $(u, v) \neq (v, u)$ となってしまいますが，集合の記法であれば $\{u, v\} = \{v, u\}$ が成り立ちます．そこで無向グラフで頂点 u, v を結ぶ辺を $\{u, v\}$ と表します．

■ツリー　というわけで有向なパスとサイクルができたところでですね，次は木と根つき木の話に行きます．

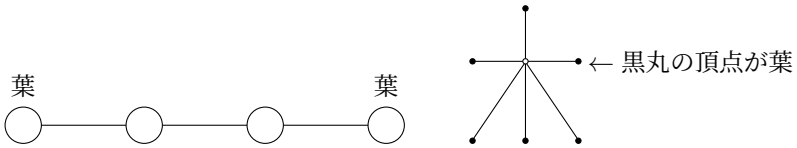
まず根無しのツリーを定義しますね．ツリーというのはサイクルを含まない連結な無向グラフ．連結っていうのはどの頂点の間にも道がある，パスがあるようなもののことをいいます．

いちいちいつも気に入って書いちゃうんですけど，漢字の「木」っていうのはグラフ理論の木なんですね．林は違うんですよ（笑）．ちなみに林もグラフ理論の用語なんです．森っていう用語もあります．だからグラフ理論の本を読んでると「林は森である」とか書かれちゃって，何いってるかよくわかんないんです（笑）．



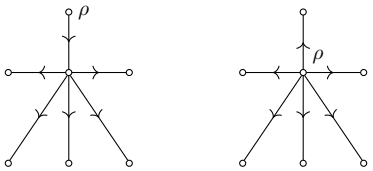
例えば図の「林」の頂点に a, b と書いてます．この a と b の間に道ないですよね．a と c の間は道があるけど，a と b は繋がってないですから，林は連結 connected じゃないわけですね．一方で木みたいにひと続きになってるものを，連結グラフといいます．P_4 も一本道ですごい単純ですけど，木なんですよ．木の例です．

次に，木において次数 1 の頂点のことを葉っぱ，leaf っていいます．P_4 の場合は左右の端は葉っぱですね．木の例だと次のようになります．



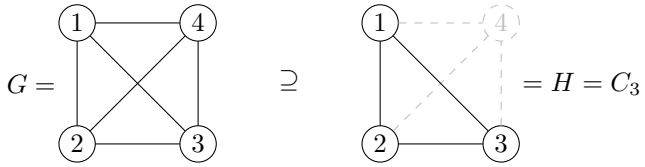
■根付き木　今日話すのは根無しの木じゃなくって，根付き木の話なんですよ．根付き木っていうのは木の各辺に向きがあって，根と呼ばれる唯一の特別な頂点 ρ 以外のどの任意の頂点 v に対しても，ρ から v の有向パスがあるもの．伝統的に，根はギリシャ文字で ρ って書かれる．アルファベット小文字の p じゃないですよ．

文章で書くと長ったらしくなりましたけれども，例えばさっきの「木」を根付き木に変換しましょう．ここから根付き木を作りましょうというと，根 root って呼ばれる特別な頂点 ρ をどこでどこ取ってもいいんですけど，例えば一番上が ρ だとしますよね．そうすると ρ 以外のどの頂点に対しても，ρ からその頂点への有向パスがあるってことなので，こういう風に根を 1 箇所定めると，他の辺の向きっていうのは自動的にその根から遠ざかる方向に向き付けられるわけですね．



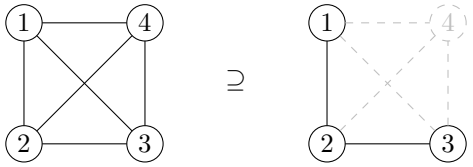
これだと簡単すぎるような感じに見えるかもしれないですけど，木の場合は「どの 2 頂点に対しても，必ずそれらをつなぐパスがただ一つ存在する」って性質があるので，ρ とその頂点 v をつなぐパスが 1 個しかないんですね．だから「ρ から v への有向パスになるように，ρ から遠ざかる方向に向きを付けることができる」という感じです．木の場合はね，根の位置さえ決めてしまえば，勝手に有向グラフになることになります．これが根付き木の定義ですね．

■部分グラフと部分木　何をしたかったかというのを思い出すと，系統ネットワークに含まれるツリーを考えていきたいって話をしたんです．すると「グラフが含まれる」って話をしないといけないわけですね．この左側に書いてある G っていうものにどんなグラフが含まれているか書きます．2 つのグラフ G と H に対して，頂点集合は $V(G)$ は $V(H)$ を含んでいる．さらに，向きがあってもなくてもいいんですけど，エッジやアークの集合にもこの包含関係がある．このとき H は G の部分グラフであるといい，集合じゃないけど，$H \subseteq G$ という記号で書きます．

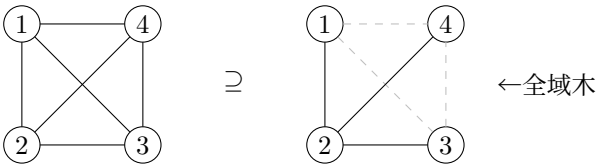


無向グラフの例になりましたけれども，図の左がグラフ G だとしますと，右の C_3 は G の部分グラフですよね．元々 $\{1,2,3,4\}$ っていう頂点集合があって，右の $\{1,2,3\}$ って頂点集合は含まれているし，辺も元々の辺に含まれていますので，右はグラフは G の部分グラフなわけです．

■全域木　さらに部分木と全域木の話をしたい．部分グラフの中で木であるようなもののことを部分木っていう風に言います．上の図の C_3 は部分グラフだけど木じゃないので，単なる G の部分グラフですよね．一方で下図の右辺は部分グラフだし，しかも木ですよね．なのでこれは部分木という風に言えます．

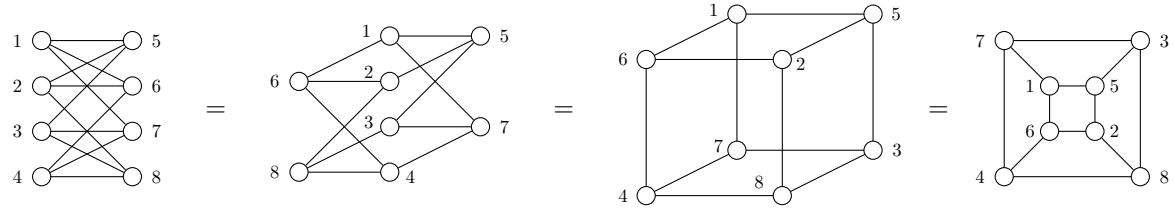


さらに，今日すごく大事なのが全域木というものです．頂点 1, 2, 3 からなる P_3 って，元々の G の全頂点を網羅してないわけですよ．全頂点を網羅してるような部分グラフ，$V = V(H) = V(G)$ のとき全域，G の全部に広がってるっていう意味で全域部分グラフと言います．しかもそれで部分木である場合は，全域木っていう風に言うわけです．上の C_3 は部分木なんだけど，もし頂点 2, 4 を結ぶ辺まであれば元々の G の全部の頂点を通るので，これがいてくれたら G の全域木になるんですね．それがこの話です．

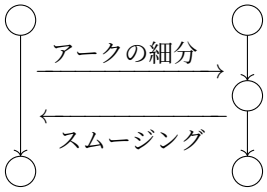


この全域木っていうのがすごい大事なので，押さえといてください．

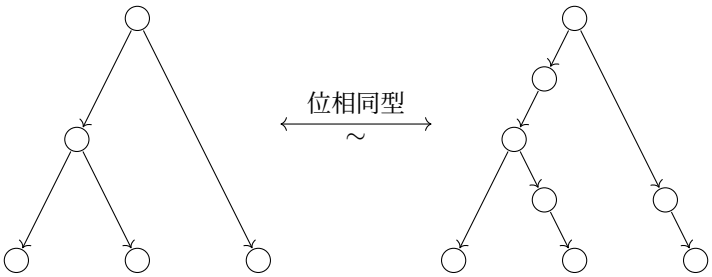
■同型と位相同型　あとはここから位相同型っていう話を定義しますが，その前に同型っていうのをさらっと言っておきます．さっき「グラフは描き方によらない」と言ったと思うんですよね．実際次の絵を見ても，全然見た目違いますけど，ちょっとレイアウトをうまく変えると同じグラフになると確かめられなると思うんです．この絵っていうのは，まさに同型なグラフの例になってるわけなんです．より数学的に厳密に言うと，頂点集合の間，および辺集合の間に 1 対 1 な関係があるときに同型なグラフだといいます．



これをちょっと緩めたものが位相同型っていう概念なんです．これを定義するのに辺の細分と頂点のスムージングっていうのを導入しておきます．細分っていうのは見ての通りで，細切れに分けることです．0 回以上分けることを細分っていう風に言います．分けなくてもいいんですよ．細分で出来上がったグラフそのもののことも，元のグラフの細分っていったりもします．逆に，細分すると，絵に出てるように「1 入って 1 出る」みたいな頂点ができますよね．これをなかったことにする，新しくできた 1 入って 1 出るのを無視する，細分の逆の操作っていうのを頂点のスムージングっていいます．

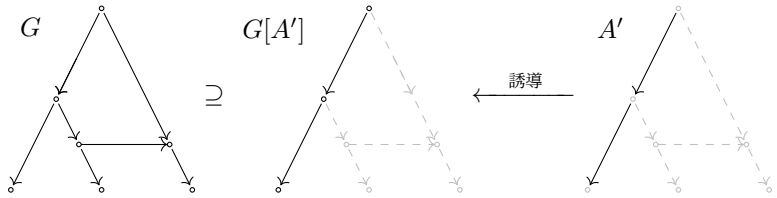


そしてスムージングできるものを全部スムージングして同型になるとき，位相同型といいます．スムージングの例だとちょっとあまりにも簡単すぎて実感もわかないと思うんですけど， 次の左のツリーと右のツリーは位相同型です．



これらは頂点の数が違うし辺の数も違うから全然同型じゃないですね．でも，頂点集合と辺集合にそれぞれ対応関係が全然ないんだけど形は一緒なわけですよ．こういう風に同型じゃないけど形は一緒，トポロジカルには一緒っていうところを位相同型といいます．

■アークが誘導するグラフ　あともう一個大事な概念があります．本当はグラフの分解って話をしたいんですけど，それを述べるために「アークが誘導する部分グラフ」の話をします．



先ほどから私がグラフを「頂点集合と辺集合のペアですよ」っていう風に言ってるんですけど，この図に出てる左のグラフが与えられてるとして，その一番右に書いてあるような部分グラフを指定したい時にですね，辺さえ指定すればいいんですよね．賢い皆さんであれば分かると思うんですが，両方の端点がないと線ができない．頂点は辺から自動的に定まるじゃないですか．「アークが含まれるような最小のグラフ」っていう風に作れる．真ん中のようにアークさえ知ってしまえば，一番右のグラフ定まりますよね．こういうのを「真ん中の選ばれた 2 本のアークのセットが右側のグラフを誘導する」って言うんですよ．

定義 5. グラフ G のアーク集合 $A(G)$ の部分集合 A' に対して，

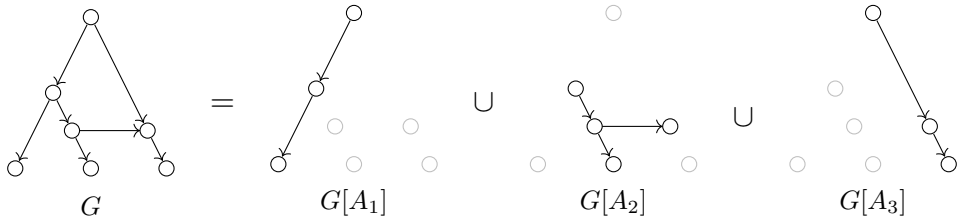
$$V(H) = \{A' \text{ に含まれる全アークの両端点の和 }\}, \qquad A(H) = A'$$

であるグラフ H を「A' が誘導する G の部分グラフ」といい，$G[A']$ で表す．

「右のグラフは真ん中のアーク集合によって誘導される部分グラフである」っていう風に言います，というのをわざわざ細かく数式で書いてあるのが，この定義ですね．角カッコで $G[A']$ と書きました．この記号は誰もが使うか知らないんですけど，よくそういう風に書いてある気がします．というわけで，これがアークが誘導する部分グラフの説明です．この記法を使って，次の分解っていう話をします．

■グラフの分解　集合に関する準備の時に，集合の分割っていうのが「空でない部分集合を重なりもなく漏れもないっていうものに分割すること」だとおさらいしたと思うんですけども，今度はこれを，与えられたグラフのアーク集合の分割で考えます．アークたちを色で塗り分けると思ってくれてもいいです．

たとえば次の図で，アーク集合 A_1 が誘導する部分グラフができますよね．左の $G[A_1]$ です．A_2 が誘導する $G[A_2]$ ってのができたり，A_3 が誘導する $G[A_3]$ ってのができてっていう風に，分割からそれぞれの誘導する部分グラフたちの集まりが得られます．すると，元々のグラフ G はこれらの $G[A_1]$, $G[A_2]$, $G[A_3]$ に分解されます．これがグラフの分解という概念です．要は，アーク集合の分割とほとんど同じということです．



ここまでが一応，グラフの基礎知識の話です．ここで質問，モヤモヤしたことあります？大丈夫？えっ，みんな大丈夫なんですか？？話が早い（笑）．

5　系統ネットワークの構造定理

いよいよ，進化の記述に用いるグラフの系統樹と系統ネットワークの定義をしていこうと思います．進化を記述するために使う有向グラフの中では，遡って進化のヒストリーを調べたいので，頂点集合と辺集合は有限であって欲しいですよね．また進化の話してて遡るから当然，辺に向きがあって欲しい．でも多重辺，なんかこうバーーーって同じ頂点から同じ頂点に向かうような辺がたくさんあったら困る．それに自己ループ，自分から出て自分に入るみたいなのもあったら困る，ということがまず一つあります．あと巡回，こういうぐるぐる回るサイクルも，進化の話してるんであったら困ります．

今回の話題から取り除かれるグラフ: 多重辺，自己ループ，サイクル

自己ループがなくて多重辺がないグラフを単純グラフといいます．また，ぐるぐる回るのがないのを非巡回グラフっていうんです．これから話すのは大枠で言うと有限で単純な非巡回の有向グラフの話になります．

■系統ネットワーク　系統ネットワークは系統樹を一般化したものなので，系統ネットワークの定義をしてから特殊ケースとして系統樹の定義を述べます．

系統ネットワークなんですけれども，正式名称が長いんですよ．さっきからさりげなく系統ネットワークと短く言ってしまってますけど，正式には「X 上の根付き二分系統ネットワーク」というちょっと長い名前です．これが何を言ってるのかっていうのをはっきりさせておきたいので，書こうと思います．

慣例的にですね，X を種の集合，たとえばヒト，サル，チンパンジーみたいなそういう n 個の有限個の現存種の集合だと思ってください．X は系統ネットワークとか系統樹の中で，リーフに「この葉っぱがヒトですよ」とか「この葉っぱがチンパンジーですよ」みたいな感じでラベルとして使ったりするので，ラベル集合って呼んだりします．

それで今から述べたい，系統ネットワークの正式名称が「X 上の根付き二分系統ネットワーク」というものなんです．これ非常にジェネラルな概念なんです．

定義 6. 以下の 3 つを満たす有向非巡回グラフを，X 上の根付き二分系統ネットワークという．

① X と葉集合 $L := \{v \in V \mid \mathrm{indeg}(\rho) = 1 \text{ かつ } \mathrm{outdeg}(\rho) = 0\}$ の間に 1 対 1 対応が存在する．
② $\exists! \rho \in V$ s.t. $\mathrm{indeg}(\rho) = 0$ かつ $\mathrm{outdeg}(\rho) \in \{1, 2\}$.
③ $\forall v \in V \setminus (X \cup \{\rho\})$, $\big(\mathrm{indeg}(\rho), \mathrm{outdeg}(\rho)\big) = (1, 2)$ or $(2, 1)$.

① まずは「X 上の」というところを説明します．X とそのリーフ集合，ツリーの中の行き止まりの集合のところに 1 対 1 の対応関係があるって言いたいんです．ちゃんと書こうと思うと，1 入って 0 出る，$\mathrm{indeg}(v)$ が 1 で $\mathrm{outdeg}(v)$ が 0 であるリーフの集合を L って書きます．これと X の間に $1:1$ 対応の関係がある，というのが 1 個目の条件です．これが「X 上の」って言ってる意味です．

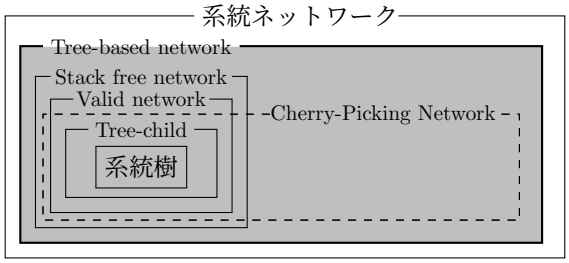
② さっき根付き木の定義で「特別な頂点である ρ ってやつがただ一つ存在する」と言ったんですね．この存在記号の横にあるビックリマークが，「ただ一つ」って意味です．ただ一つ ρ っていうやつが存在します．どんなやつかというと，V の要素のどこからも入ってこない，それから出るだけっていう始まりのやつで，ρ に入るものは何もないんです．$\mathrm{outdeg}(\rho)$ が 1 か 2 というのはどっちでもいい．大した話じゃないです．始まりが ρ から 1 本出るパターンか ρ から分かれるパターンかっていう，ただそれだけのことなんで，まあどっちでもいいです．1 または 2 ってことですね．

③ あとは根でも葉でもないその他の頂点に関して言うと，L が X と 1 : 1 対応してるので X と同一視します．葉っぱでも根っこでもないようなすべての頂点に対して，$\mathrm{indeg}(v)$, $\mathrm{outdeg}(v)$ の組が，1 入って 2 出るか，または 2 入って 1 出るかのどっちかである．

こういう①②③を満たすような有向非巡回グラフのことを，X 上の根付き二分系統ネットワークといいます．

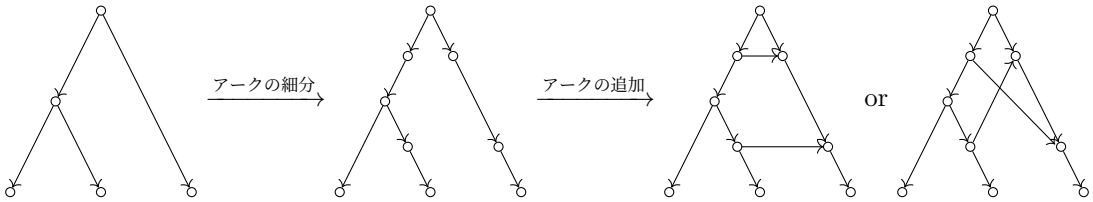
系統樹に関しては「もし 2 本の合流が存在せず分岐ばっかりならば系統樹」という定義です．

■tree-based network　ようやくこれで系統樹と系統ネットワークの定義ができました．系統樹っていうすごく狭い有向非巡回グラフが，合流を許すことによって，すごく広くなったわけ．ただクラスとしては広くなったんですけど，広すぎる．最初話しましたけど，いろんな問題が難しくなっている．だからちょうどいい塩梅のクラスを見つけたいっていう研究がたくさんあるんですよ．これがすごい乱立してて，「この問題解くのにこういうクラス考えるといい」とかいろんな人が考えてるんです．

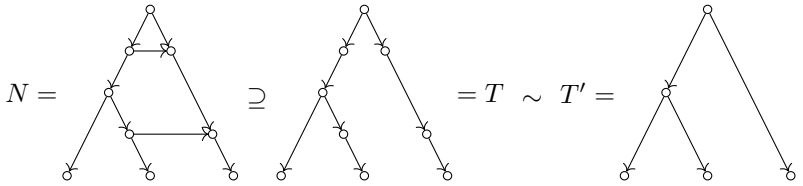


今から話す tree-based network というやつはですね，それらをひっくるめた結構かなり大きいクラスで，最近非常に intensive に研究がされています．この名前だと系統樹ベースの系統ネットワークっていう，名前が良くないかもしれないんですけど，そういうやつなんです．系統樹を含んでますよね．なんかの系統樹と位相同型なものを含んでいて，それプラス余分なアークがくっついてるって感じ．直感的に言うと，系統樹 $+\alpha$ みたいなものとして捉えることができるクラスのことなんです．

Tree-based network の作り方は単純，簡単な作り方で作ることができます．元になってる系統樹を用意して，そこからアークを細分するんですね．アークを細分すると，1 入って 1 出るっていう新しい頂点ができるわけなので，新しくできた頂点同士の間に端点を共有したいようなアークを付け足していきます．何で端点を共有したいかっていうと，そうじゃないと binary 条件，indeg と outdeg の一方が 1 で他方が 2 って制約が満たされなくなっちゃう．ただそれだけの理由なんです．そのプラスアルファのものを付け足していきます．もし使っていない頂点があったら後でスムージングします．そうやって作るようなものですね．ツリーをベースにして余分な辺を付け足すという操作で作れるのを tree-based network っていいます．作るのすごい簡単ですよね．もちろんこのアークの余分な加え方は 1 通りじゃなくて，端点を共有しなければ何でもいいので，追加した辺をクロスさせても tree-based network って感じで，いろんな作り方ができます．



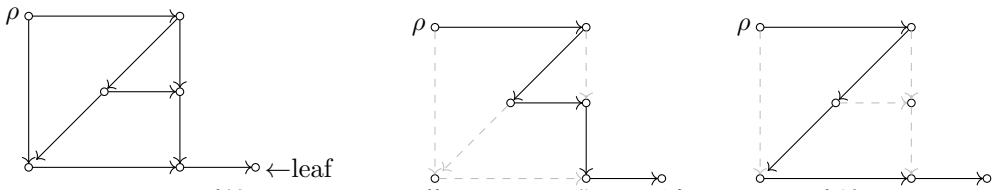
■全域系統樹　逆にですね，元々何から作られてるかを知らないとき元は何だったのかを調べる話をしたい．遡るのはちょっと難しいんですよ．Tree-based network の定義は直感的で分かりやすいんですけど，どっちかっていうと真ん中にある，ネットワークに含まれてるツリーの方が主役なので，そっちのを主役にした別の定義を紹介しますね．



真ん中にあるやつ T のことを，これから N の全域系統樹と呼ぶことにします．これは tree-based network 中に全域木として含まれているような系統樹．より正確に言うと，X 上の何らかの根付き二分系統樹と位相同型な全域木のことを言います．この場合だと，一番右側のやつがまさに X 上の何らかの系統樹で，真ん中のやつ T は T' と位相同型なんですね．スムージングしたら同型になります．そして余分な辺を無視すれば含まれているので，T は N の全域木であるわけですよ．こんなものを全域系統樹っていう風に言います．Tree-based network の定義と全然同値な言い換えで，その中でどういう風に言ってもいいんですけど，真ん中の N に含まれてる系統樹のことを全域系統樹っていう風に言います．

何で全域木にこだわってるのかっていうと，一般の部分系統樹だと数え上げが多項式時間じゃできないってことがもう証明されてるんです．だから全域木にフォーカスしてるわけなんですね．

Tree-based network なり系統ネットワークなりが与えられて遡るのは難しいって言いましたけども，何が難しいのかって話を一つします．さっき系統ネットワークは分岐も合流もありという風に定義したじゃないですか．今の作り方だと系統樹に合流を足して tree-based network 作ってるんで「さっきの定義と何が違うの？」っていう風に思うかもしれない，なんか違わないんじゃないと思うかもしれないですけど，面白いことに違うんですね．結構狭いんですよ．tree-based じゃない系統ネットワークのすごい単純な例があります．



tree-based でない系統ネットワーク．葉につながる唯一の頂点にアークが合流し，それら 2 本の始点を結ぶアークがないので，合流を避けて全域木を作ることができない．

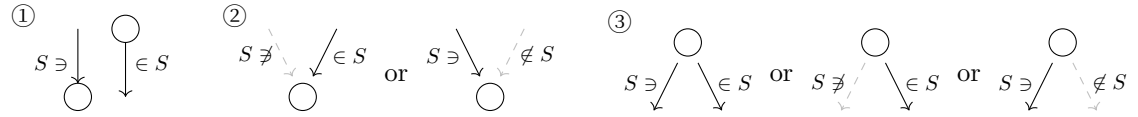
このグラフに関して言うと，その根と葉っぱが指定されてるので，指定の根から始まって指定の位置のリーフで終わるようなツリーを書こうとすると全域木が作れないんですね．だから全域系統樹が存在しない例なんです．どう頑張ってもないんですね．なので tree-based network っていうのは，系統ネットワークよりも狭いということがわかる例なんです．

■全域系統樹の存在判定　じゃあ当然，与えられたネットワークが tree-based かどうか，全域系統樹はそもそも存在するのかどうかっていうのを判定したいっていう計算問題が基本的なものとしてあるわけですよね．また，もし存在するにしてもそれをどうやって見つけたらいいのかっていう話になりますよね．そこで全域木を作る時に一つヒントになるのが，全域系統樹なんで，アークしか選ぶ余地ないわけですよ．全域木なんで頂点集合は元のネットワークと同じですから，どのアークを選べばいいのかっていう問題になる．適当に選んだら当然木にならなかったりとか，全域系統樹じゃない木になっちゃうので，「どういうルールでアークを選択すれば全域系統樹になるんですか」っていう話ですよね．

このクイズに対する答えはもう与えられています．最初にちょっと話した Francis と Steel という人たちが論文でですね，必要十分条件を与えてるんですよ．すごい簡単な必要十分条件で，必要条件としては明確なんですけど，それが十分条件でもあるってことを示しています．

定理 7 [Francis–Steel2015]. $S \subset A(N)$ が全域系統樹を誘導するための必要十分条件は，S が次の制約条件①〜③を満たすこと．

① $\mathrm{indeg}_N(v) = 1$ または $\mathrm{outdeg}_N(v) = 1$ ならば，S が (u, v) を含む．
② $\mathrm{head}(a_1) = \mathrm{head}(a_2)$ ならば，S は $\{a_1, a_2\}$ の一方だけを含む．
③ $\mathrm{tail}(a_1) = \mathrm{tail}(a_2)$ ならば，S は $\{a_1, a_2\}$ の少なくとも一方を含む．

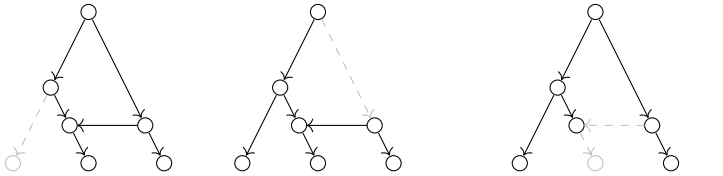


必要条件なのは納得できると思います．

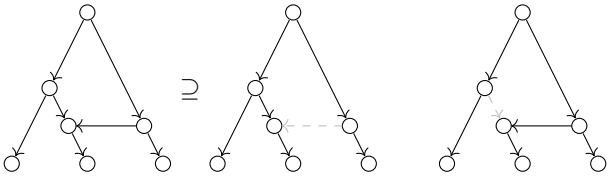
① まず頂点の入次数が 1 だったり出次数が 1 だったりする場合は，その頂点に入るアークとか出るアークを絶対採用しなくちゃいけない．もしそれを採用しなかったら，新しく根になっちゃったりリーフになっちゃったりする．行き止まりになっちゃったり変に途切れちゃったりするんで，その 1 入ってるアークとか 1 出てるアークは絶対選ばなきゃいけない．これが一つ目の条件です．
② 次に大事なのが②番，2 本合流してる場合ですね．これ 2 本とも選んじゃったら木にならないわけですし，かといって 0 本を選ぶっていう風にしたらそこが根，ルートになっちゃう．ということは左右のどっちか片方，ちょうど 1 本だけ選ばなきゃいけないということになるわけですね．
③ ③番は，ある頂点から 2 本出ている場合は「ちょうど 1 本」じゃなくて「少なくても 1 本選べばいい」って話になるので 3 通りあるわけなんです．

全域系統樹を作るのにこの①②③が必要というと，それはそうだろうって感じです．ところが実は，これらが必要十分条件だってことを彼らが示したんですね．つまり系統樹がどうのとか全域木がどうのとか系統ネットワークがどうとか何も考えなくてよくて，この①②③の条件を満たすように適当にアークを選択する，ただそれだけで勝手に全域木を誘導するアークセットになるという話なんですよ．彼らはこうやって①②③を満たすようなアークの集合の集まりと全域系統樹っていうのが対応してるってことを特徴付けたわけなんですね．

今言葉で説明しましたけど，条件①②③を満たさないような感じで適当に集められちゃうと，これ実際，望み通りのものとは違うものが出てきちゃったりとかして全域系統樹にならない．



①②③を満たすようにアークを選ぶとですね，こんな図のように簡単な場合だと，系統ネットワークの中には制約を満たすアーク集合の作り方が 2 通りあるなってことが分かるわけなんです．



簡単な場合だとすぐわかるんですけど，一般にネットワークがでかくなったらどうなるのかっていう話です．冒頭で述べた目指したい問題としてはですね，制約を満たすアークの取り方っていうのが存在するのかとか何通りあるのかとか列挙したりとか，その中でも重みがあったらどれがベストなものなのかとかランキング作りたいとか，そういう話なんです．それを解くのに構造定理ってのを証明したんです．

■構造定理の例　これは有限 Abel 群の構造定理っていうもの，大学に入ると人によってはやると思います．

定理 8. 任意の有限 Abel 群は，巡回群の直積で書ける．

全然内容を知らなくていいんですけど，雰囲気だけつかむため乱暴な言い方をすると，素因数分解みたいに複雑なものを一意的に分解できるような形，分解方法を与えています．これを応用すると所望の位数，サイズの有限 Abel 群っていうものの数え上げとか列挙とかに使うことができる．そういうものなんですね．物事を簡単なパズルのピースみたいなものに分解して，数え上げとか列挙とかできちゃう話です．

$\mathbb{Z}_{2^4} \times \mathbb{Z}_{3^2}$	$\mathbb{Z}_{2^4} \times \mathbb{Z}_3 \times \mathbb{Z}_3$
$\mathbb{Z}_{2^3} \times \mathbb{Z}_2 \times \mathbb{Z}_{3^2}$	$\mathbb{Z}_{2^3} \times \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_3$
$\mathbb{Z}_{2^2} \times \mathbb{Z}_{2^2} \times \mathbb{Z}_{3^2}$	$\mathbb{Z}_{2^2} \times \mathbb{Z}_{2^2} \times \mathbb{Z}_3 \times \mathbb{Z}_3$
$\mathbb{Z}_{2^2} \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_{3^2}$	$\mathbb{Z}_{2^2} \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_3$
$\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_{3^2}$	$\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_3$

位数 72 の Abel 群の構造．素因数分解 $72 = 2^4 \times 3^2$ に合わせ，
4 の分割（5 通り）と 2 の分割（2 通り）をかけた格好になる．

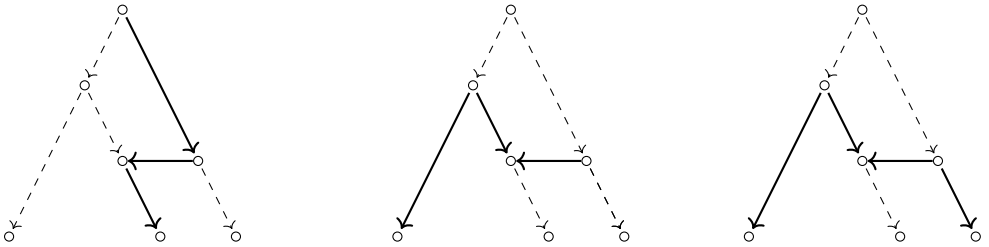
そういうものなんだみたいなイメージを持ってくれれば十分です．インスピレーションで雰囲気だけ伝わればいいんです．聞きたい人はあとで学校の先生に聞いてください（笑）．

■極大ジグザグ・トレイル　有限 Abel 群の話は置いといて全然大丈夫．今日これから述べる系統ネットワークの構造定理ではそういう知識いらなくて，覚えてほしいアイディア，キーワードは「極大ジグザグ・トレイル」って考え方だけなんです．

極大ジグザグ・トレイルっていうのは私が考えたものです．数学的にちゃんと定義するとちょっとややこしいんですけど，連結部分グラフのうち，隣り合うやつが head か tail を共有してるような部分グラフ，交互に下がって上がってみたいになってるやつをジグザグ・トレイルといいます．その中で，それ以上伸ばせなくないものを極大といいます．

定義 9. X 上の根付き二分系統ネットワーク N の極大ジグザグ・トレイルとは，$|A(Z)| \geqq 1$ なる N の連結部分グラフ Z のうち，全ての $1 \leqq i \leqq m-1$ で $\mathrm{head}(a_i) = \mathrm{head}(a_{i+1})$ または $\mathrm{tail}(a_i) = \mathrm{tail}(a_{i+1})$ が成立するような $A(Z)$ の置換 $(a_1, \ldots, a_m)$ が存在するもの．

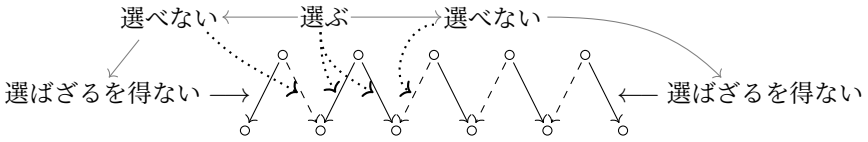
要するにこういうやつです．



ジグザグ・トレイルでないパス　　極大でないジグザグ・トレイル　　極大ジグザグ・トレイル

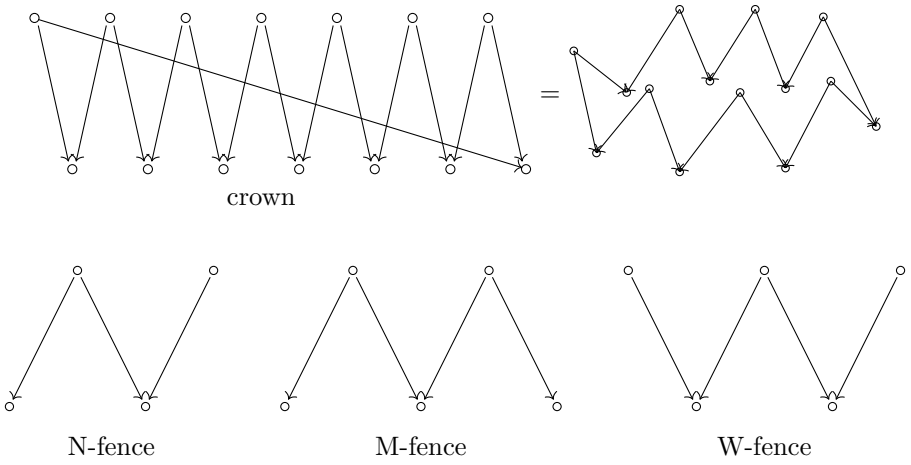
上の図だと，右 2 つは極大ジグザグ・トレイルだけど，一番左のは隣り合ってるやつが始点同士/終点同士を共有はしてないわけなので，違います．順方向に進んでいるだけ，ただの有向パスです．極大/極大じゃないでいうと，右は極大．真ん中はトレイルではあるんだけどまた伸ばせますから極大じゃない．ということで，伸ばしてきたやつのことを極大という風にいいます．

Francis–Steel の制約②③を思い出すと，V 型とか ∧ 型という状況に対して「片方しか選んじゃダメ」とか「少なくとも 1 個は選ばなきゃダメ」とかいう制約でした．なので，例えば選ぶ/選ばないを 1/0 でマークすると，2 本入る状況があったら，ある場所を 1 選ぶとマークしたら隣は 0 にならなきゃいけないわけですよね．逆に 0 でマークしたら反対は 1 にならなきゃいけない．2 本出る状況だったら，左側を 1 と決めても，選び方が (11) ってこともあれば (10) のこともある．そんな風に，片方の選択が別のところに影響を及ぼすわけですよ．これがずっと連なってたら，選択が波及して，よそのところまで影響を及ぼしたりするわけですね．だからここでジグザグ・トレイルを定義しようとしてる意味は，影響が及ぶような一番でかい範囲というのを定めようとしているんです．



頂点に入ってくる V 型，入るアークと出るアークみたいなところだけ選択の余地があるわけなんで，それで連なってるようなもののことを極大ジグザグ・トレイルって呼ぶことにしたんですよ．

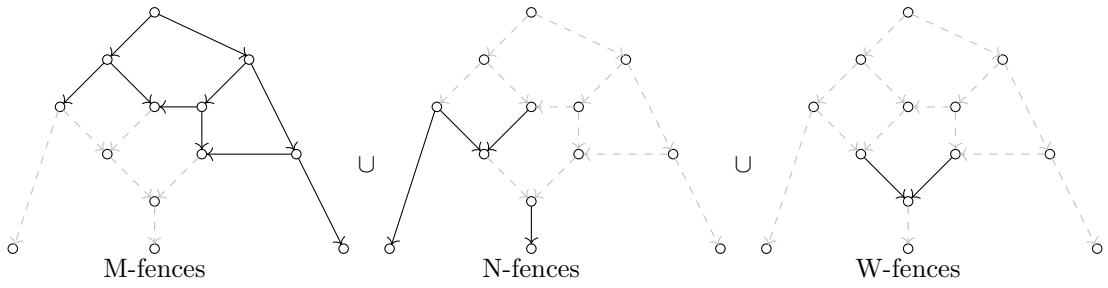
ジグザグ・トレイルの形の分類はすごい簡単です．この V 型とか A 型とか，ギザギザとかジグザグだけでできてるので，4 種類しかないんですね．1 個はクラウンっていう，王冠みたいな輪っかになるやつ．そうじゃない，輪っかにならないんだったら，アルファベットの M や N みたいなやつ，それから W みたいなやつがあります．



■主定理　一生懸命定義しましたけどなんでこれ定義したのかっていうと，任意の根付き二分系統ネットワークっていうのは，実は極大ジグザグ・トレイルに一意的に分解できるんですね．これがパズルのピースになってるんですよ．……良い反応で良かったです（笑）．これが嬉しいところなんですよ．

定理 10 [Hayamizu2021, Theorem 4.2]**.** 任意の根付き二分系統ネットワーク N は，N の極大ジグザグ・トレイル $Z_1, \ldots, Z_l$ に一意に分解される．

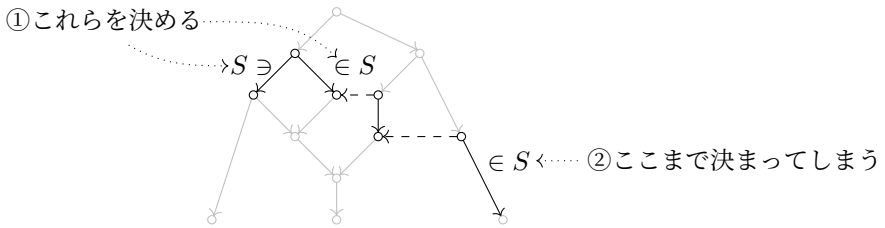
すごい簡単な話なんだけど，言われてみれば本当だって感じになってるんですよ．しかもただ単に 1 個アーク選んで伸ばせるところだけ伸ばせばいいんで，作り方もすごい簡単ですよね．どれか 1 つから始めて，隣り合ってるところをジグザグ辿っていって，「これ以上いけないわ」ってところで「じゃあこれは Z_1 にしましょう」とか言って，1 個 1 個なめていけばいいわけですよね．という感じで，こんな風に全部分解できます．



今すごい描画しやすい例だけ出しましたけど，どんなに複雑になっても絶対必ずこういうピースで出来上がってることが証明できるわけです．

これができると何が嬉しいかっていうと，制約条件を満たす/満たさないっていうやつが，その影響が及ぼす範囲だけで決まるんです．Francis–Steel の制約条件①，②，③の判定にはネットワーク全体を見る必要が全然なくて，分解し終わったそのごく単純なパズルのピースだけ見て決めればいいわけですよね．

だから次の図のジグザグ・トレイルだったら，「左端の 2 本を採用します」っていったら隣は選べなくなっちゃうし，そしたら隣は選ばなきゃいけなくてその隣は選べない……みたいな感じで，連鎖的につながっていくわけですよ．結構，大部分が指定されるわけですね．そんな感じで全部書き出すことができます．



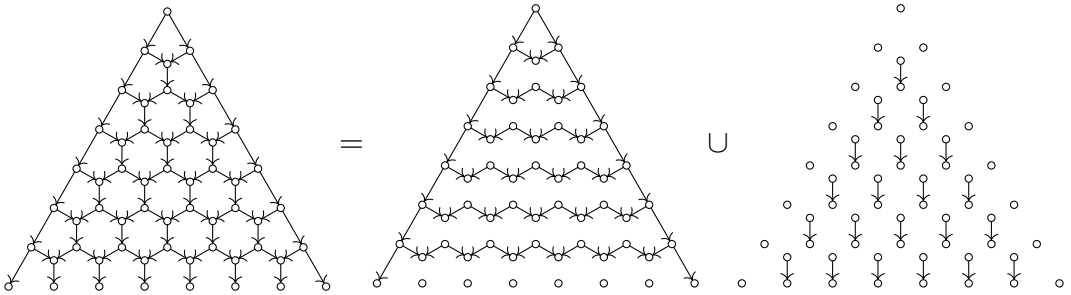
確かめてみるとわかるんですけど，クラウンだったら表裏付けた (1010) と (0101) の 2 通りしかなく，N 型だったら実は 1 通りしかありません．M 型だと山の数，アークの本数÷ 2 通りだけあって，そして W 型だと制約条件を満たすやり方がないんですね．4 パターンの分類のうち W 型だけないので，分解して W 型が 1 個でも出てきちゃったら，全域系統樹はないっていえるわけですね．

定理 11 [Hayamizu2021, Corollary 4.6]**.** X 上の根付き二分系統ネットワーク N が tree-based であることは，N の極大ジグザグ・トレイル分解に W-fence が現れないことと同値である．

まとめると，全域系統樹が存在する/しないことの必要十分条件は W がない/あるってことだし，あとジグザグ・トレイルの中での選び方っていうのが独立に決められるわけですから，全域系統樹はそれぞれジグザグ・トレイルでの選び方の直積の形で書けるというのが，この構造定理の言ってる主張であります．

これでもう解の全部を陽に書くことができたので，数えるとか列挙するとかっていう問題もそれぞれ分解をして，分解した後に自明になっちゃうわけですね．ジグザグ・トレイル分解は 1 回ずつアークを調べればできる話なんで，線型時間 $\Theta\big(|A(N)|\big)$ でできます．それが終わったら，それぞれのトレイルの中で選択肢が何通りあるかっていうのを数えてやればいい．W 型が現れて 0 通りなら全域系統樹ないんだなってわかるし，ある場合はたとえば 3 個あるわっていう風にわかる．また全域系統樹の列挙は，個々の極大ジグザグ・トレイルの中で一個一個組み替えておけばいいということになります．

■極大ジグザグ・トレイル分解の応用　簡単なデモンストレーションとしてはこんなのがあります．ジグザグ・トレイル分解しました．



G　　G の M-fence たち　　G の N-fence たち

このネットワークは結構複雑に見えます．全域系統樹の数は $7! = 5040$ 個あって，多いなって思う．けど，8 個のリーフを持つ系統樹は実は $2^{21} = 2097152$ 個で，全然 5040 個より多いです．根付き 2 分系統ネットワークに限っても $(2|X| - 3)!!$ で 13 万個ぐらいあるので．何せ 5040 個より全然多いんですよ．だからこのネットワーク，何の情報もなさそうなんだけど，あり得る進化のシナリオ全体から候補を結構絞り込んでくれてると分かります．また，どのぐらいネットワークが複雑なのかっていうのも定量化もできたりとかするって感じですね．

今日いってなかったですけど，最適化は要するに「パズルのピースの中で最強のパーツを集めれば全体でも最強になる」って話なんで，1 個求めるのはすごい簡単なんです．ランキングはね，若干難しいんです．でも構造定理を応用すると何個あっても上位だけ抽出するってことが高速にできるようになります．

■まとめ　というわけで今日の話をまとめます．系統ネットワークと系統樹の組合せ論っていう見慣れない話をしてきましたけれども，皆さん楽しめてもらえたんじゃないかなと思います．今日はできてるところだけ話しましたけど，できてないことがまだまだいっぱいいろいろとあって面白い分野なので，興味がある人がいたら．ちょっと動画見るなり記事を見るなりして，ぜひ調べてみたりしてみてください．ありがとうございました．

参考文献

[Doolittle1999]　W. F. Doolittle, “Phylogenetic Classification and the Universal Tree,” *Science* **284** (1999), no. 5423, 2124–2128.

[Francis–Steel2015]　A. R. Francis and M. Steel, “Which Phylogenetic Networks are Merely Trees with Additional Arcs?,” *Systematic Biology* **64** (2015), no. 5, 768–777.

[Hayamizu2021]　M. Hayamizu, “A Structure Theorem for Phylogenetic network and its implications for tree-based networks,” *SIAM Journal on Discrete Mathematics* **35** (2021), no. 4, 2490–2516.

[Puigbò–Wolf–Koonin2013]　P. Puigbò, Y. I. Wolf and E. V. Koonin, “Seeing the Tree of Life behind the Phylogenetic forest,” *BMC biology* **11** (2013), no. 1, 1–3.