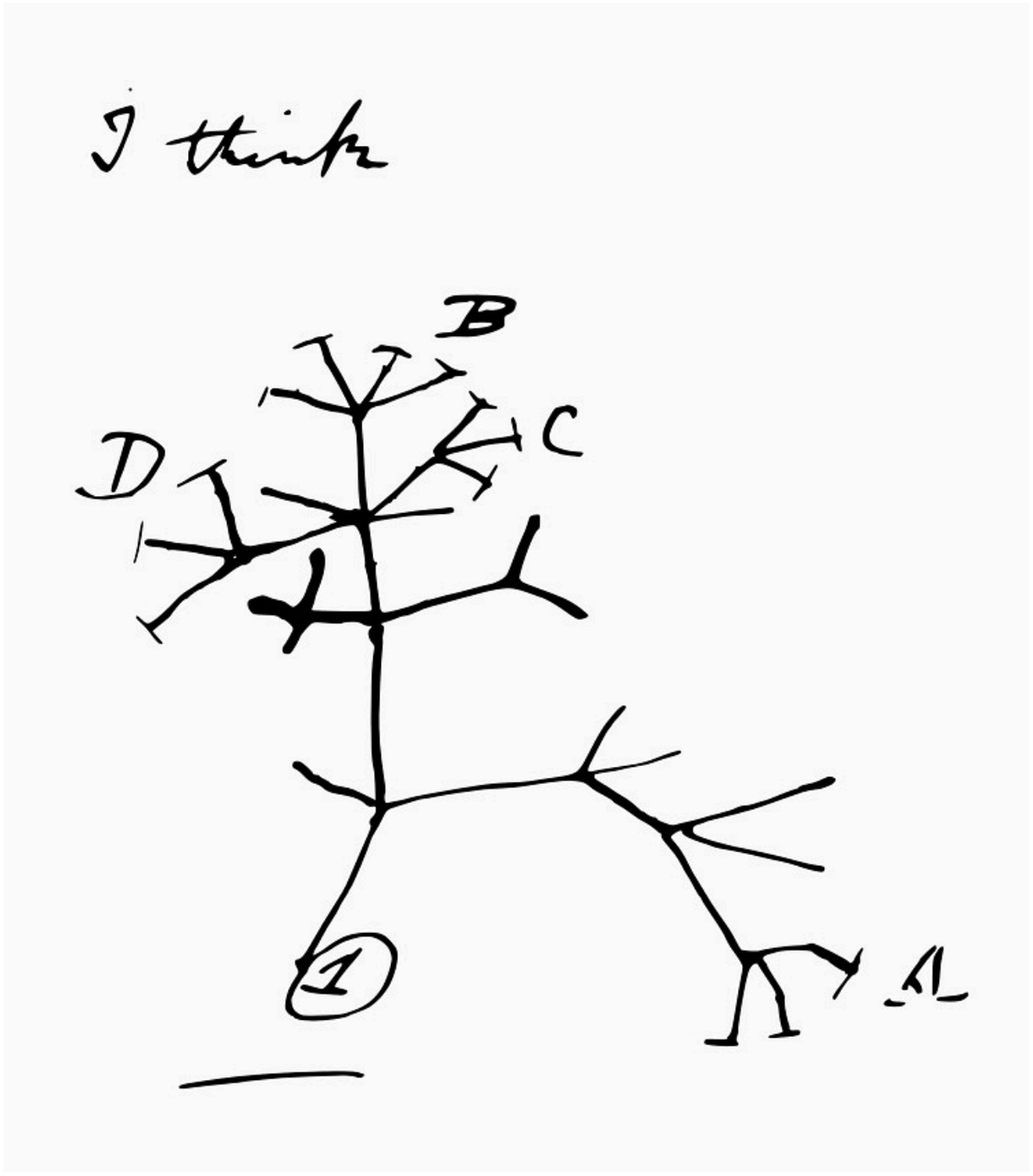
Species tree inference and update on very large datasets using approximation, randomization, parallelization, and vectorization



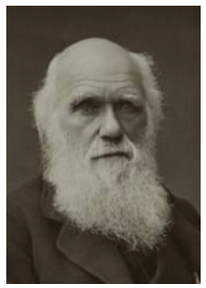
Siavash Mirarab

Electrical and Computer Engineering
University of California at San Diego

Phylogenetic reconstruction from data

Gorilla

`ACTGCACACCG`



Human

`ACTGCCCCCG`

Chimpanzee

`AATGCCCCCG`

Orangutan

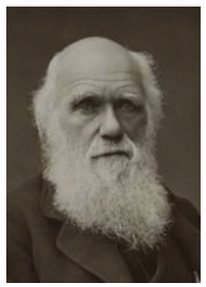
`CTGCACACGG`

Phylogenetic reconstruction from data

Phylogenetic reconstruction from data

Gorilla

ACTGCACACCG



Human

ACTGCCCCCG

Chimpanzee

AATGCCCCCG

Orangutan

CTGCACACGG

Phylogenetic reconstruction from data

Gorilla	Human	Chimpanzee	Orangutan
ACTGCACACCG	ACTGCCCCCG	AATGCCCCCG	CTGCACACGG

Gorilla	ACTGCACACCG
Human	ACTGC-CCCCG
Chimpanzee	AATGC-CCCCG
Orangutan	-CTGCACACGG

$$D$$

Phylogenetic reconstruction from data

Gorilla ACTGCACACCG

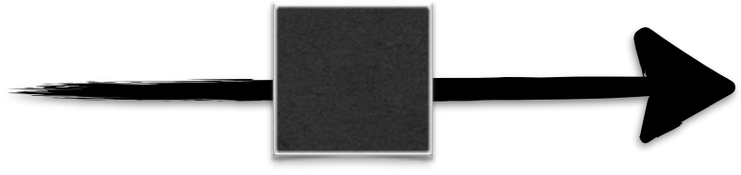
Human ACTGCCCCCG

Chimpanzee AATGCCCCCG

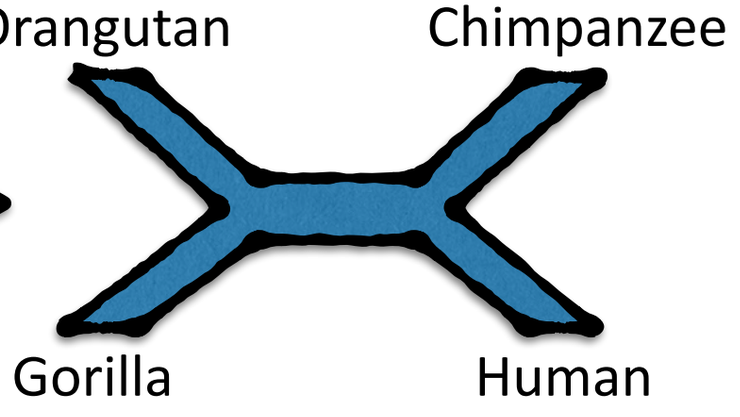
Orangutan CTGCACACGG

Gorilla ACTGCACACCG
Human ACTGC-CCCCG
Chimpanzee AATGC-CCCCG
Orangutan -CTGCACACGG

D

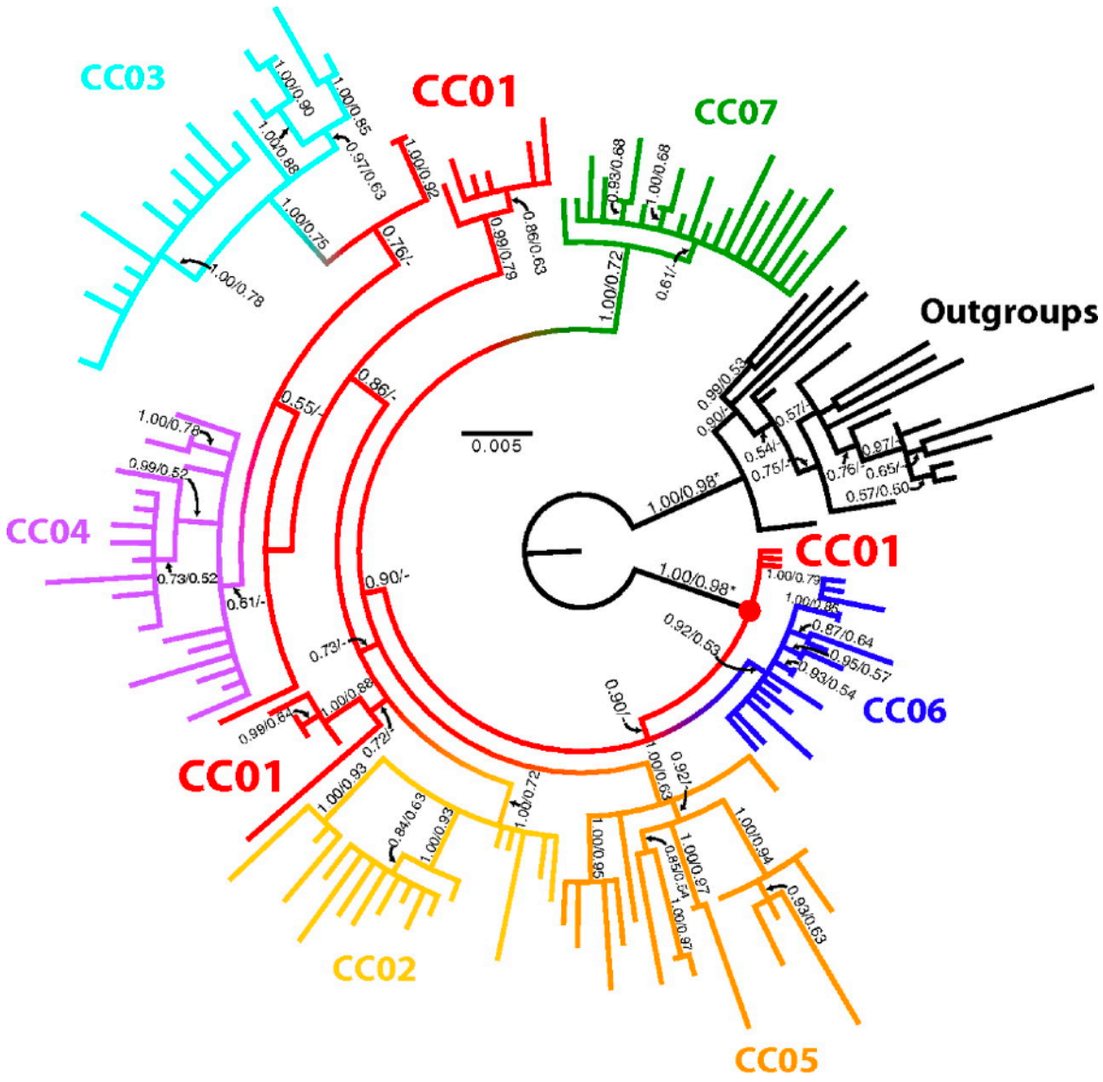


$P(D|T)$

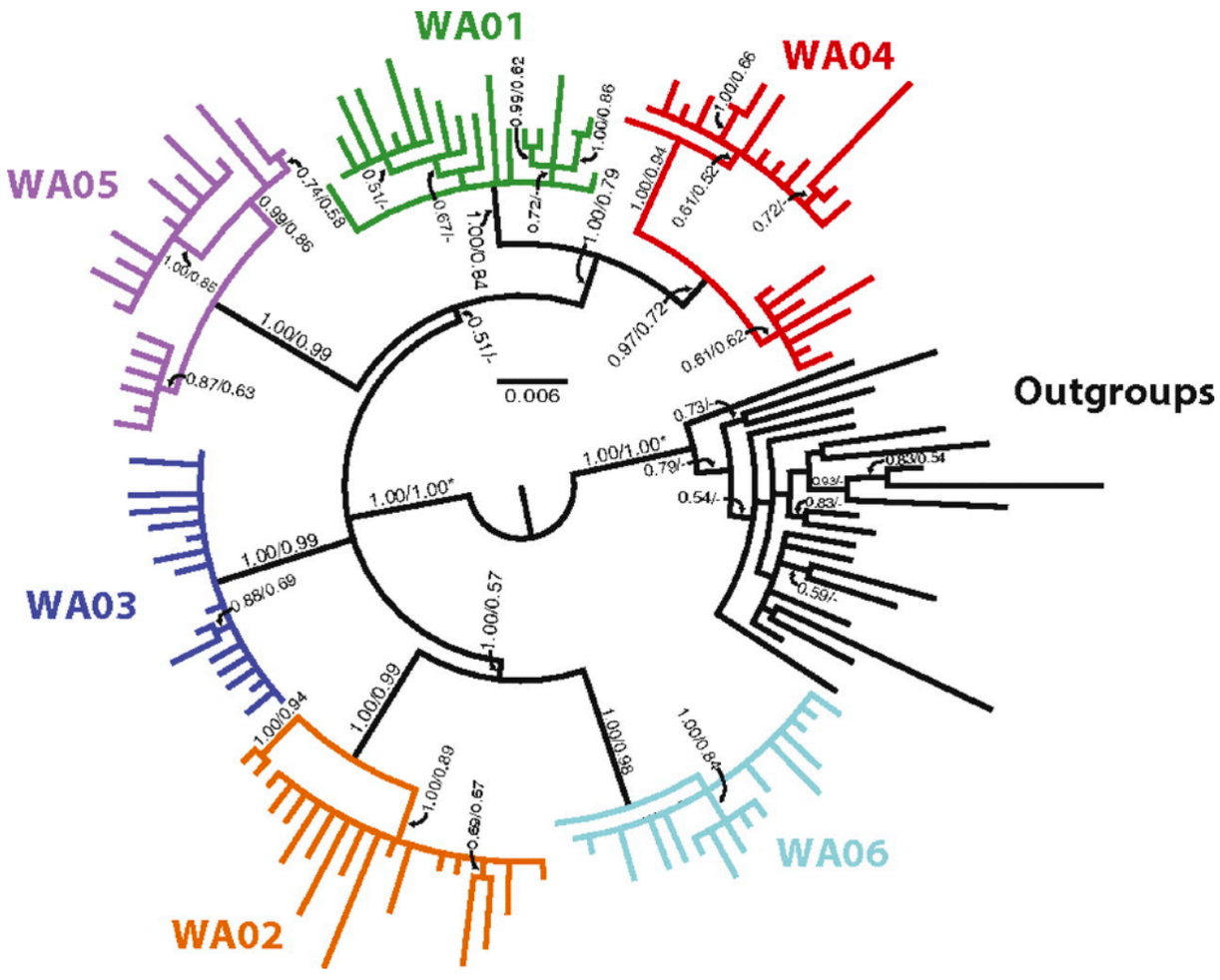


T

Applications: HIV forensic



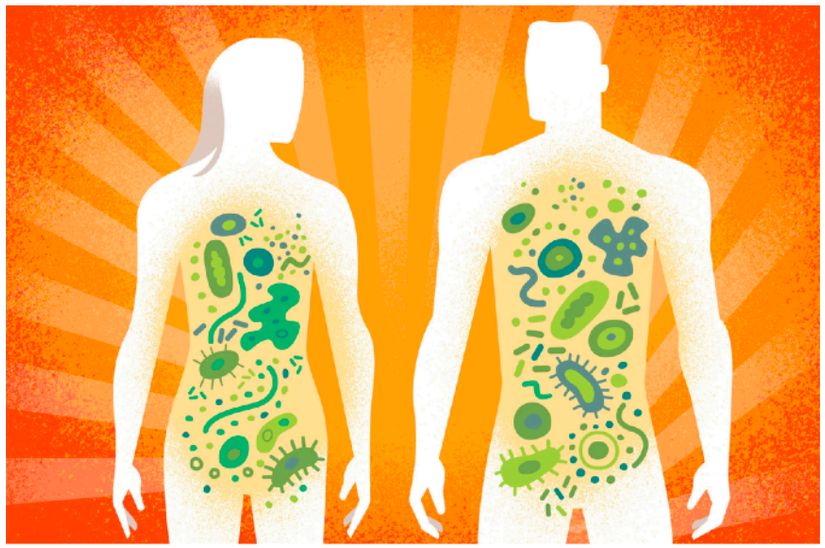
Texas case



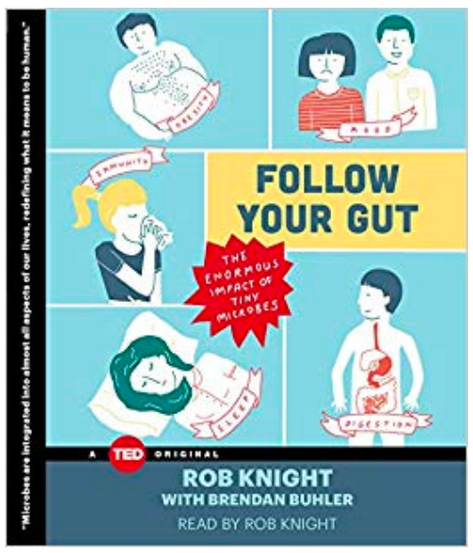
Washington case

Scaduto et al., PNAS, 2010

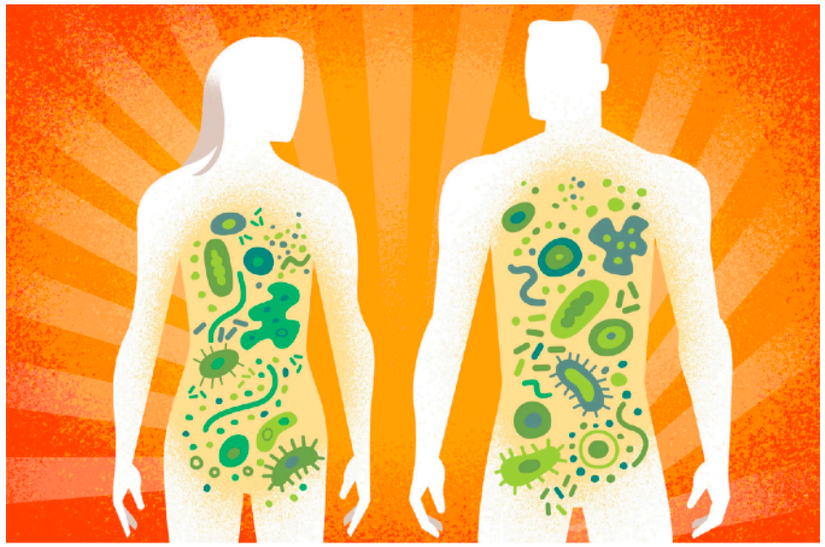
Applications: microbiome



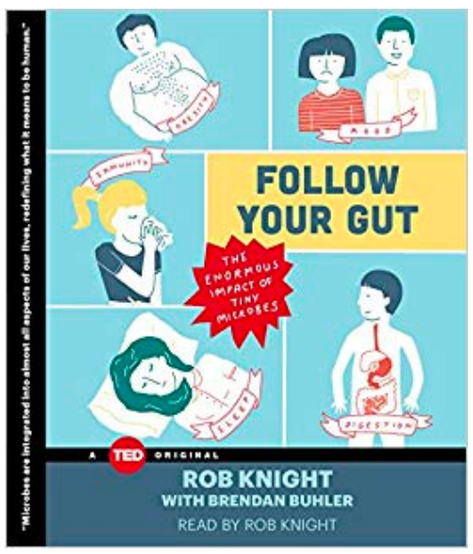
https://www.nytimes.com/2017/11/06/well/live/unlocking-the-secrets-of-the-microbiome.html

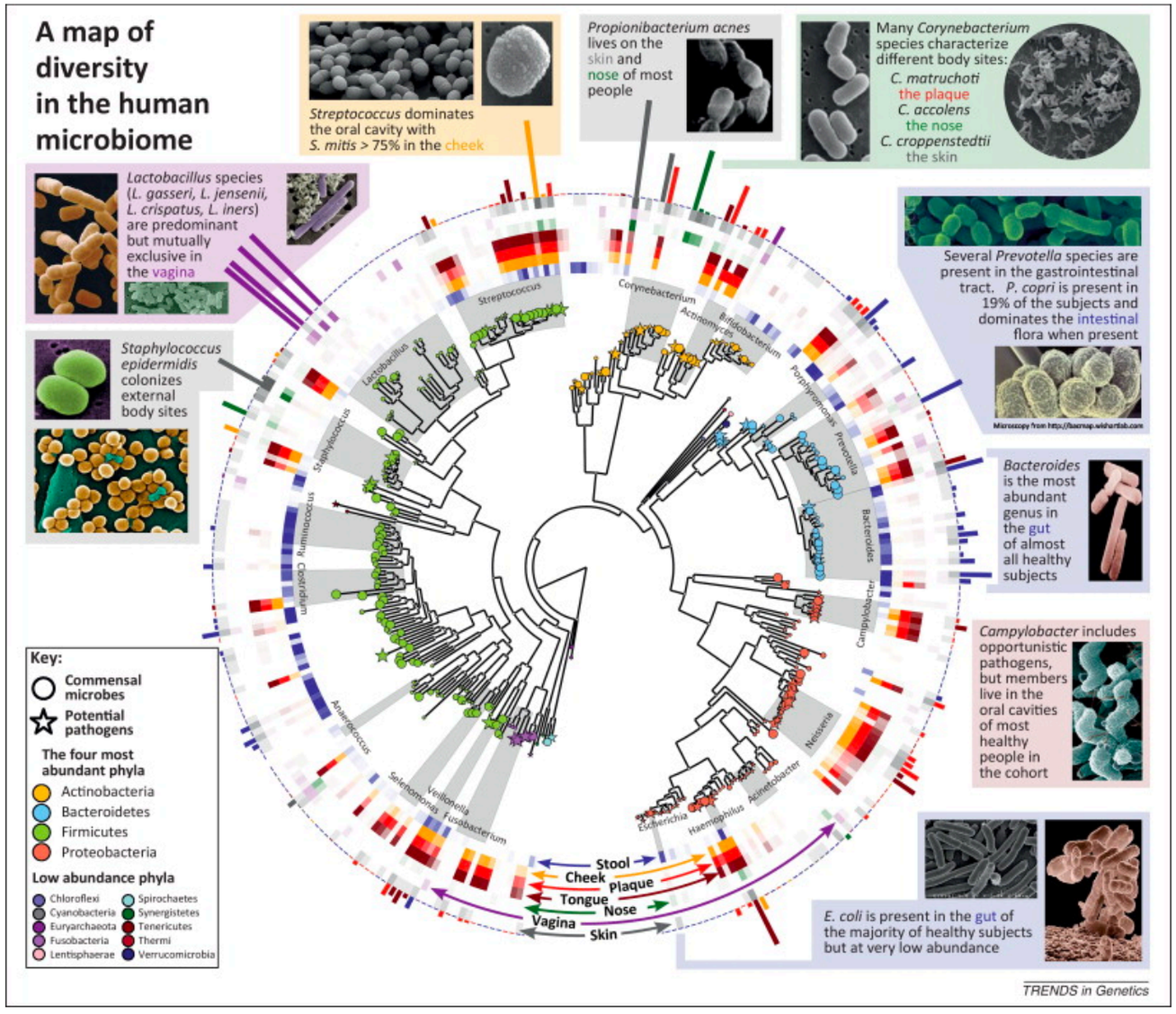


Applications: microbiome



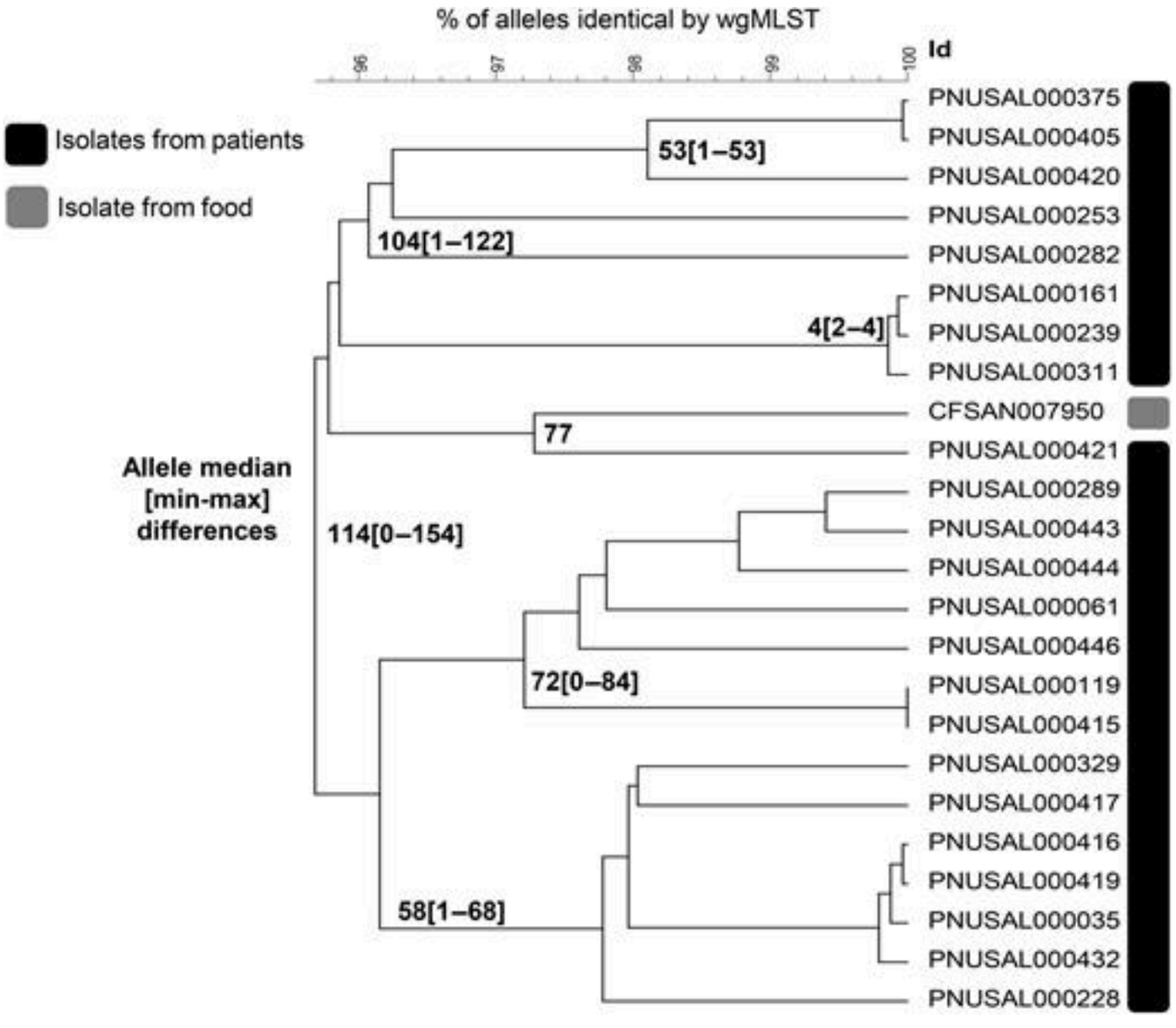
https://www.nytimes.com/2017/11/06/well/live/unlocking-the-secrets-of-the-microbiome.html





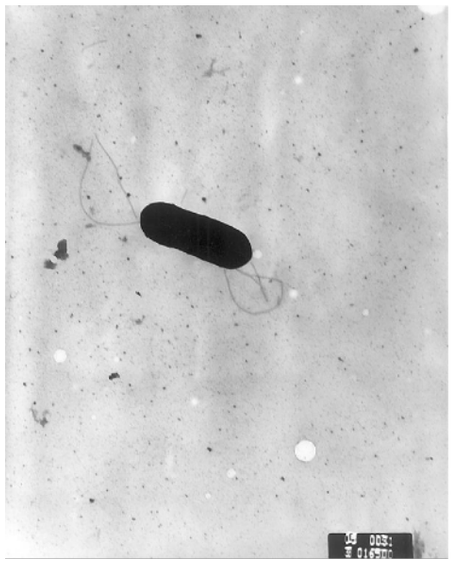
Morgan, Xochitl C., Nicola Segata, and Curtis Huttenhower. "Trends in genetics (2013)

Applications: food safety

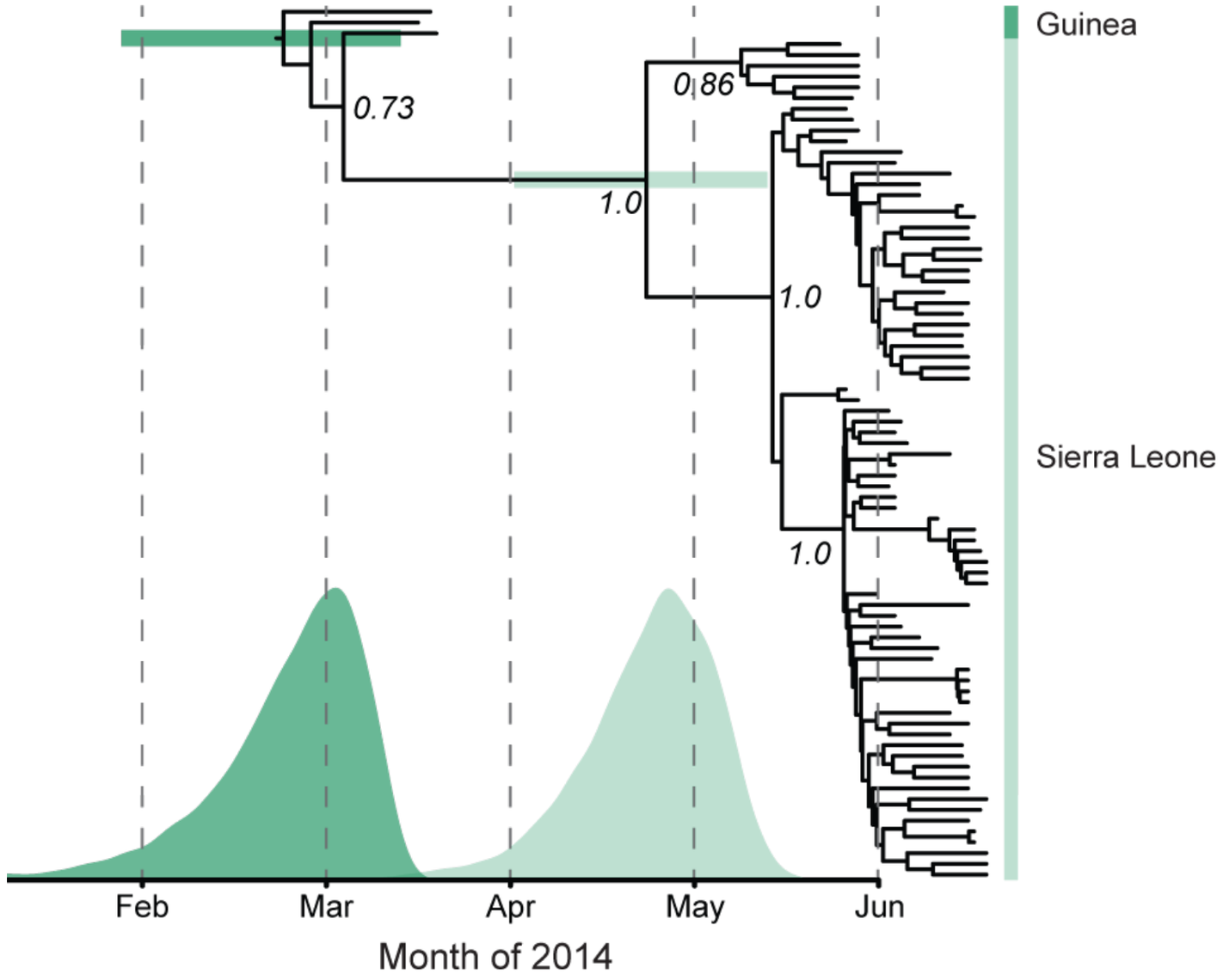


Jackson, Brendan R., et al. *Reviews of Infectious Diseases* (2016)

Tracking the source of a listeriosis outbreak



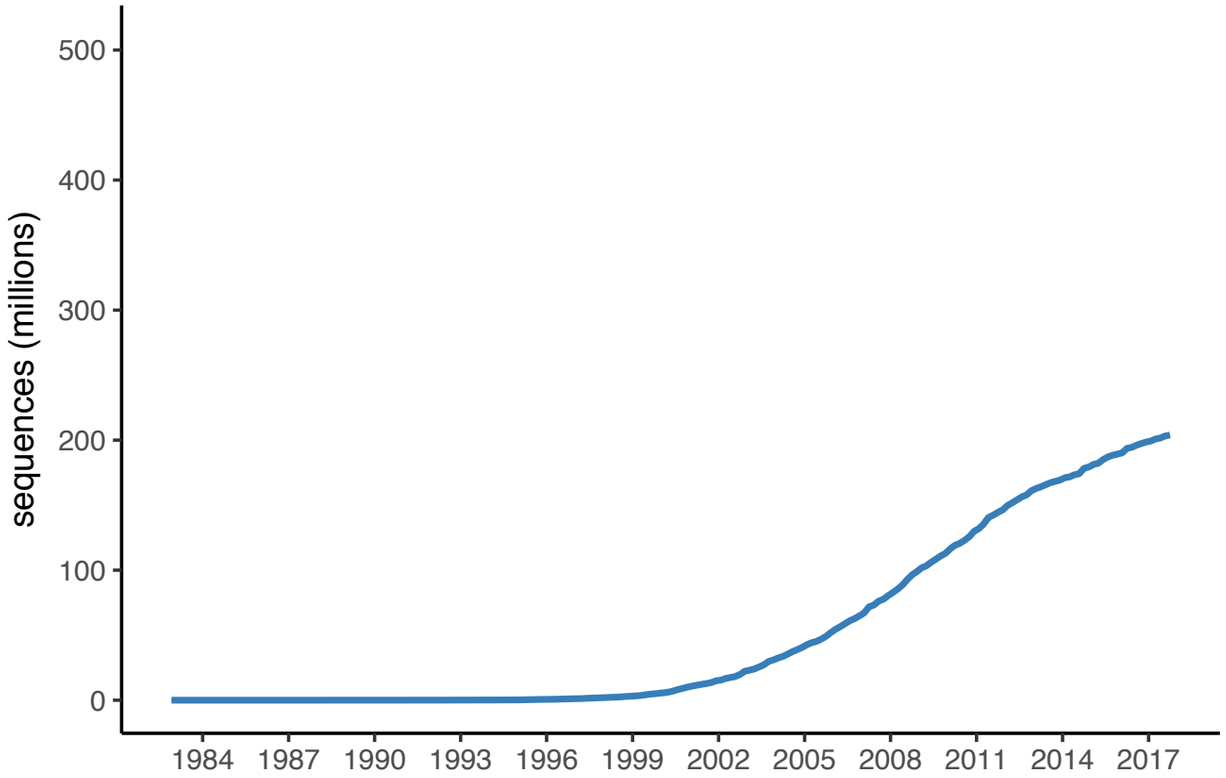
Applications: ebola



source: Gire et al., Science, 2014

Sequence data growth

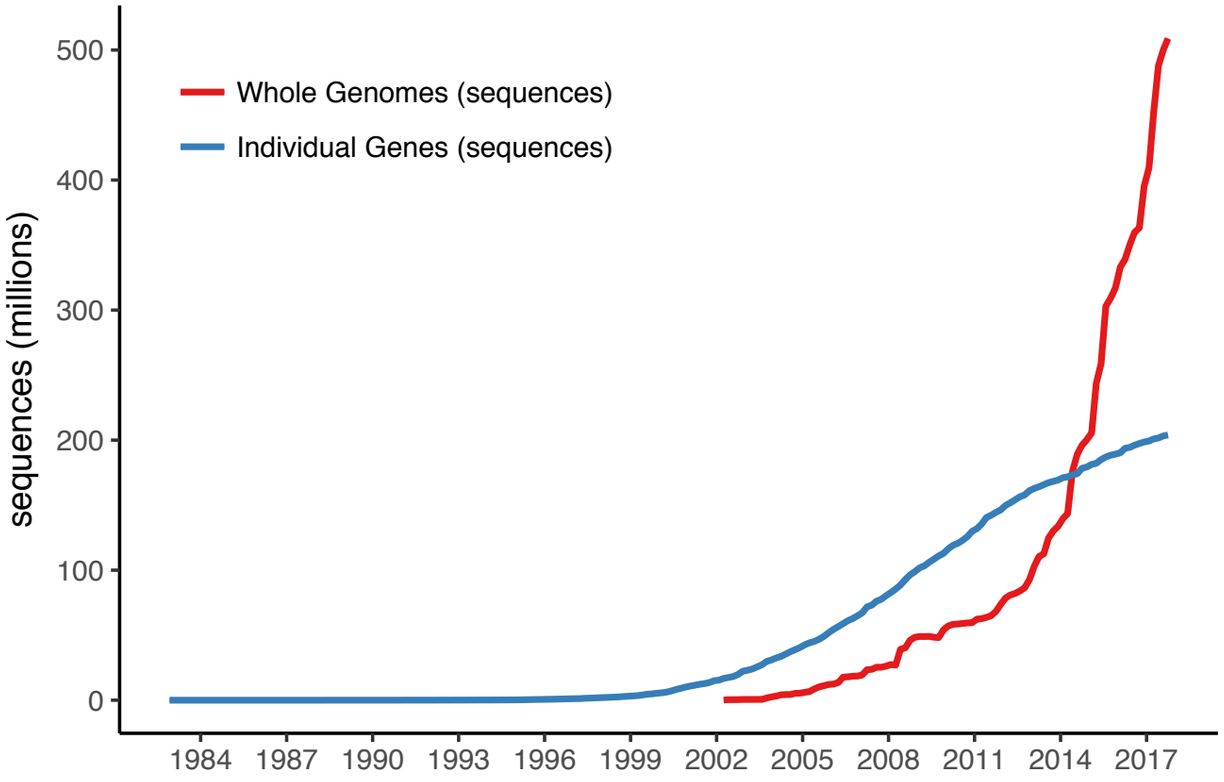
data source: NCBI (http://www.ncbi.nlm.nih.gov/genbank/statistics)



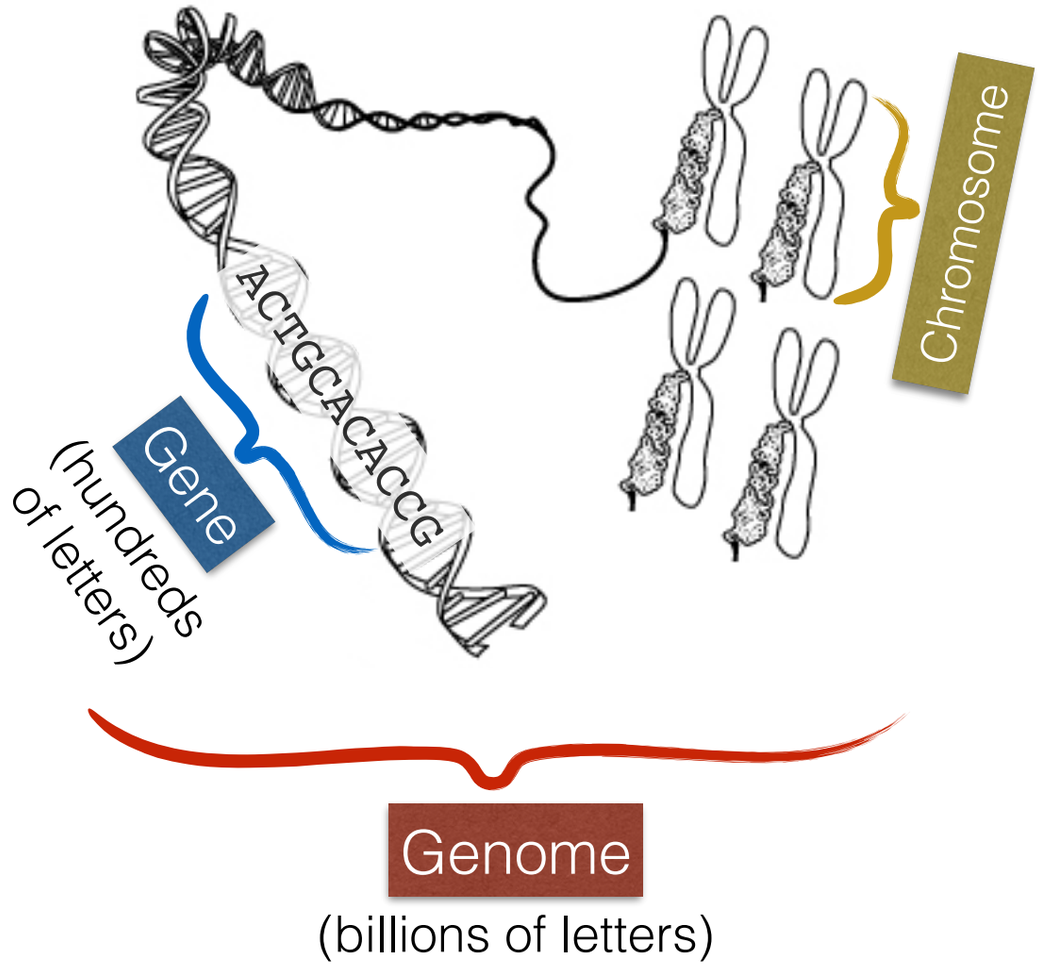
- Rapid growth in the available sequences

Sequence data growth

data source: NCBI (http://www.ncbi.nlm.nih.gov/genbank/statistics)



- Rapid growth in the available sequences

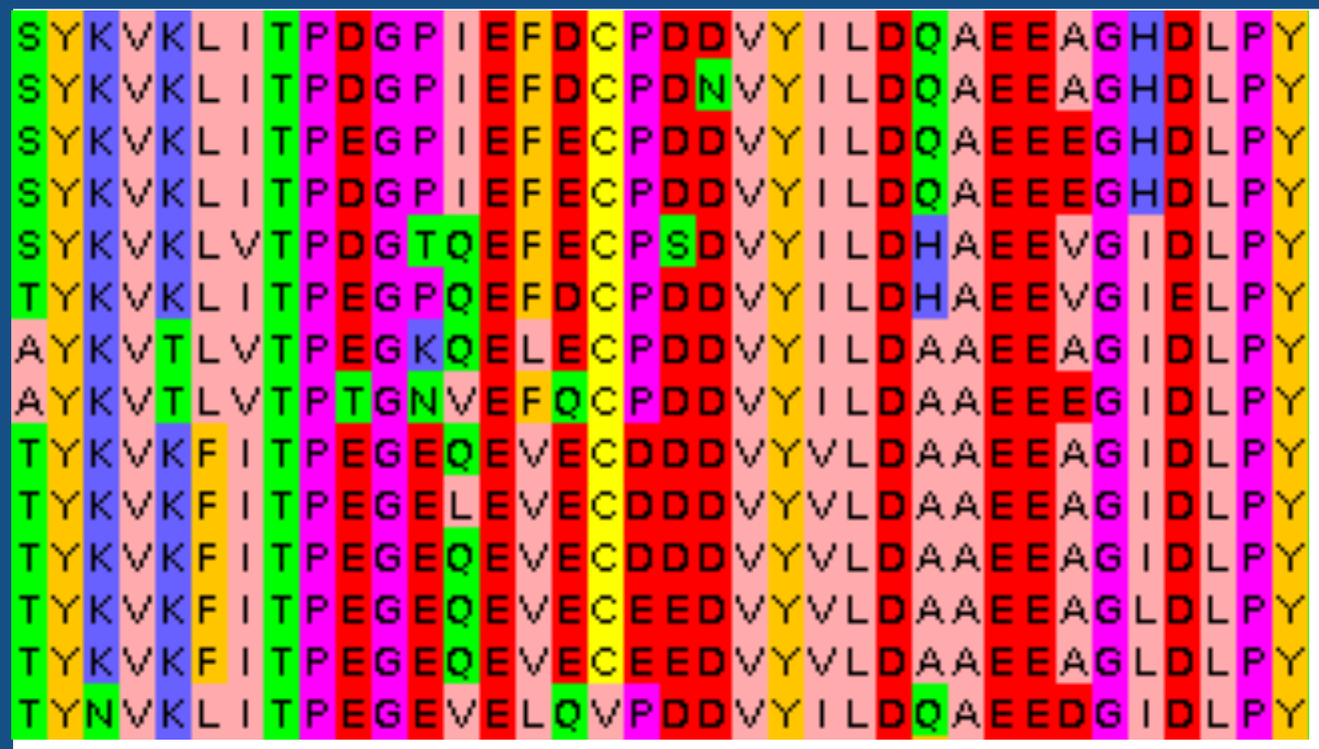


- Our focus has shifted to “whole genomes”

More genomic regions

Billions of columns

More sequences



A million rows

Phylogenetic reconstruction : a computational nightmare

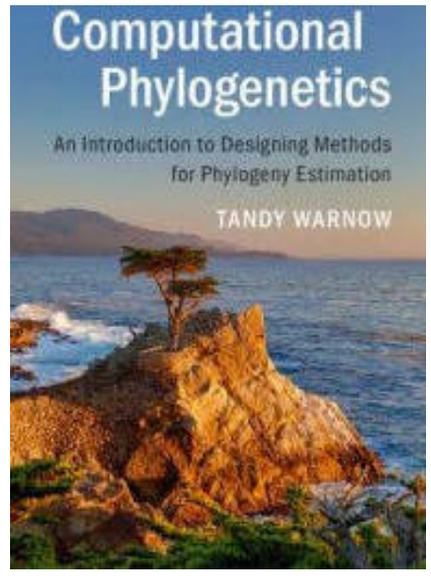
- Almost all problems are NP-Hard!

- The search space is huge.

  - Focusing on topology alone, there are $(2n\text{-}5)!!$ trees with **n** leaves

  - We also care about branch lengths: $\mathbb{R}^{2\text{n-}3}$

- We are interested in n in $10^1 \ldots 10^6$ range



Tandy Warnow

Dealing with uncertainty

- Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support

Dealing with uncertainty

- Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support

- Genome evolution is complex. We need complex statistical models for accurate inference

Dealing with uncertainty

- Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support
- Genome evolution is complex. We need complex statistical models for accurate inference
- We need extensive simulations to test methods

To scale to large datasets …

- Approximate and heuristic solutions

To scale to large datasets …

- Approximate and heuristic solutions
- Make the problem easier
- Divide-and-conquer
- Constrained search

To scale to large datasets …

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.

How about accuracy?

Increased data *can* make problems easier, but …

How about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
- Are used to answer harder problems
- Allow the use of more complex models
- Riddled with erroneous data

How about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
  - Are used to answer harder problems
  - Allow the use of more complex models
  - Riddled with erroneous data
- Often, methods lose their accuracy on large datasets

Examples of improving scalability

To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.

ASTRAL

- Optimization problem (NP-Hard):

  <u>Find the median tree among a set of input trees</u>

  Distance between two trees := the number of quartet trees they do not share

ASTRAL

- Optimization problem (NP-Hard):

  <u>Find the median tree among a set of input trees</u>

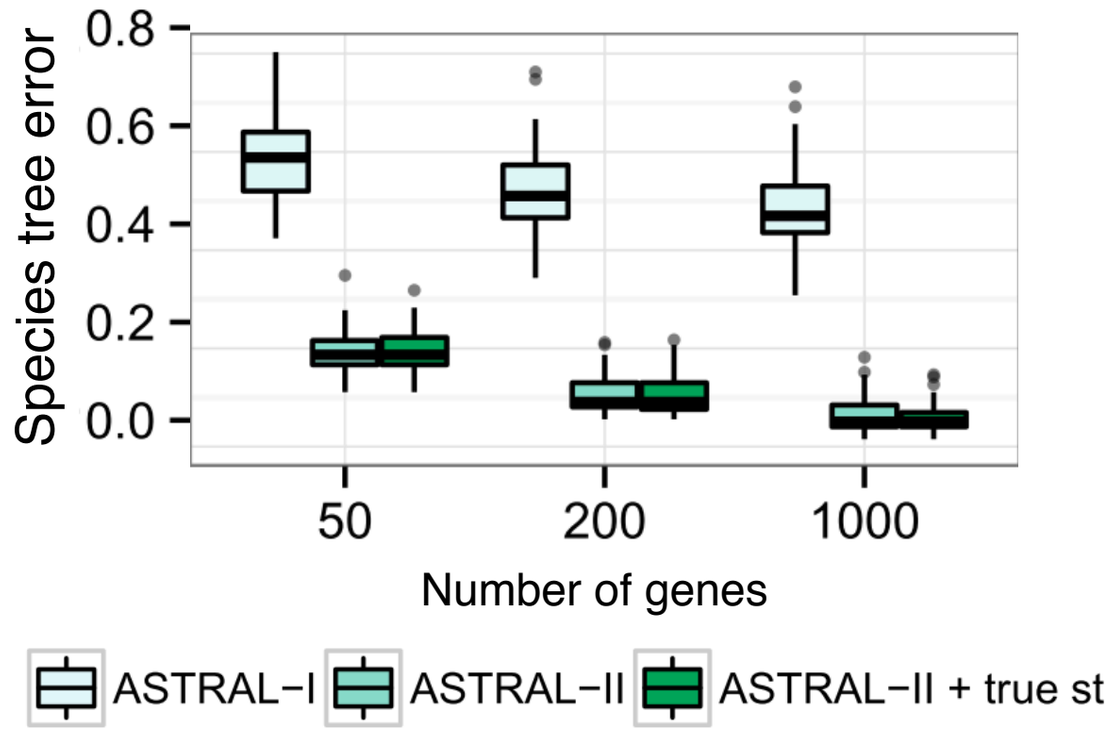
  Distance between two trees := the number of quartet trees they do not share

- ASTRAL: an exact solution using a dynamic programming algorithm

  - Solves the problem exactly on a constrained search space

Choosing Constraints

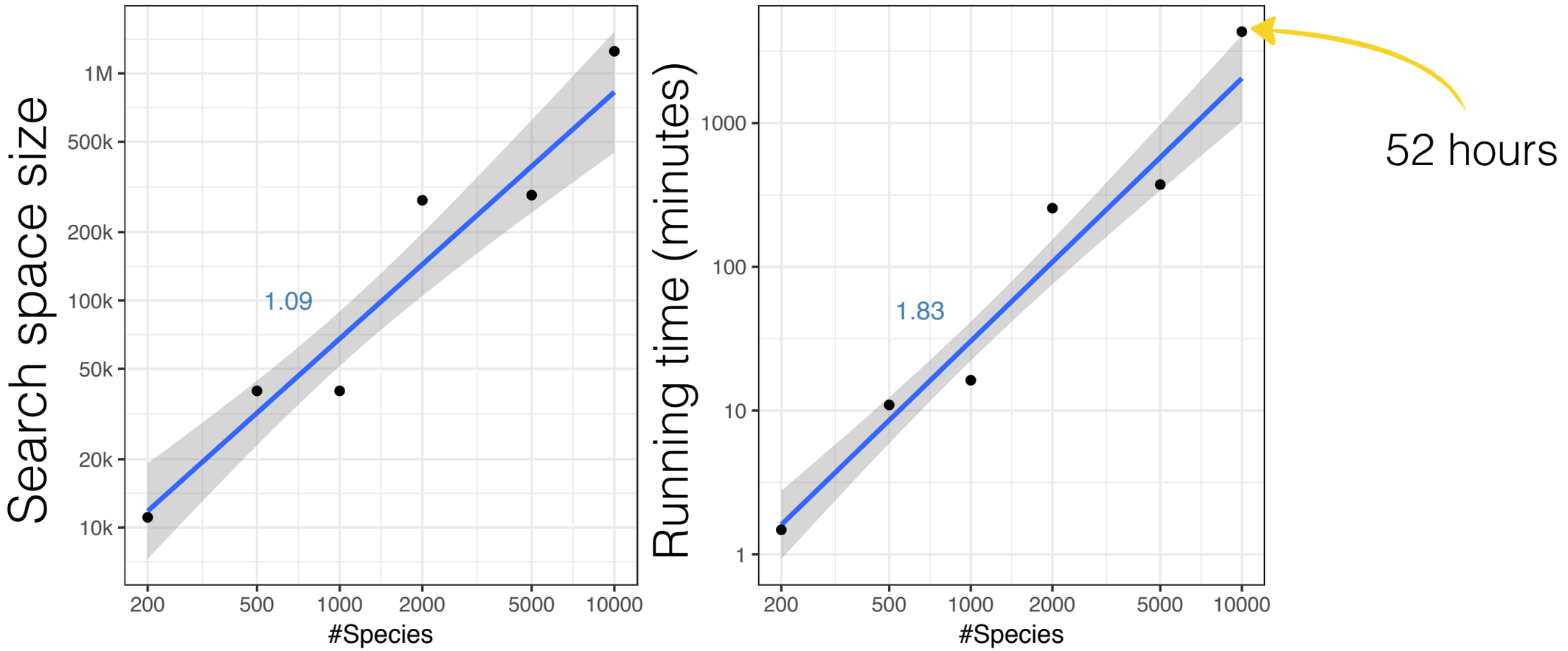
- Restricted search space should
  - Not compromise statistical consistency
  - Be large enough that accuracy is not sacrificed
  - Be small for scalability



ASTRAL evolution

- Search space should ideally grow polynomially with the dataset size
  - Tandy talked about ASTRAL-I and ASTRAL-II, which do not such have guarantees
- ASTRAL-III [Zhang et al., 2018]
  - Guarantees polynomial size search space
  - Increases the speed of the dynamic programming

Changing the number of species (n)

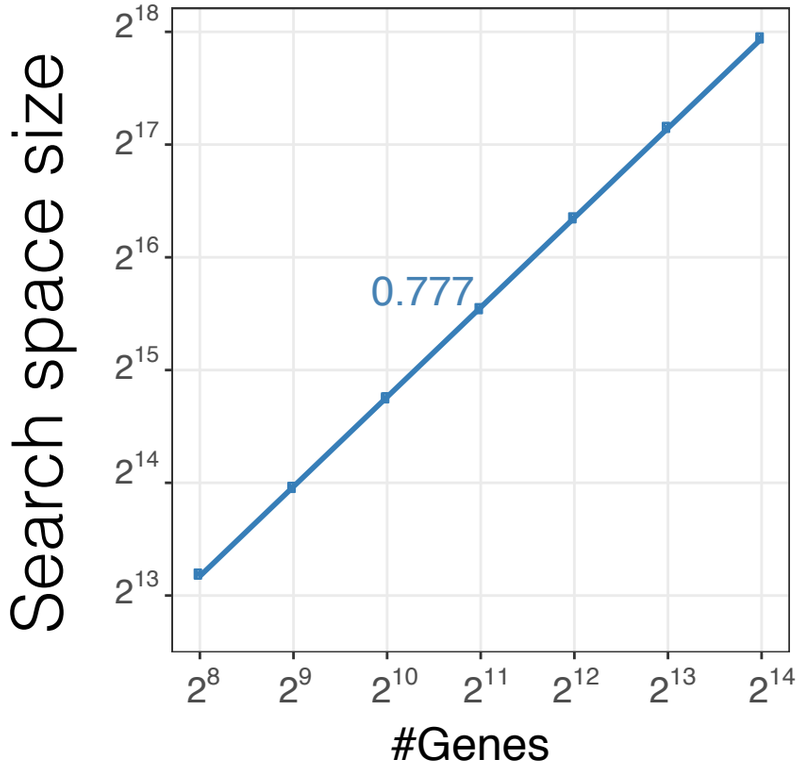


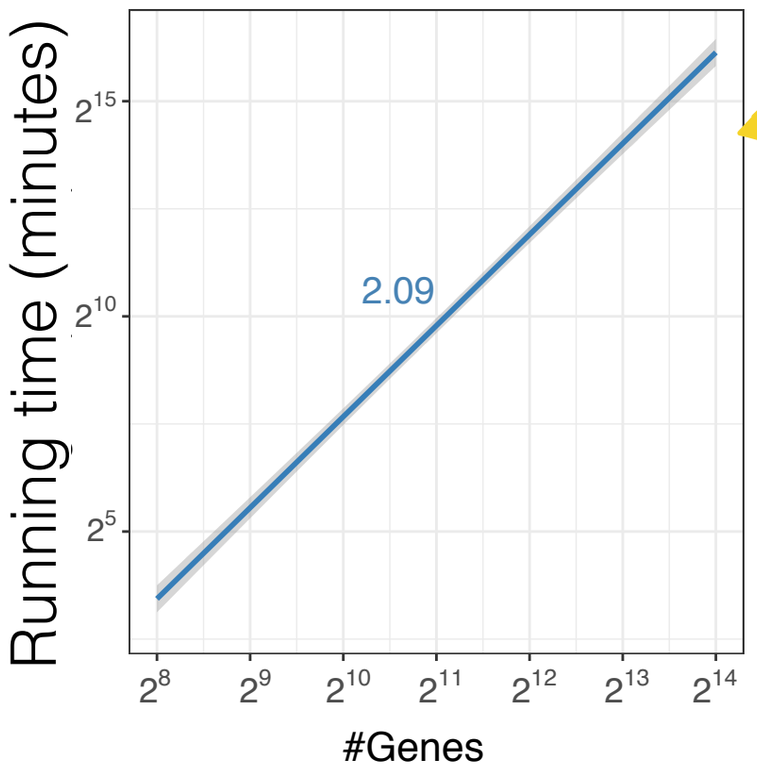
Simulations: n = 200 to 10,000 ,k = 1000 gene trees

Empirical running time seems to increase close to $O(n^{1.8})$

[Zhang et al., 2018]

Changing the number of genes (k)





17 hours

Simulations: n = 48 species, k = 256 to 16,384 gene trees

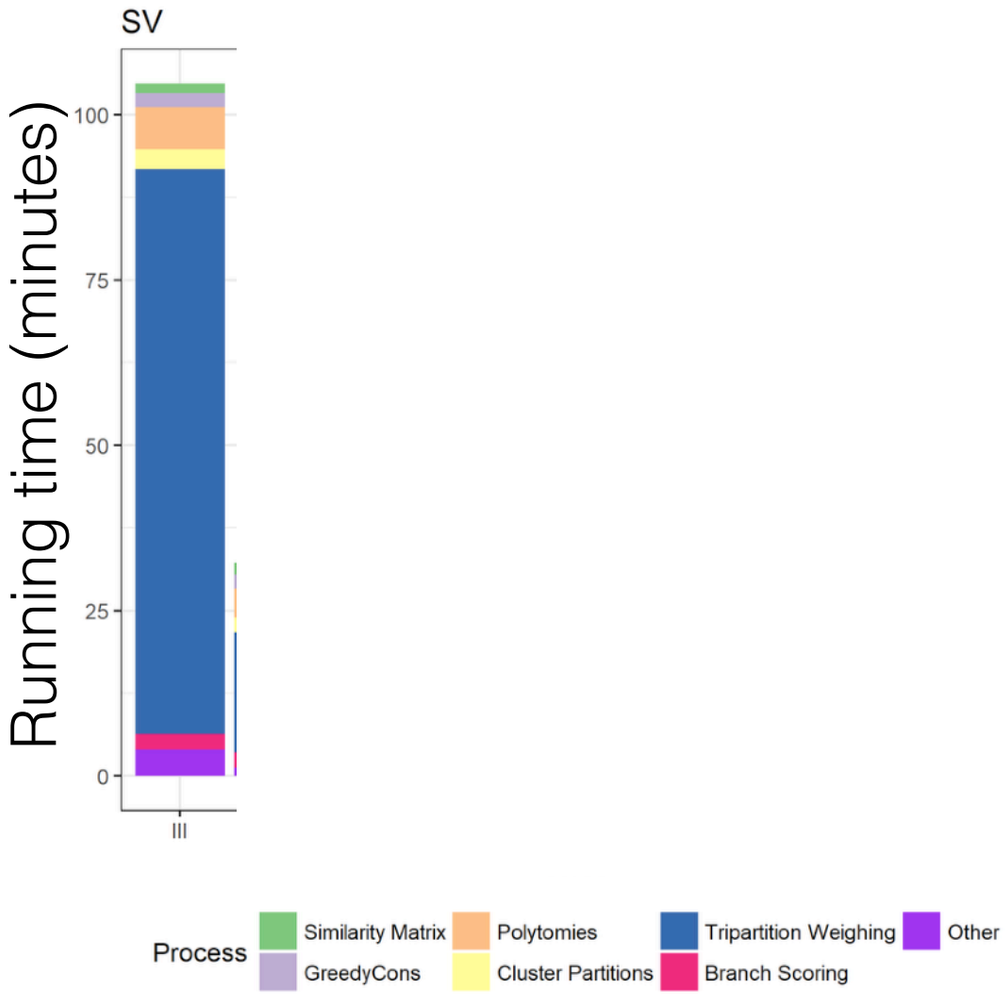
Empirical running time seems to increase close to $O(k^2)$

[Zhang et al., 2018]

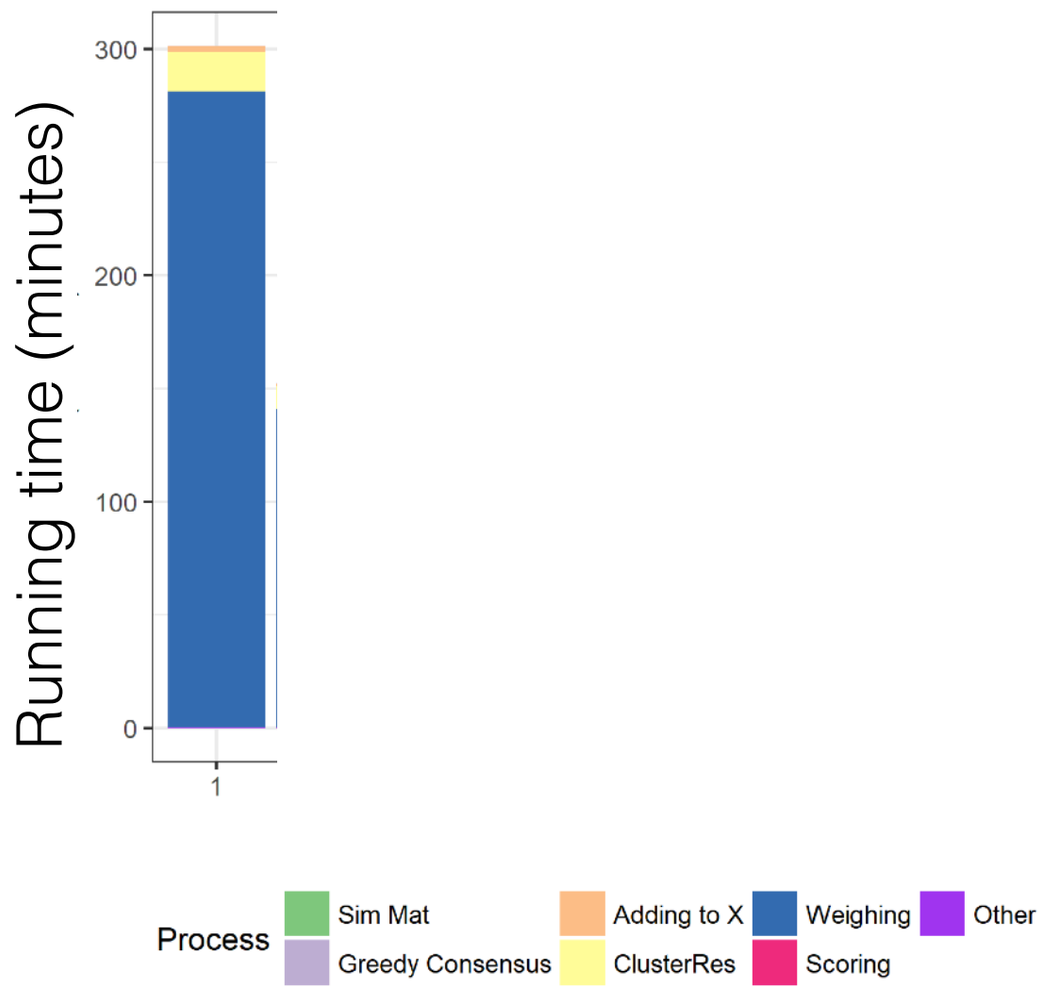
To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.

Profiling ASTRAL-III



Many species



Many genes

Further scaling improvements

Chao Zhang

- Developed a randomized algorithm for a key step in the dynamic programming.

  For a set X of subsets of an alphabet, find:

$$\{(A,B)\,|\,A,B \in X, A \cup B \in X, A \cap B = \emptyset\}$$

- Using, Abelian group theory, we can do compute this in time quadratic in $|\mathrm{X}|$.

  - Has an astronomically low probability of failure

Further scaling improvements

Chao Zhang

- Developed a randomized algorithm for a key step in the dynamic programming.

  For a set X of subsets of an alphabet, find:

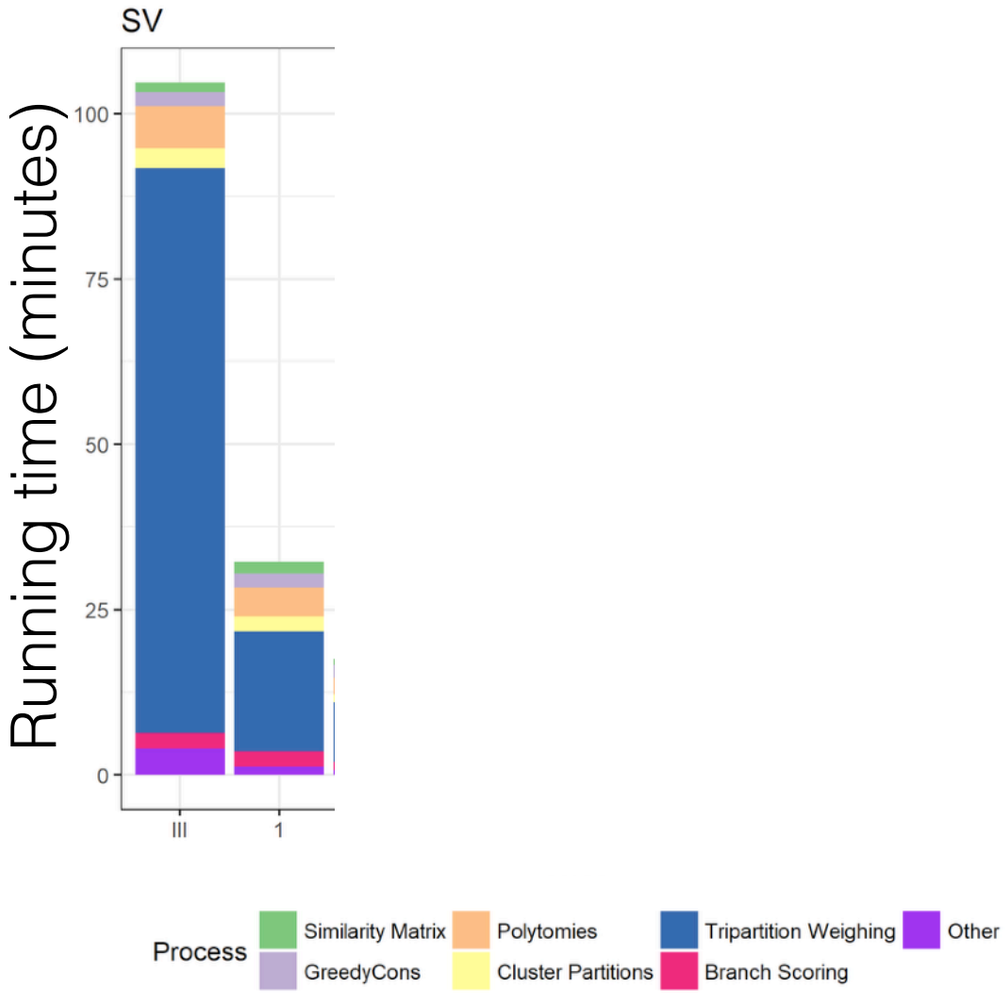
$$\{(A,B)\,|\,A,B\in X, A\cup B\in X, A\cap B=\emptyset\}$$

  - Using, Abelian group theory, we can do compute this in time quadratic in $|\mathrm{X}|$.

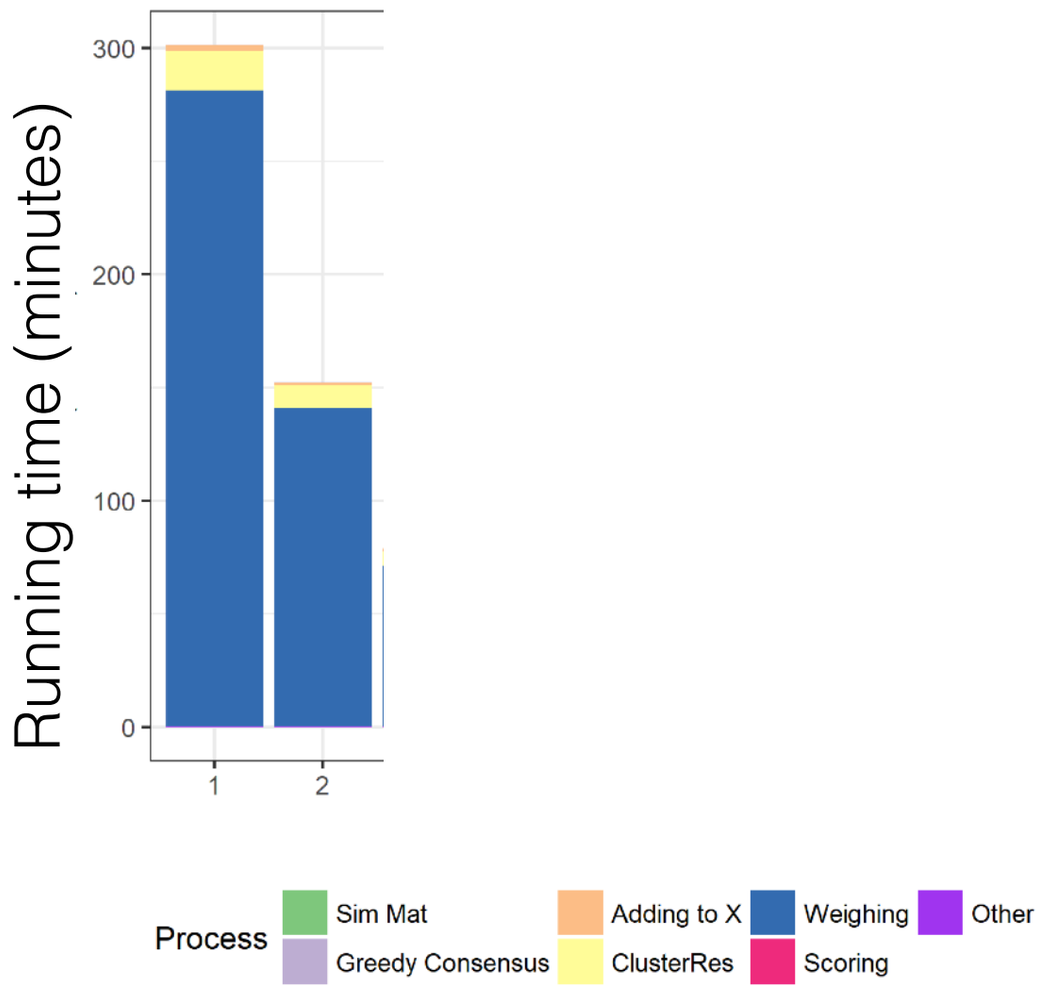
    - Has an astronomically low probability of failure

- Implemented the kernel in C++ and added vectorization.

Improved scalability

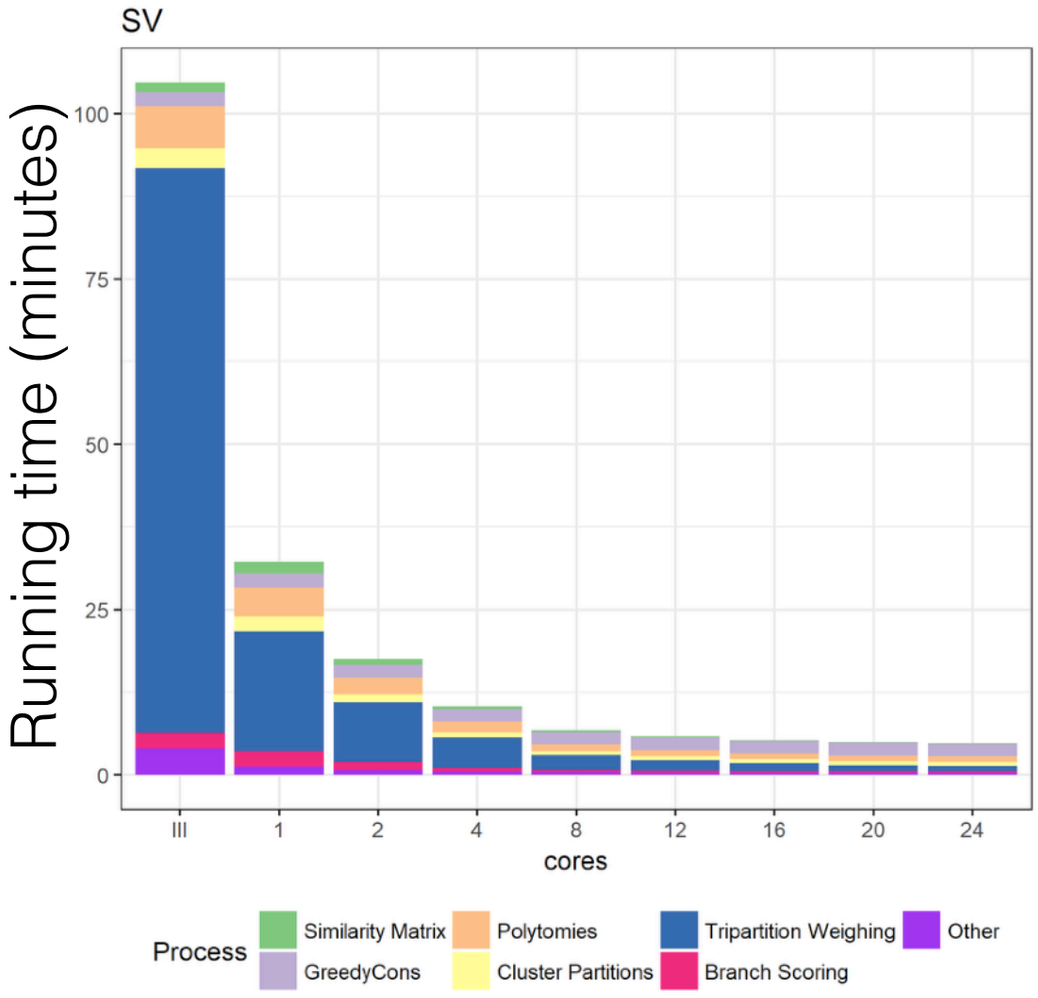


Many species

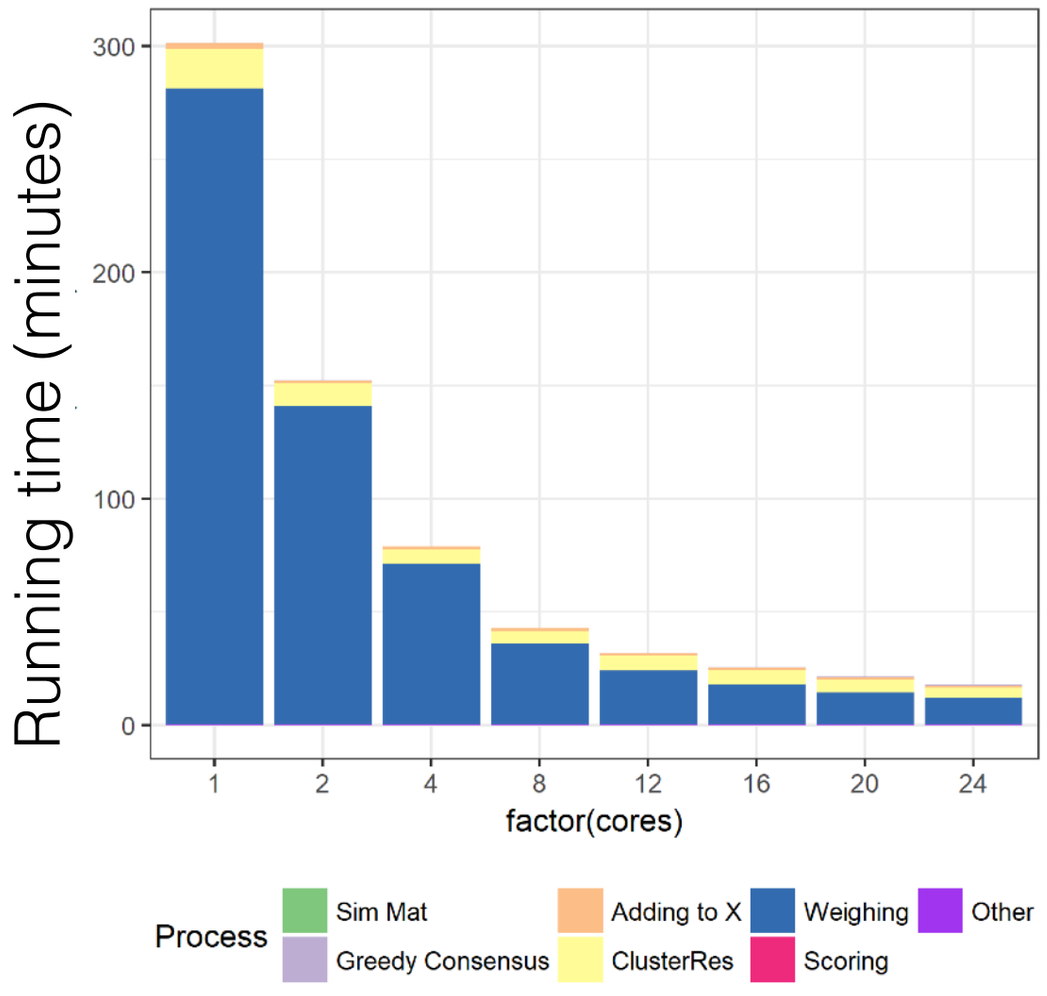


Many genes

Parallelize the main step of the dynamic programming (blue)

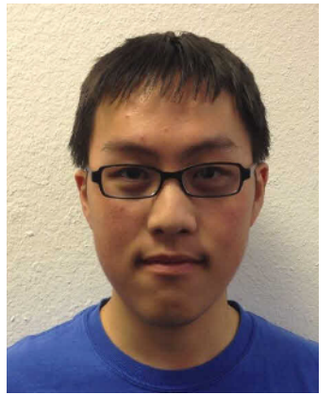


Many species



Many genes

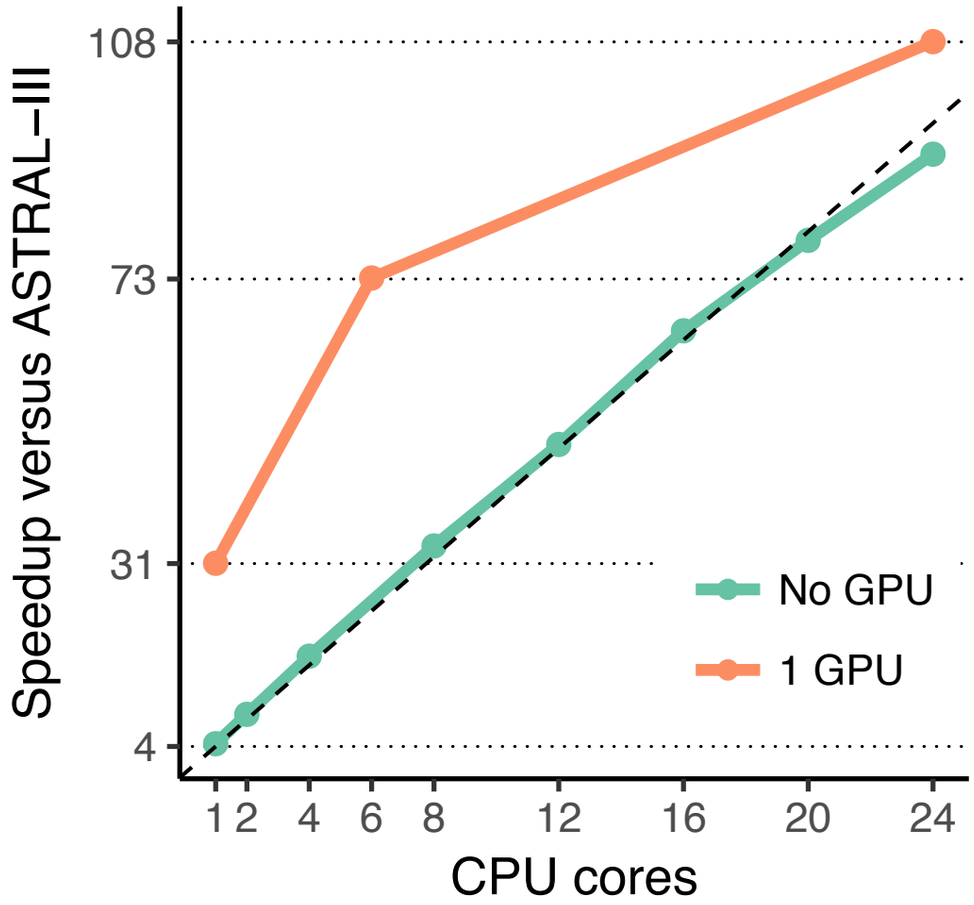
Scaling + GPU



John Yin

- Can analyze datasets with 10,000 species and 1000 genes in less than a day given 24 cores and a GPU

- https://github.com/smirarab/ASTRAL/tree/MP-similarity

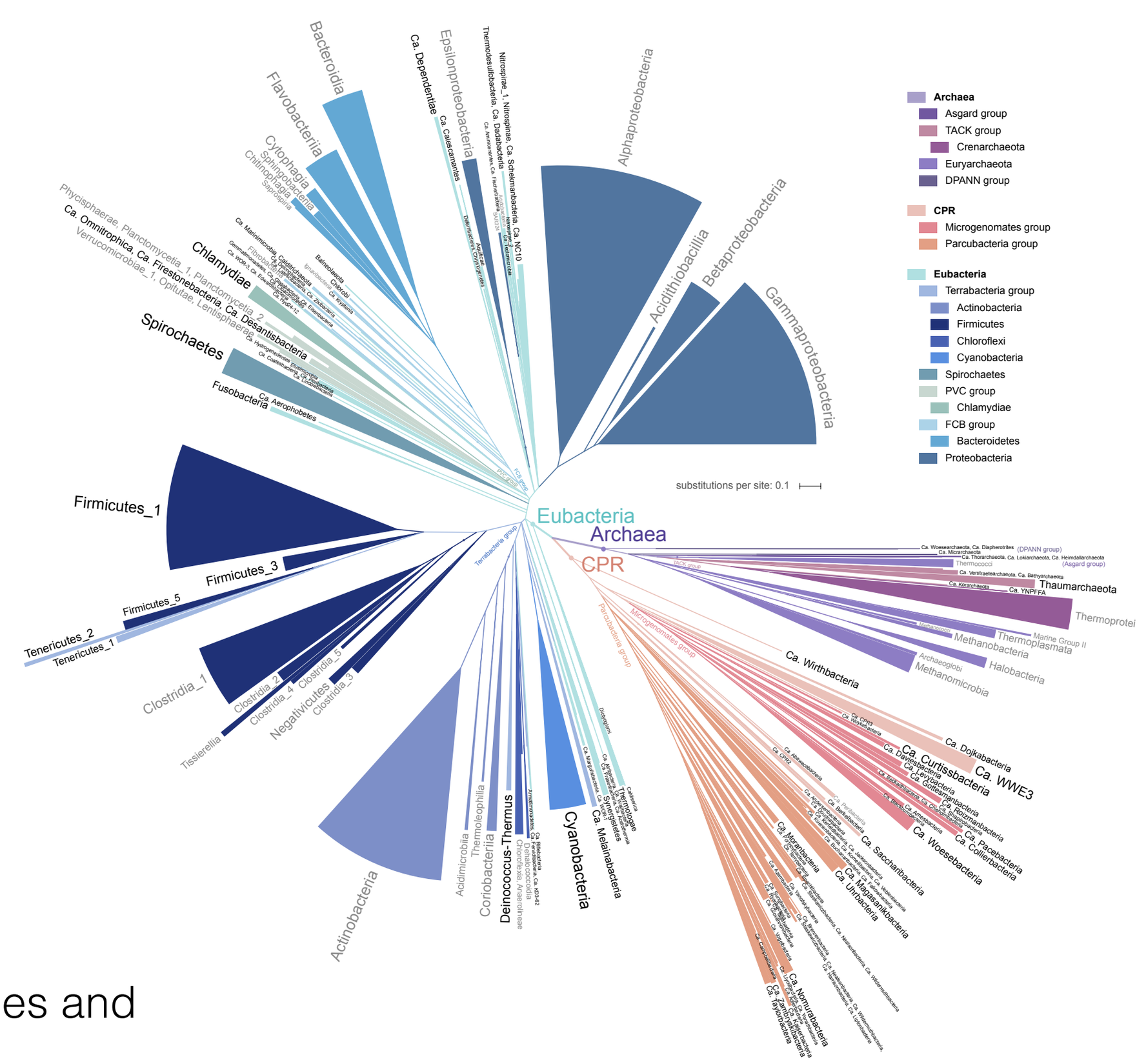


Many genes

Qiyun Zhu

Rob Knight

Uyen Mai



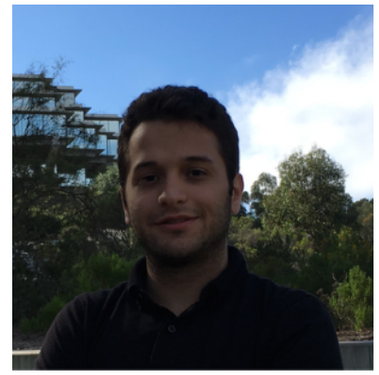
- 10,000 microbial species and 381 genes

- ASTRAL infers the tree in 24 hours (4 GPUs)

To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.

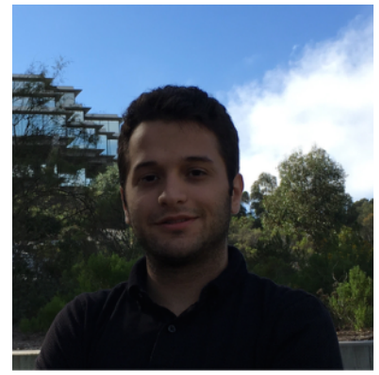
Statistical support



Erfan Sayyari

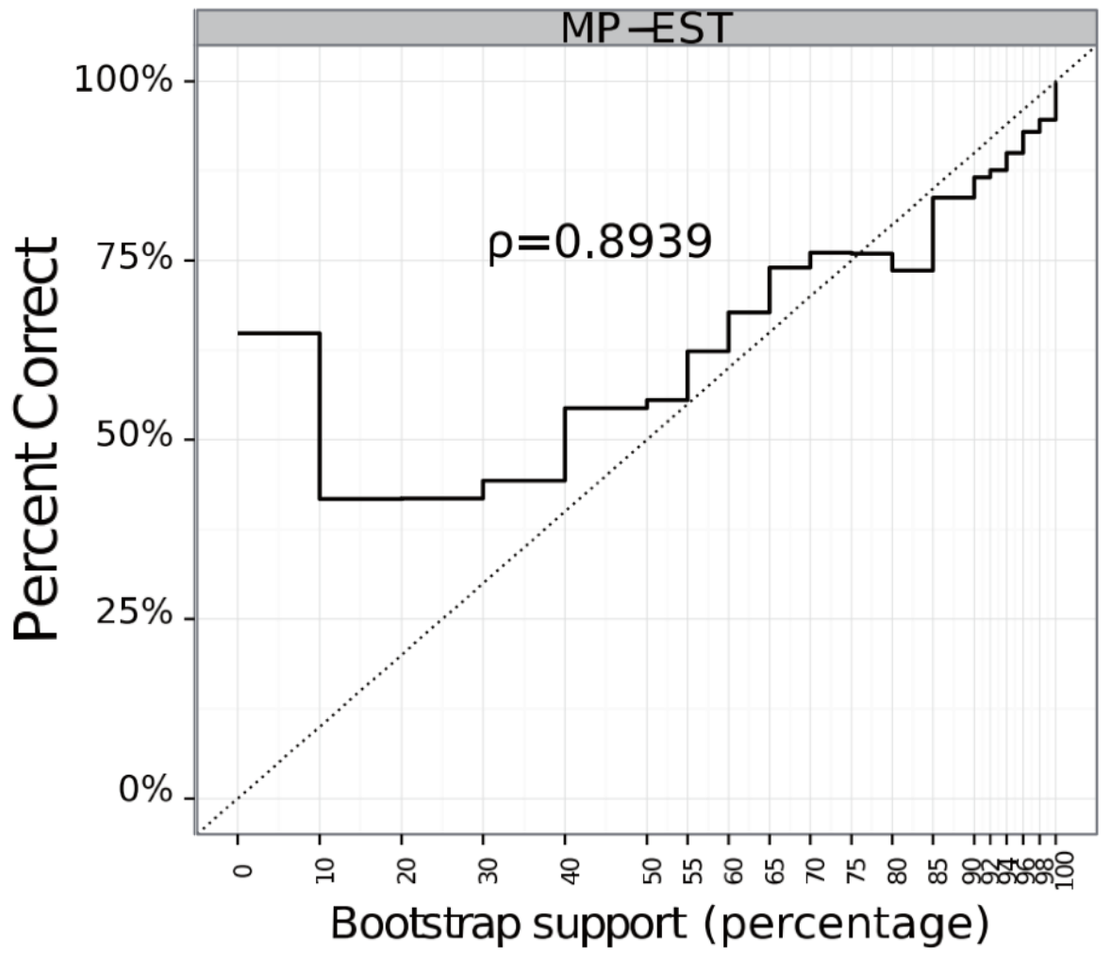
- Traditional approach:
  bootstrapping each gene, then
  bootstrapping species tree
  - Slow: requires
    bootstrapping all genes
    (e.g., 100 times slower)

Statistical support



Erfan Sayyari

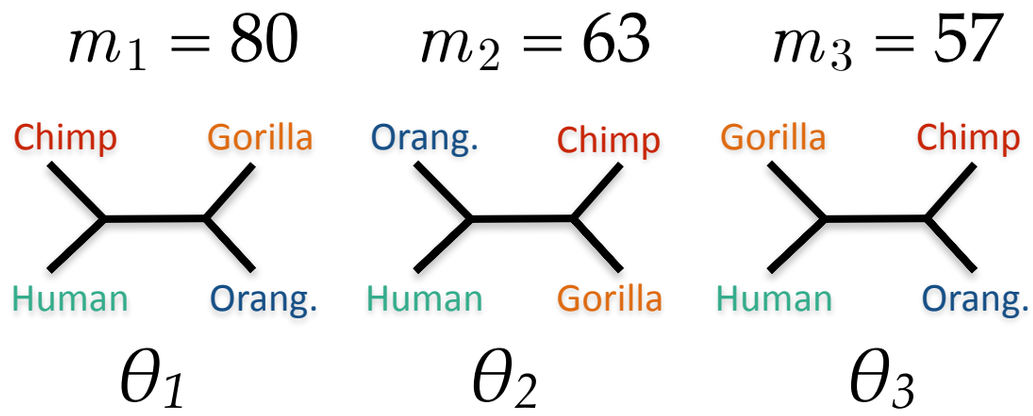
- Traditional approach: bootstrapping each gene, then bootstrapping species tree
  - Slow: requires bootstrapping all genes (e.g., 100 times slower)
  - Inaccurate and hard to interpret
    [Mirarab et al., Sys bio, 2014; Bayzid et al., PLoS One, 2015]



[Mirarab et al., Sys bio, 2014]

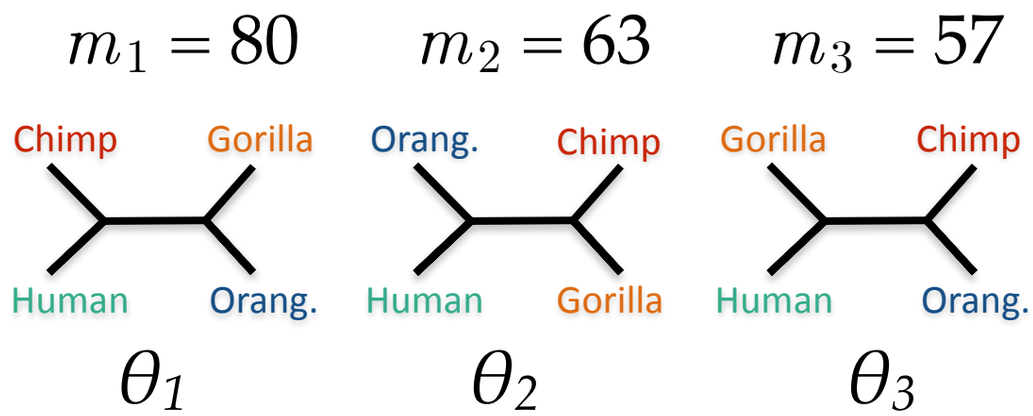
Local posterior probability

- Quartet frequencies follow a multinomial distribution



Local posterior probability

- Quartet frequencies follow a multinomial distribution

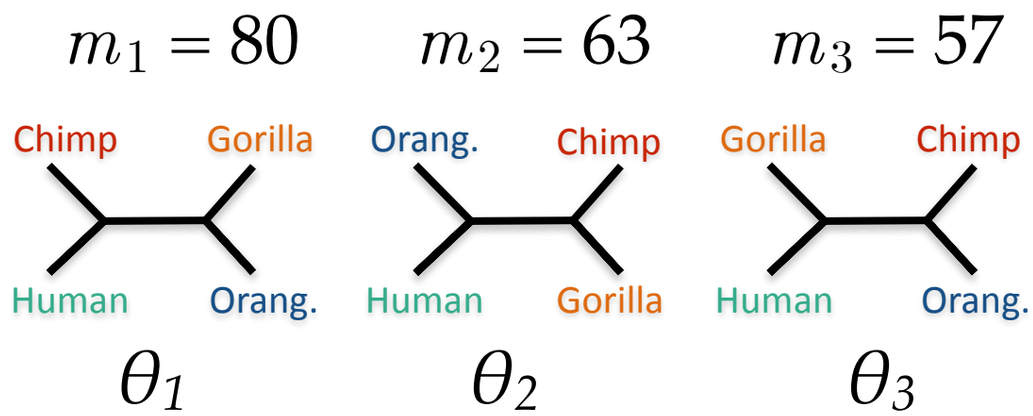


- $\boldsymbol{P}$ (gene tree seen m_1/m times = species tree) $= \boldsymbol{P}(\theta_1 > 1/3)$

- Solved analytically: "local posterior probability" (localPP)

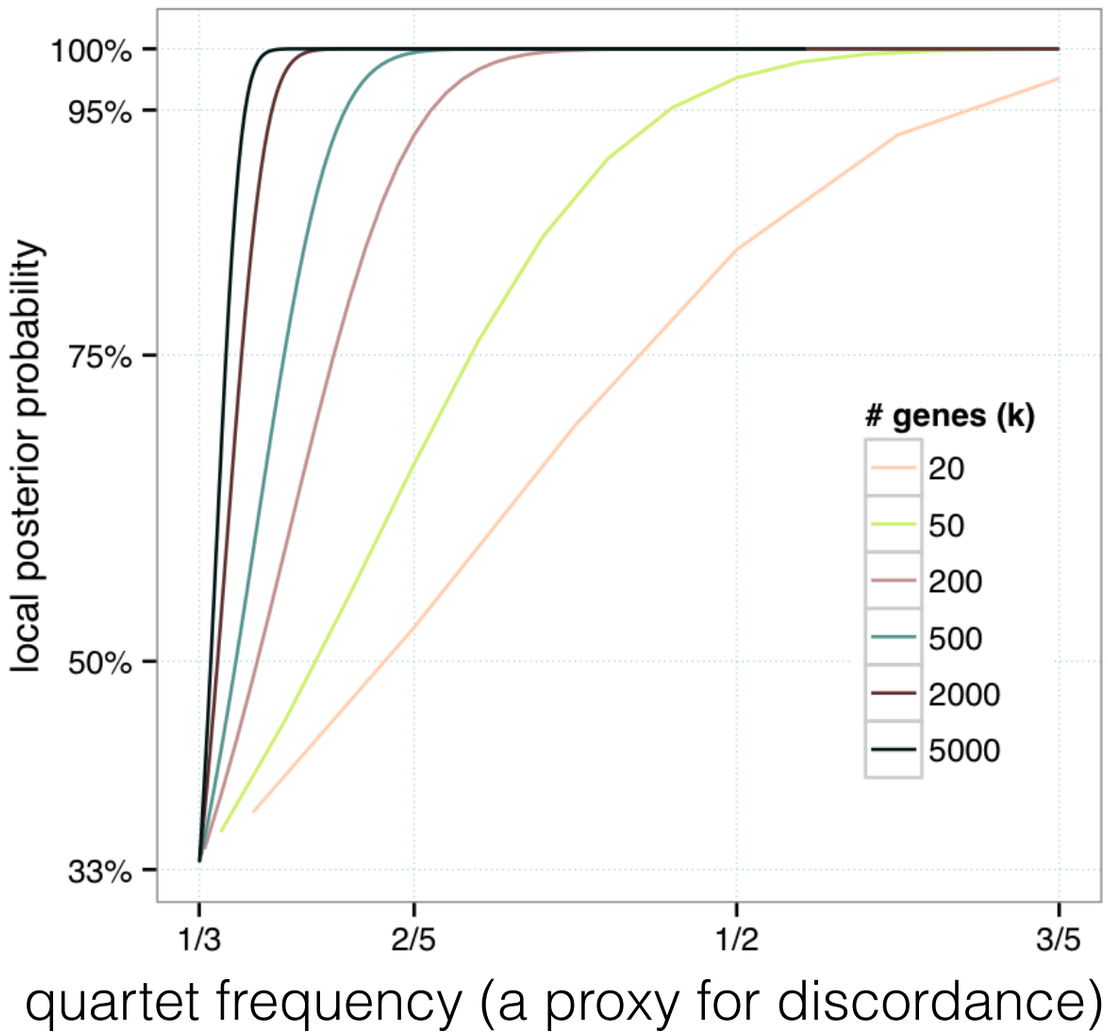
Local posterior probability

- Quartet frequencies follow a multinomial distribution



- $\boldsymbol{P}$ (gene tree seen m_1/m times = species tree) $= \boldsymbol{P}(\theta_1 > 1/3)$
  - Solved analytically: "local posterior probability" (localPP)
- $n > 4$ leads to an exponentially growing number of cases
  - Approximate by using averaged all quartet scores, which can be computed in time quadratic in n.

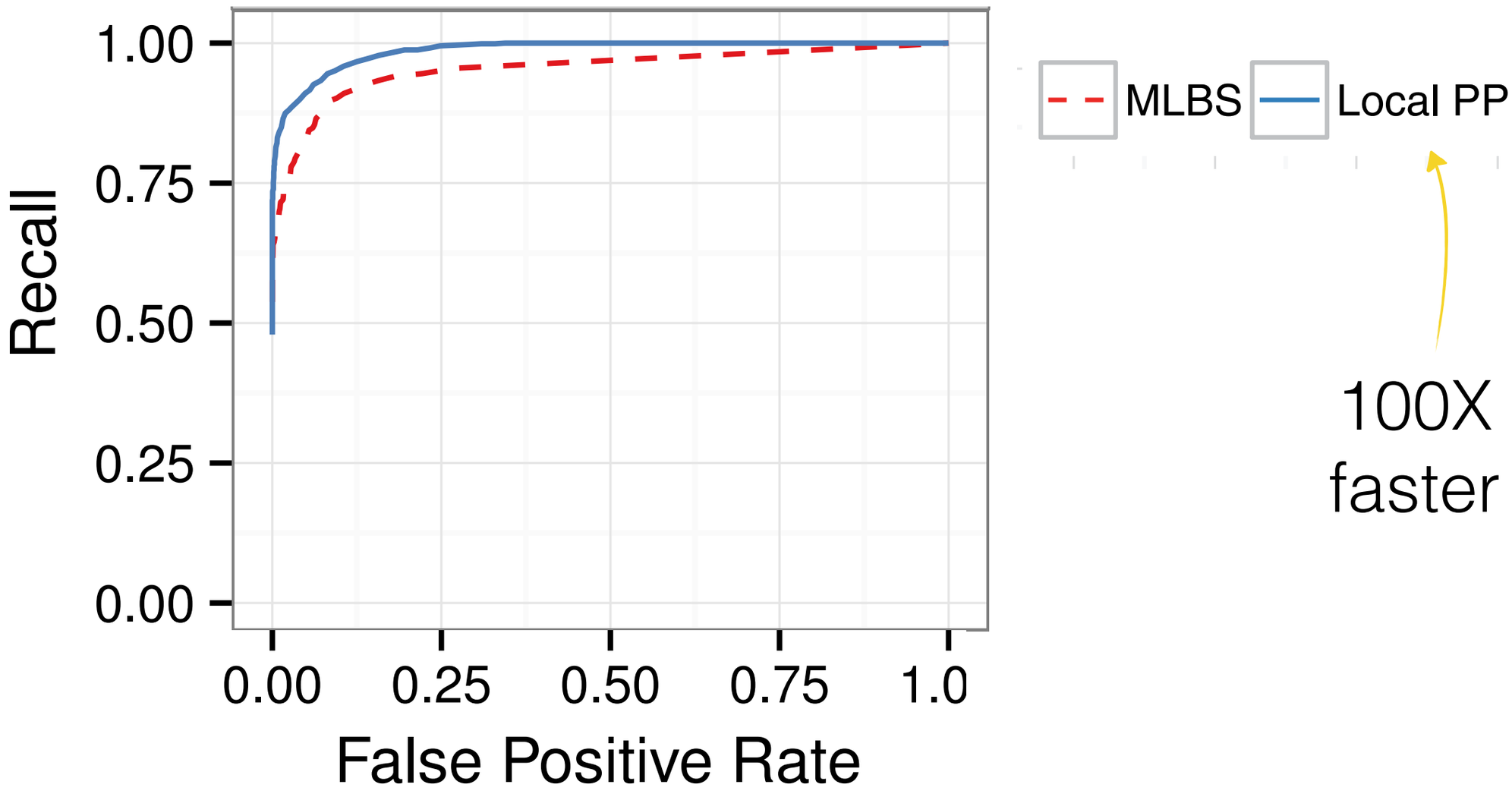
Quartet support vs. localPP



Increased number of genes ⇒ increased support

Increased discordance ⇒ Reduced support

localPP is more accurate than bootstrapping



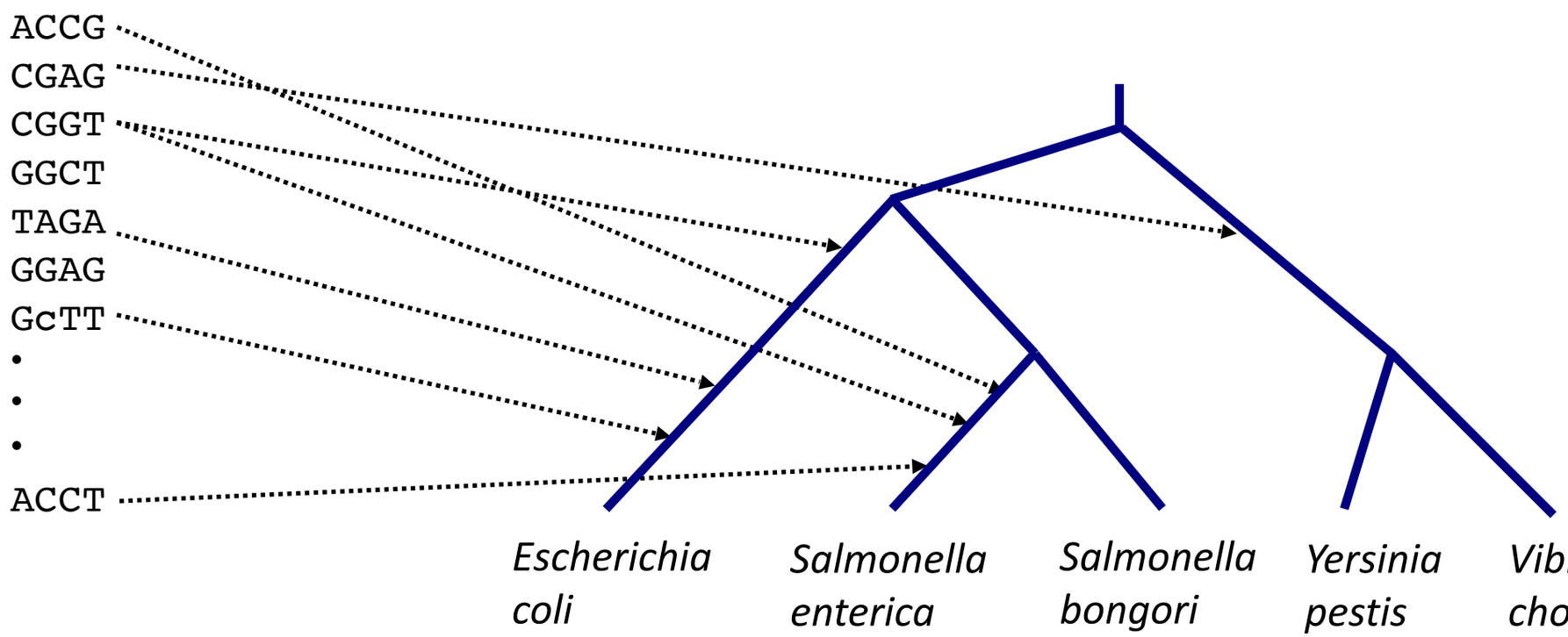
Avian simulated dataset (48 taxa, 1000 genes)

[Sayyari and Mirarab, MBE, 2016]

To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.

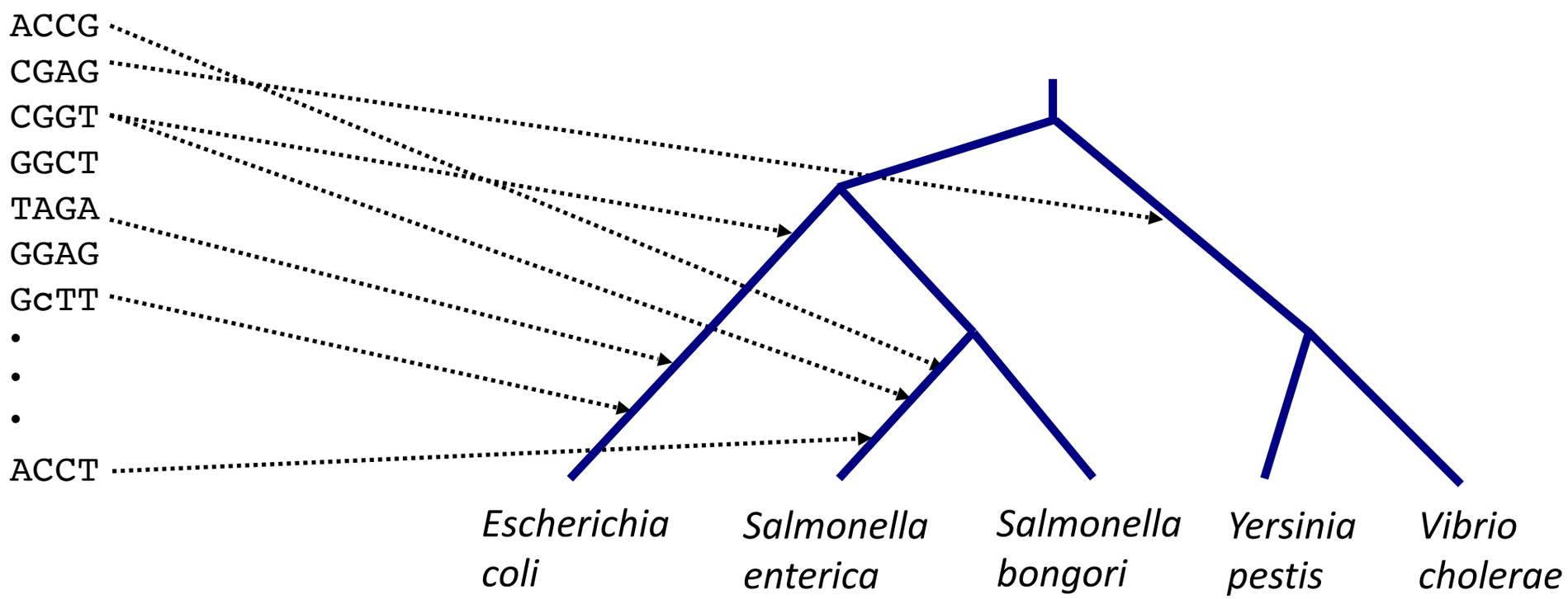
Phylogenetic placement



Query sequences

A backbone dataset

Phylogenetic placement



Applications:

- Place sequences of *unknown* origins on a *reference tree* of known sequences (a major goal in microbiome analyses)
- Update an existing tree quickly without recomputing

Why placement?

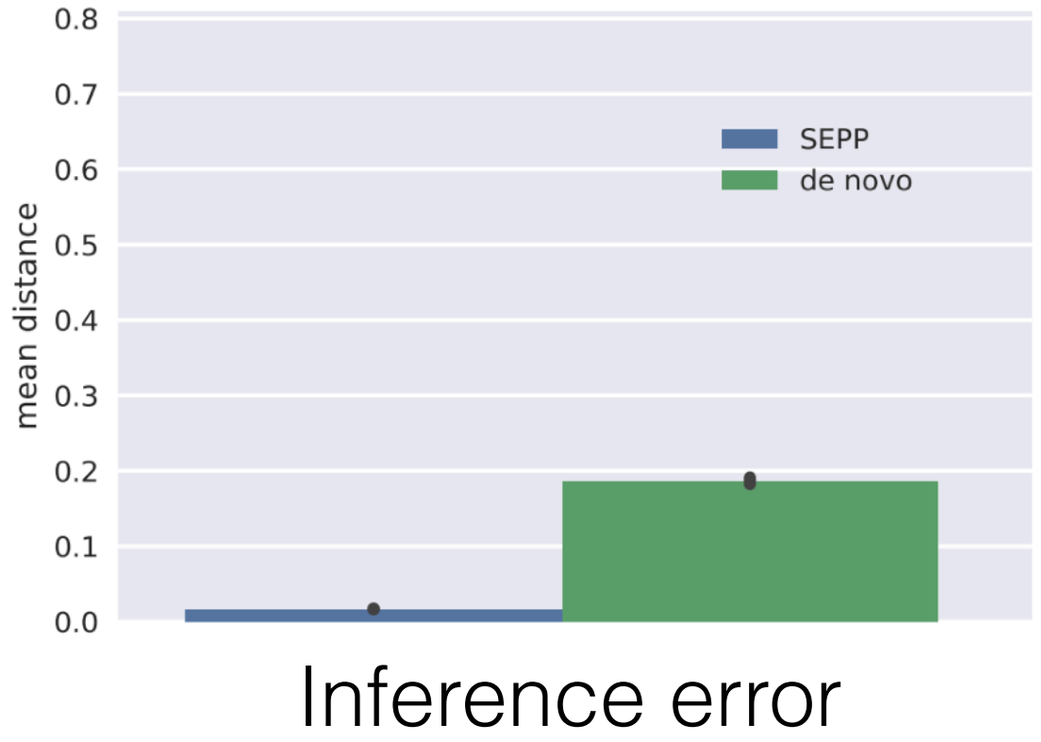
- Placement is easier than *de novo* inference
- Placement is usually sufficient for downstream applications
    - Sometimes, placement is *more accurate*.
- Placement is embarrassingly parallel

SEPP: placement for microbiome data

- Uses divide-and-conquer to align and place on very large backbone trees
- Useful for identifying microbiome data

SEPP: placement for microbiome data

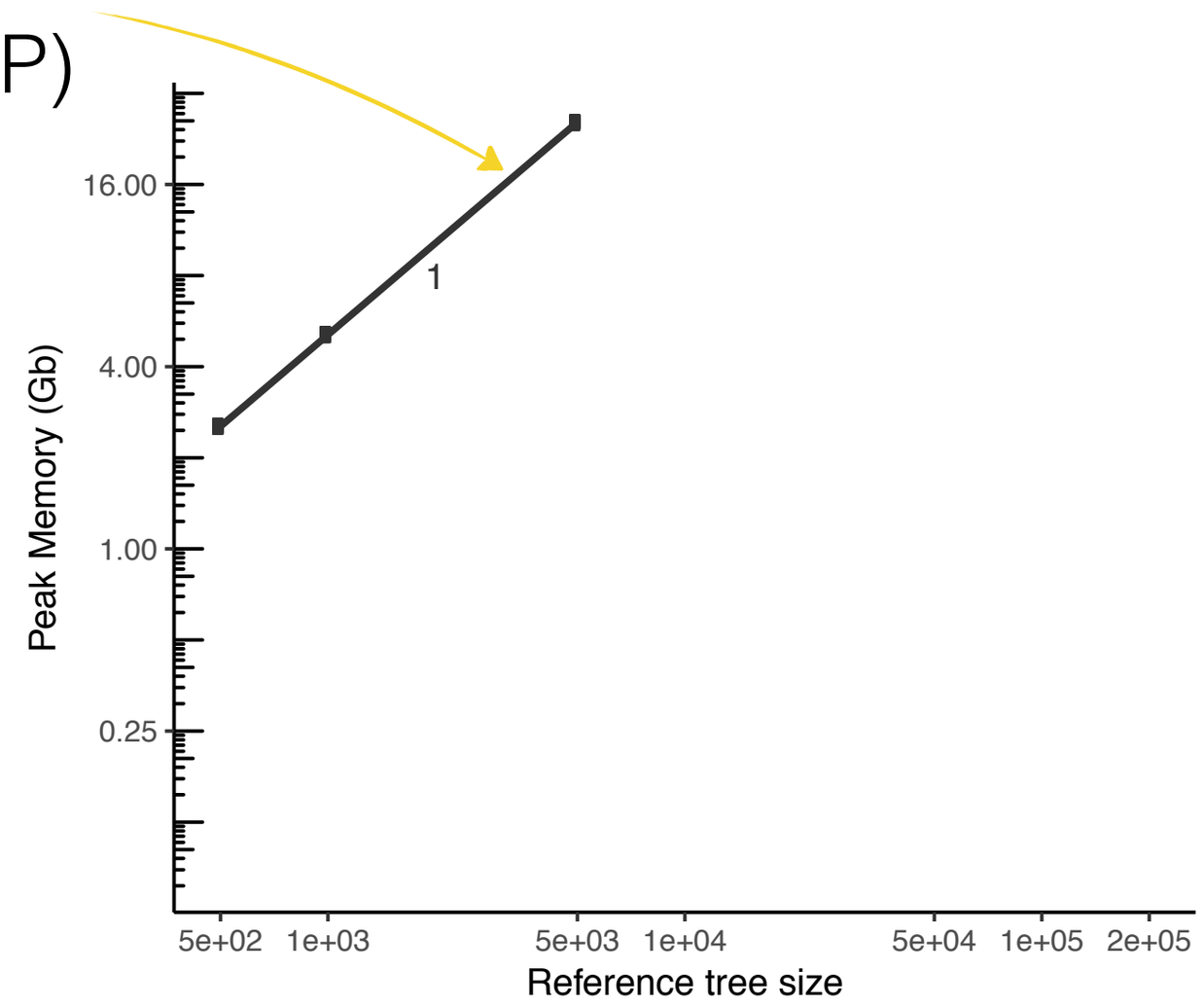
- Uses divide-and-conquer to align and place on very large backbone trees
  - Useful for identifying microbiome data
- Has better accuracy than *de novo* inference of the tree
  - When inferring from fragmentary data



[Janssen, mSystems, 2018]

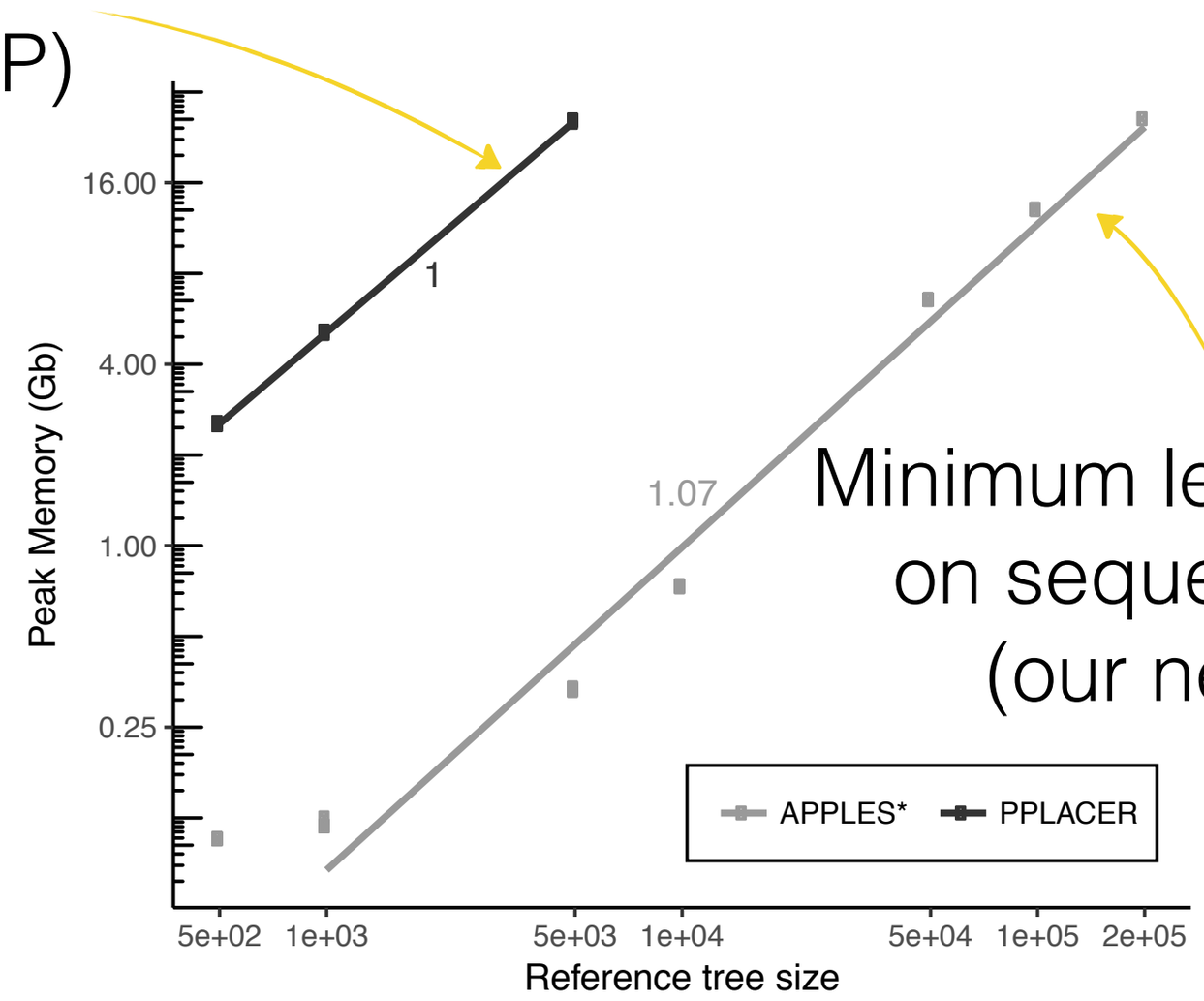
Placement algorithms:
State of the art (ML) is memory-hungry

ML
(used inside SEPP)



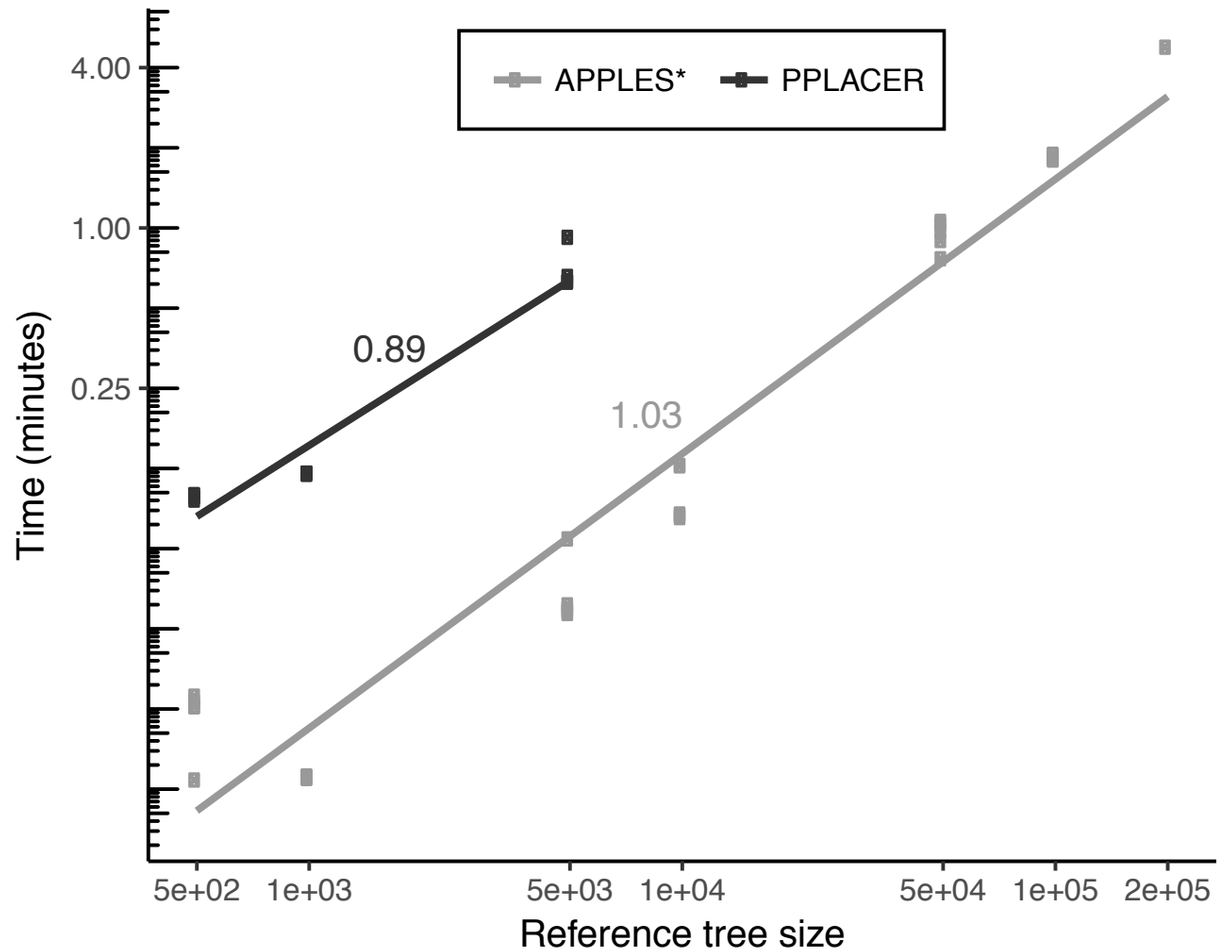
Placement algorithms:
State of the art (ML) is memory-hungry

ML
(used inside SEPP)

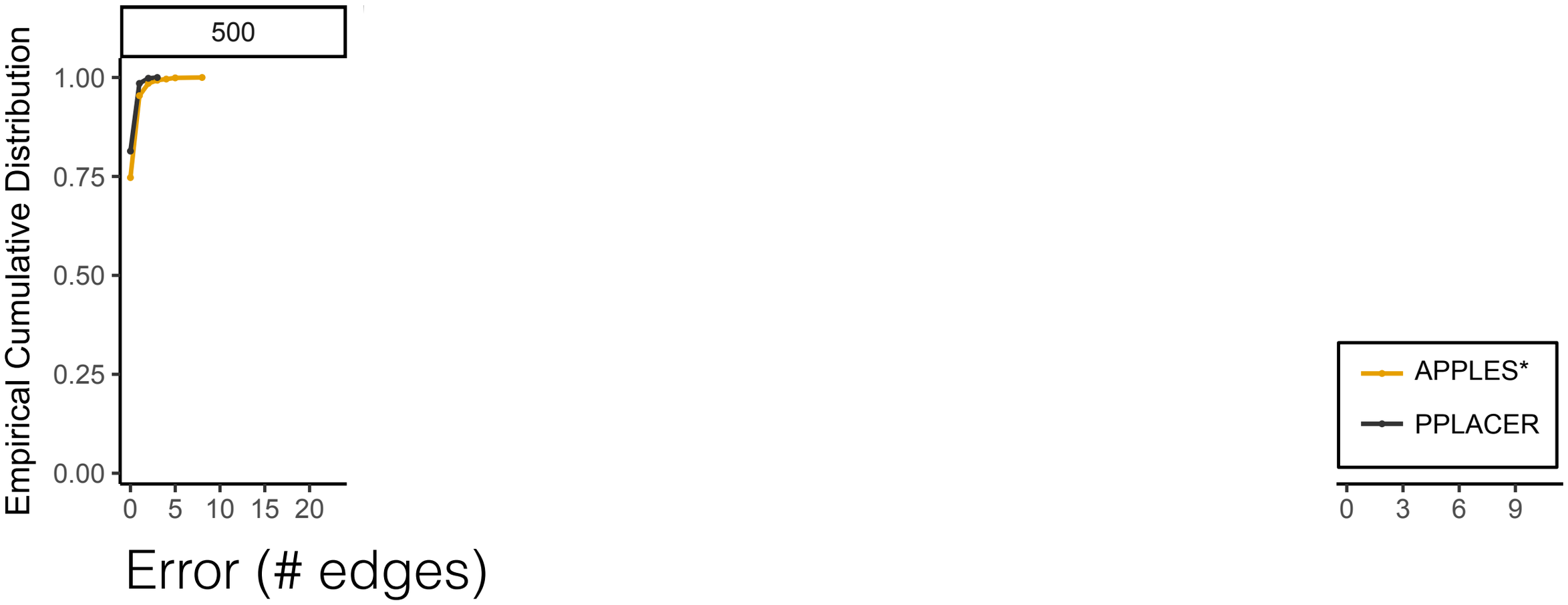


Minimum least square error
on sequence distances
(our new method)

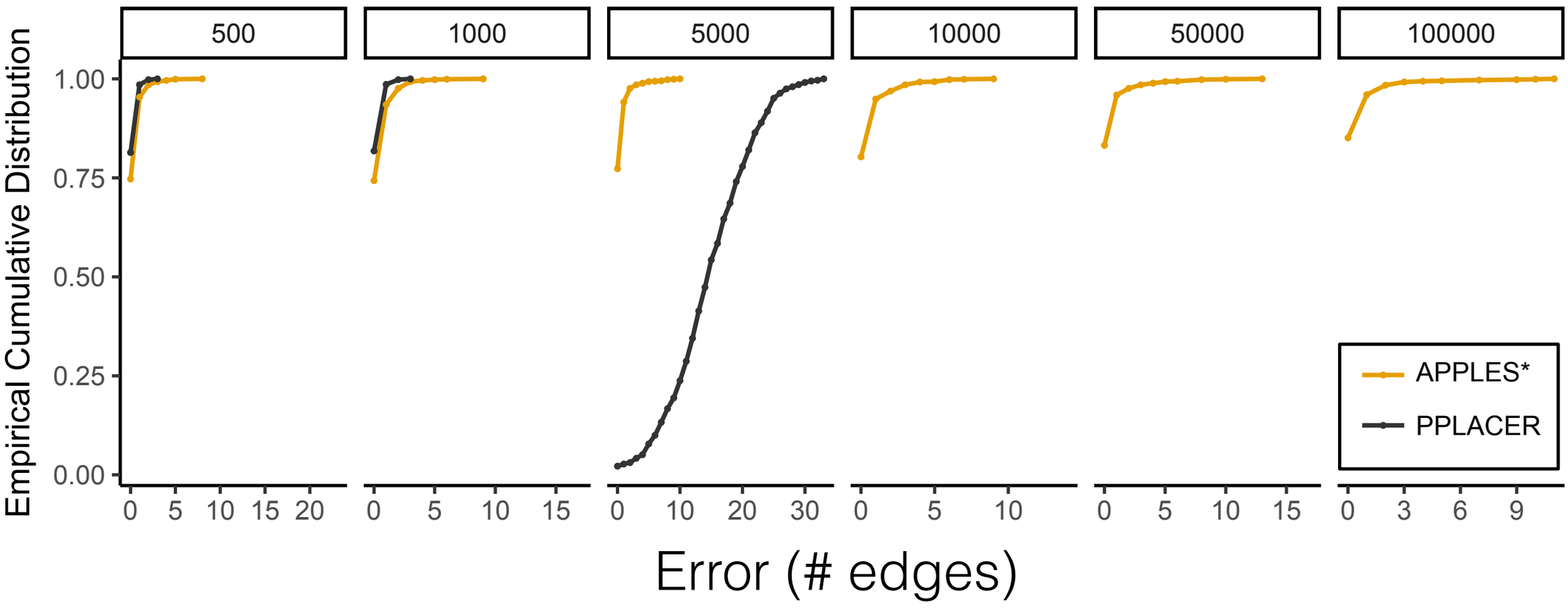
Distance-based method is also much faster than ML



The distance-based method is equally or more accurate



The distance-based method is equally or more accurate



How about placing on species trees?

- Once new data are added to gene trees, we want to update the species tree

How about placing on species trees?

- Once new data are added to gene trees, we want to update the species tree

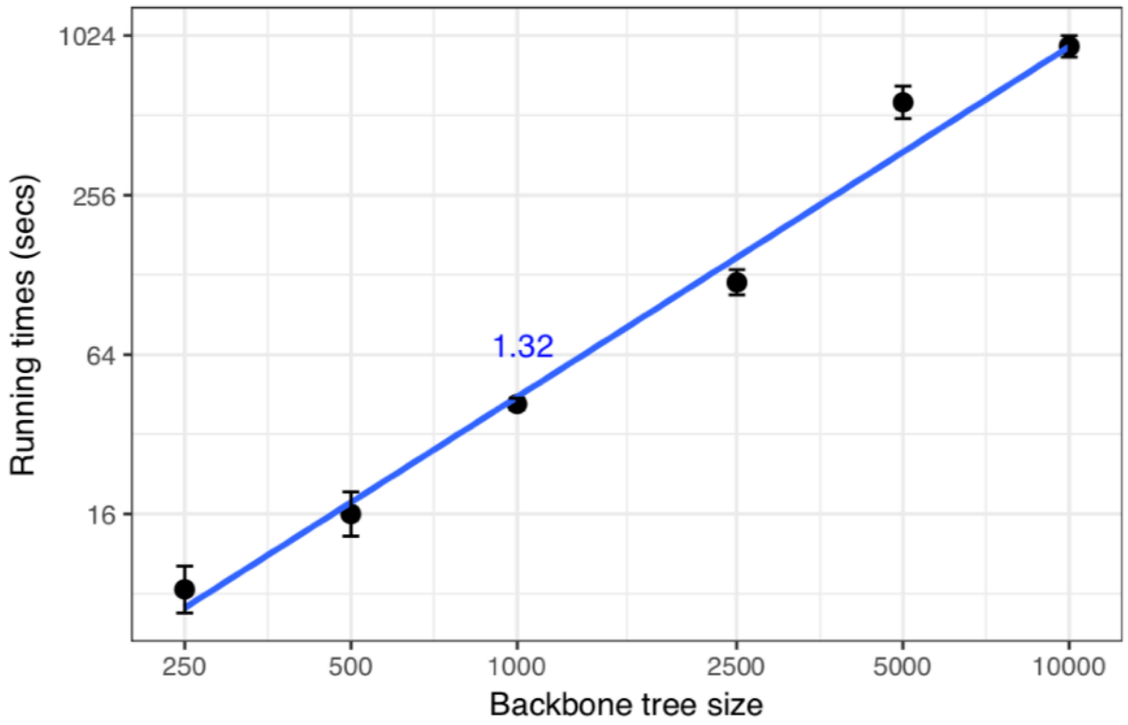
Maryam Rabiee

- INSTRAL: a (worst-case) quadratic-time ASTRAL-like method to update species trees

  - Finds the best place almost always (>99.9%)

How about placing on species trees?

- Once new data are added to gene trees, we want to update the species tree



- INSTRAL: a (worst-case) quadratic-time ASTRAL-like method to update species trees [Rabiee, biorxiv, 2018]
  - Finds the best place almost always (>99.9%)
- Sub-quadratic running time in practice

Examples of concerns with accuracy

But how about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
  - Are used to answer harder problems
  - Allow the use of more complex models
  - Riddled with erroneous data
- Often, methods lose their accuracy on large datasets

Errors abound in phylogenomic datasets

Perspective

On the importance of homology in the age of phylogenomics

Mark S. Springer & John Gatesy

Pages 210-228 | Received 10 Jul 2017, Accepted 25 Oct 2017, Published online: 08 Dec 2017

Phylogenetic analysis at deep timescales: Unreliable gene trees, bypassed hidden support, and the coalescence/concatalescence conundrum

John Gatesy *, Mark S. Springer

Resolving Difficult Phylogenetic Questions: Why More Sequences Are Not Enough

Hervé Philippe, Henner Brinkmann, Dennis V. Lavrov, D. Timothy J. Littlewood, Michael Manuel, Gert Wörheide, Denis Baurain

Error, signal, and the placement of Ctenophora sister to all other animals

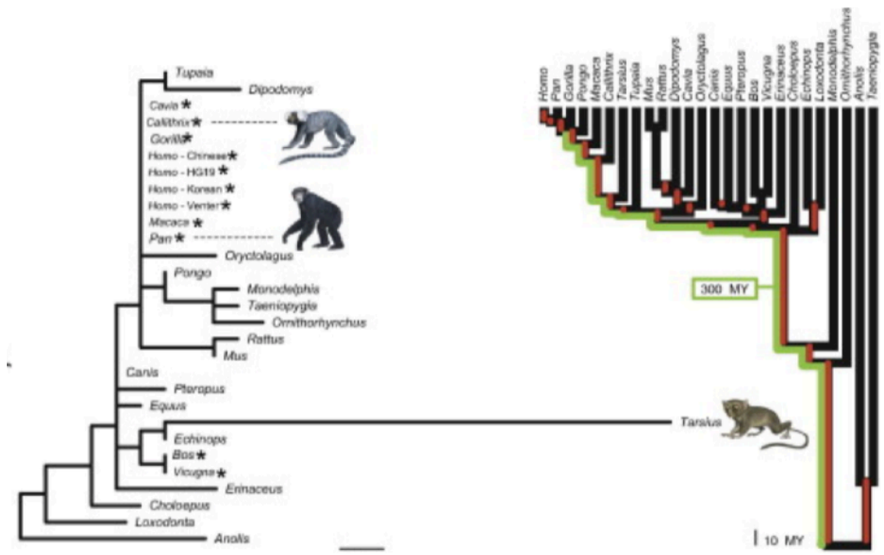
Nathan V. Whelan, Kevin M. Kocot, Leonid L. Moroz, and Kenneth M. Halanych

PNAS published ahead of print April 20, 2015 https://doi.org/10.1073/pnas.1503453112

The gene tree delusion ☆

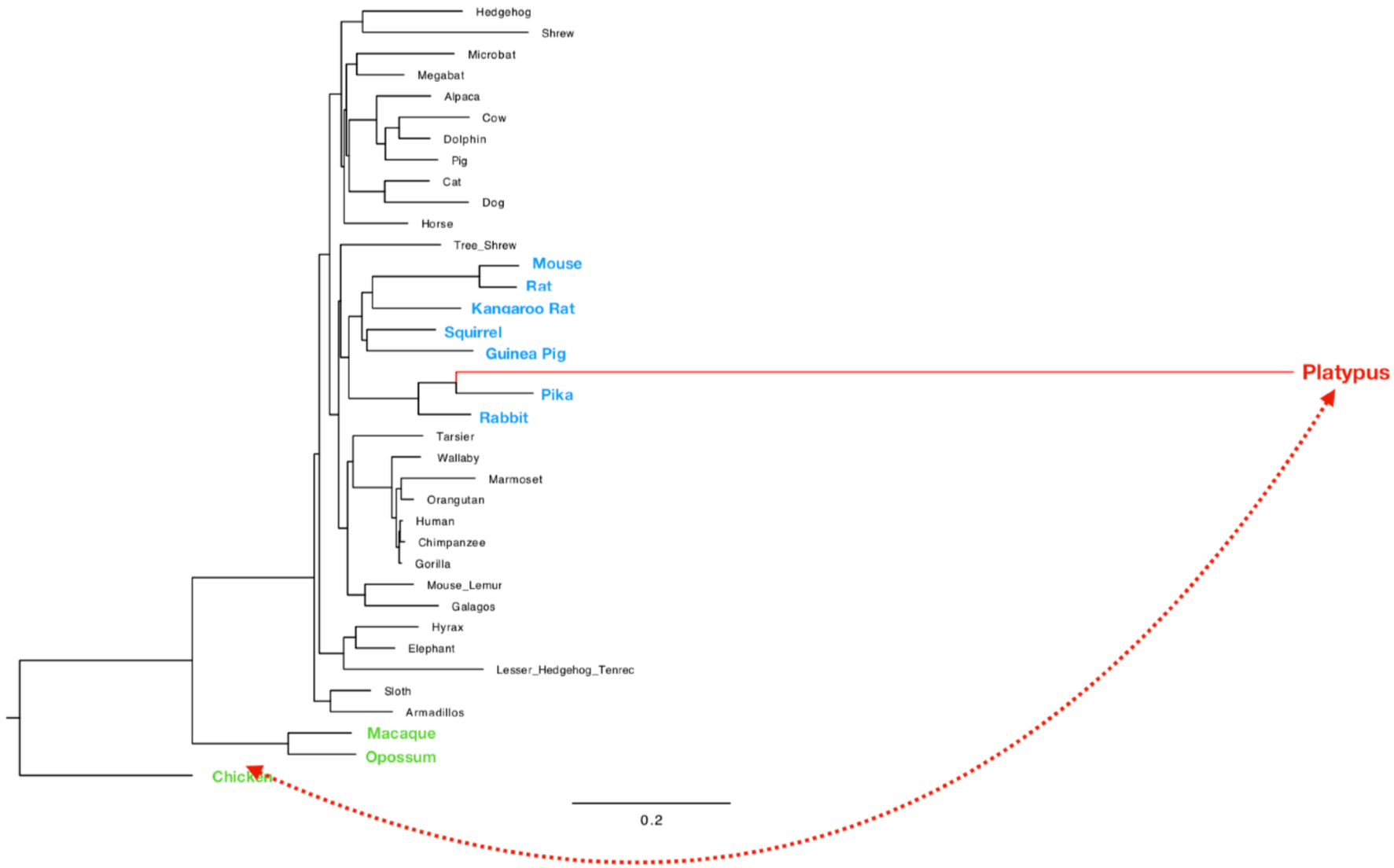
Mark S. Springer *, John Gatesy *

Errors show as long branches



Gatesy et. al. (2014)

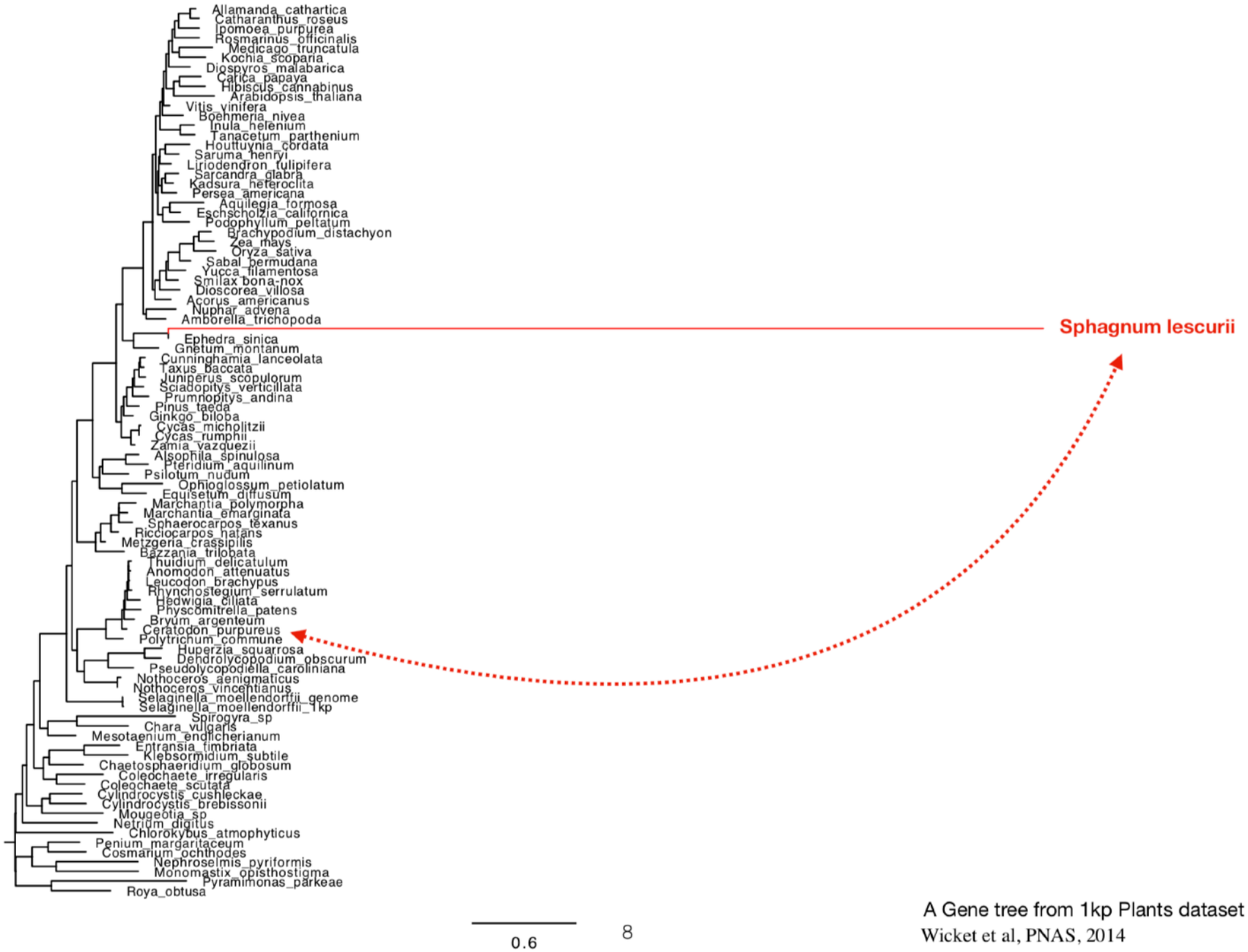
Errors show as long branches



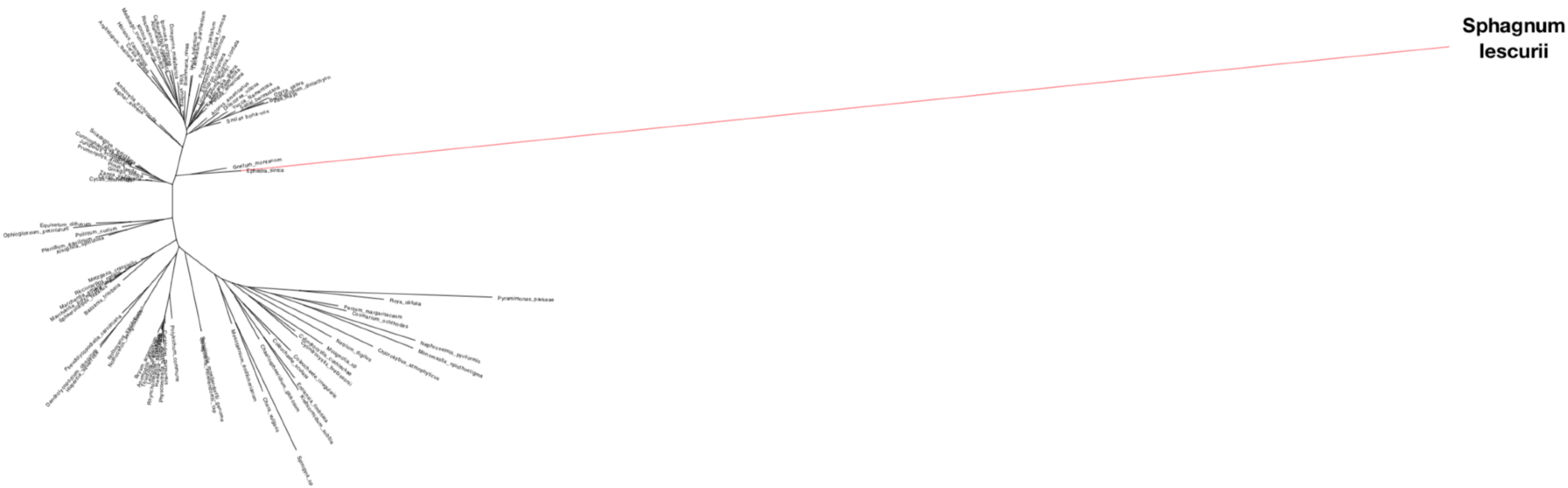
A Gene tree from Mammalian dataset

Song et al, PNAS, 2012

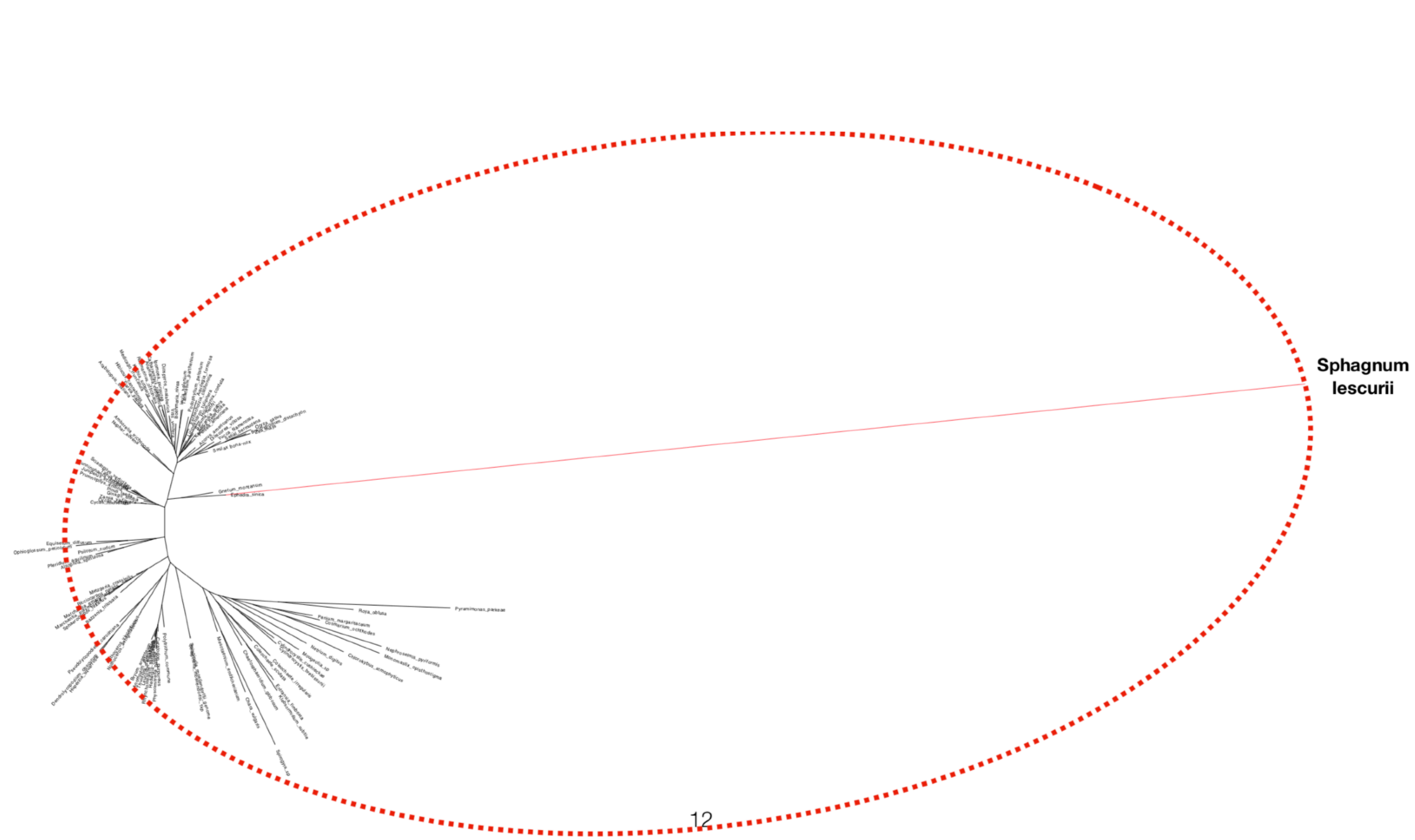
Errors show as long branches

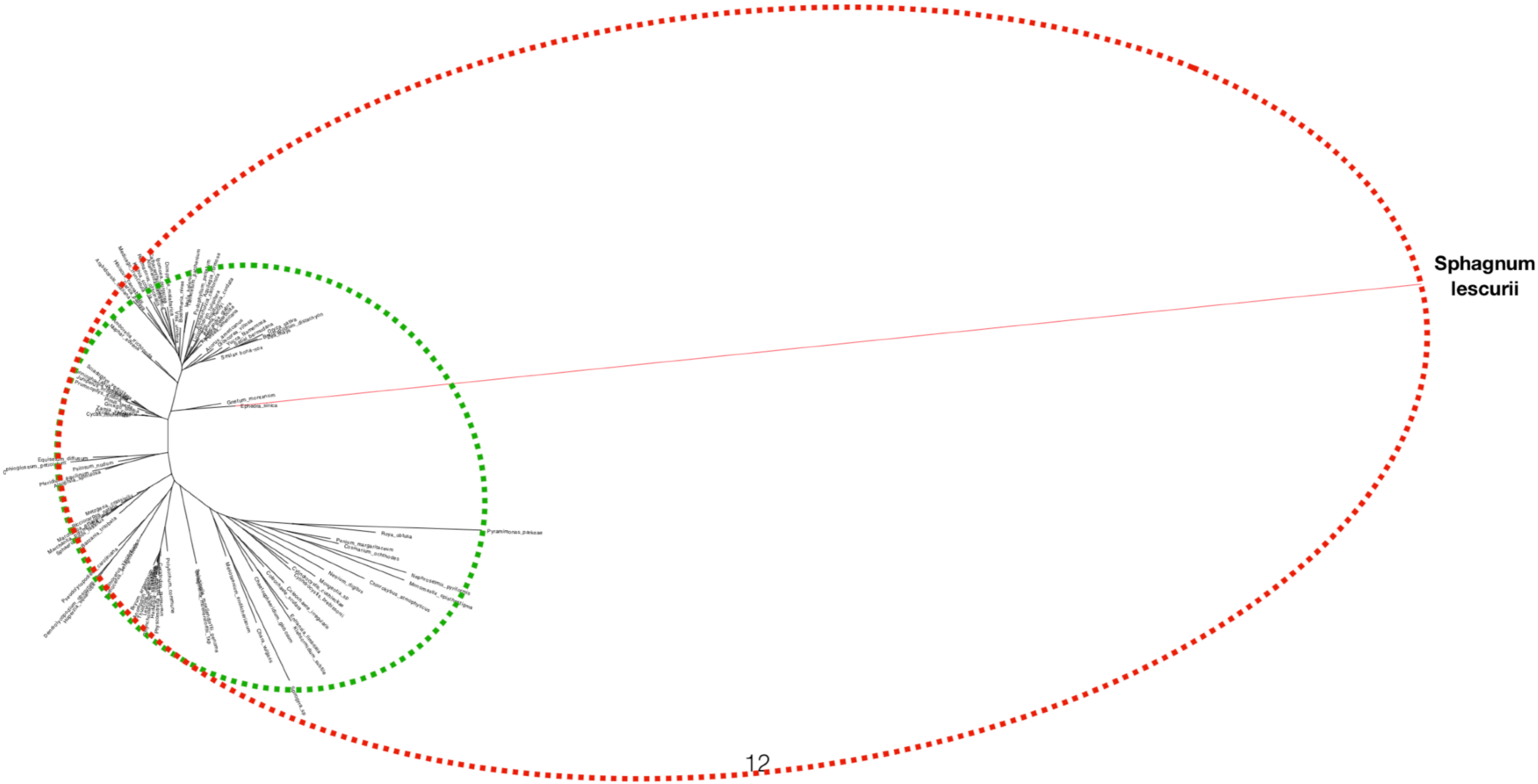


A Gene tree from 1kp Plants dataset
Wicket et al, PNAS, 2014



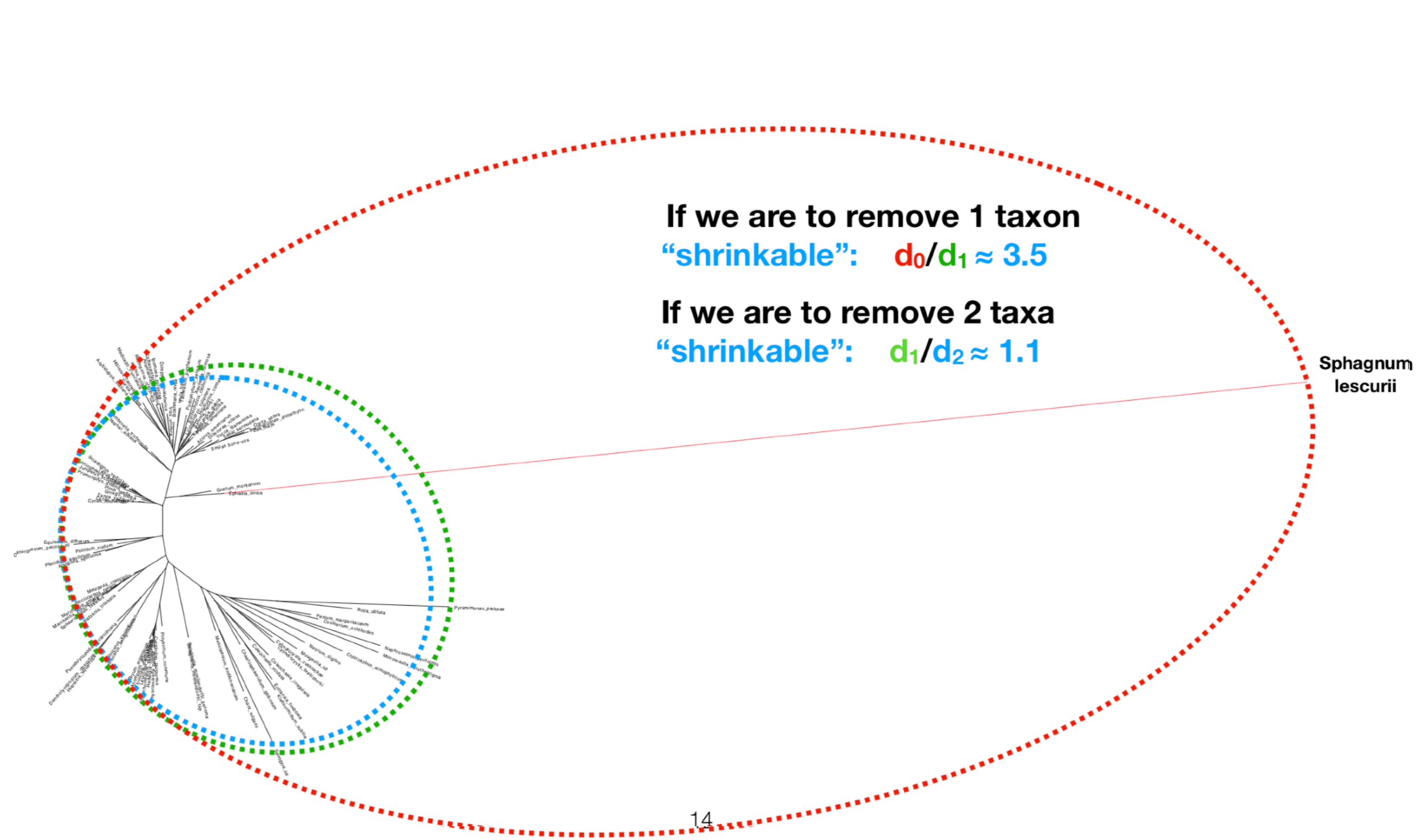
Sphagnum lescurii
Pyramimonas_parkeae
Roya_obtusa
Penium_margaritaceum
Gnetum_montanum
Ephedra_sinica
Psilotum_nudum
Chlorokybus_atmophyticus


Sphagnum lescurii
Gnetum_montanum
Ephedra_sinica
Equisetum_diffusum
Ophioglossum_petiolatum
Psilotum_nudum
Pteridium_aquilinum
Alsophila_spinulosa
Mnium_cuspidatum
Marchantia_polymorpha
Marchantia_paleacea
Sphaerocarpos_texanus
Blasia_pusilla
Pseudolycopodiella_caroliniana
Dendrolycopodium_obscurum
Huperzia_selago
Polytrichum_commune
Sphagnum_fimbriatum
Sphagnum_palustre
Sphagnum_girgensohnii
Sphagnum_subsecundum
Sphagnum_squarrosum
Sphagnum_capillifolium
Sphagnum_fallax
Sphagnum_cuspidatum
Sphagnum_riparium
Sphagnum_angustifolium
Sphagnum_fuscum
Sphagnum_magellanicum
Sphagnum_papillosum
Sphagnum_rubellum
Sphagnum_russowii
Sphagnum_tenellum
Sphagnum_warnstorfianum
Sphagnum_sp
Mesostigma_viride
Chara_vulgaris
Coleochaete_irregularis
Coleochaete_scutata
Cylindrocystis_brebissonii
Closterium_peracerosum
Cosmarium_botrytis
Klebsormidium_subtile
Mougeotia_sp
Netrium_digitus
Penium_margaritaceum
Cosmarium_pothydes
Chlorokybus_atmophyticus
Nephroselmis_pyriformis
Monomastix_opisthostigma
Pyramimonas_parkeae
Roya_obtusa
12


Sphagnum
lescurii
Pyramimonas_parkeae

If we are to remove 1 taxon
"shrinkable": $d_0/d_1 \approx 3.5$

If we are to remove 2 taxa
"shrinkable": $d_1/d_2 \approx 1.1$



An optimization problem

The *k*-shrink problem:

- Given:
  - a tree with n leaves and branch lengths
  - some $1 \leq k \leq n$

- Find:
  - for every $1 \leq i \leq k$:
    - the set of i leaves that should be removed to reduce the tree diameter maximally

An optimization problem

The *k*

- Giv
  - a
  - s
- Find
  - fo
    -

s

oved

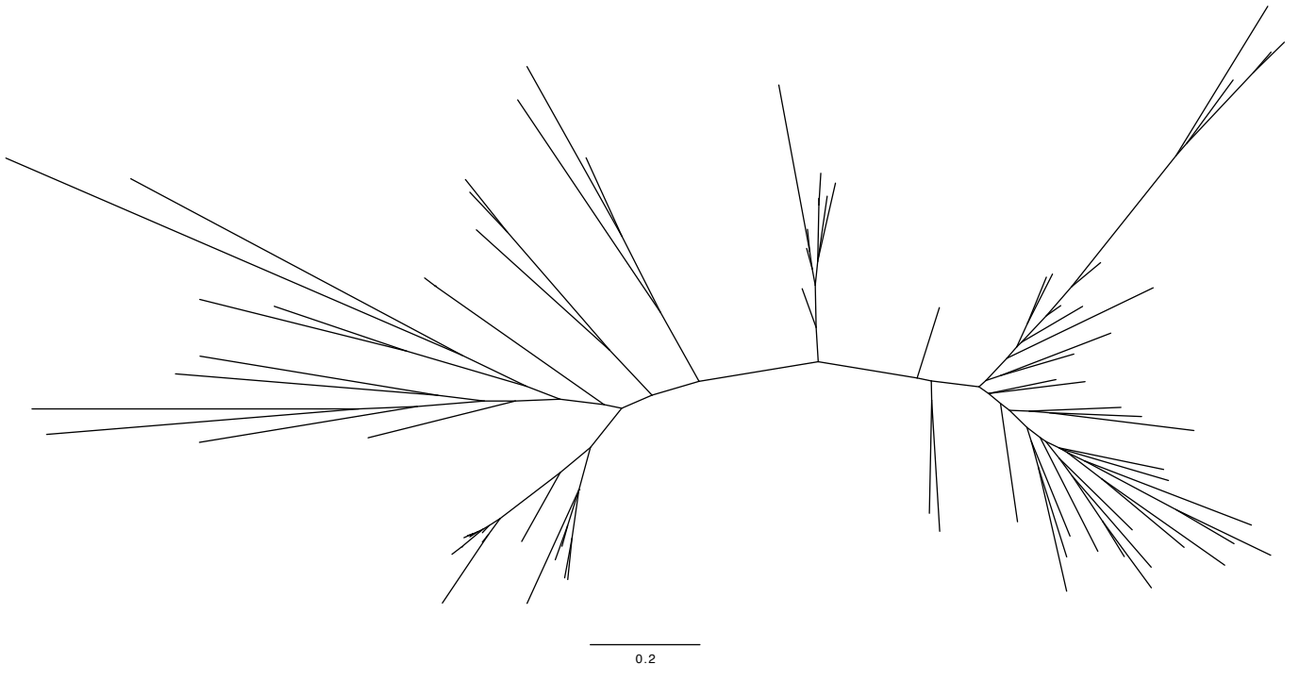
y

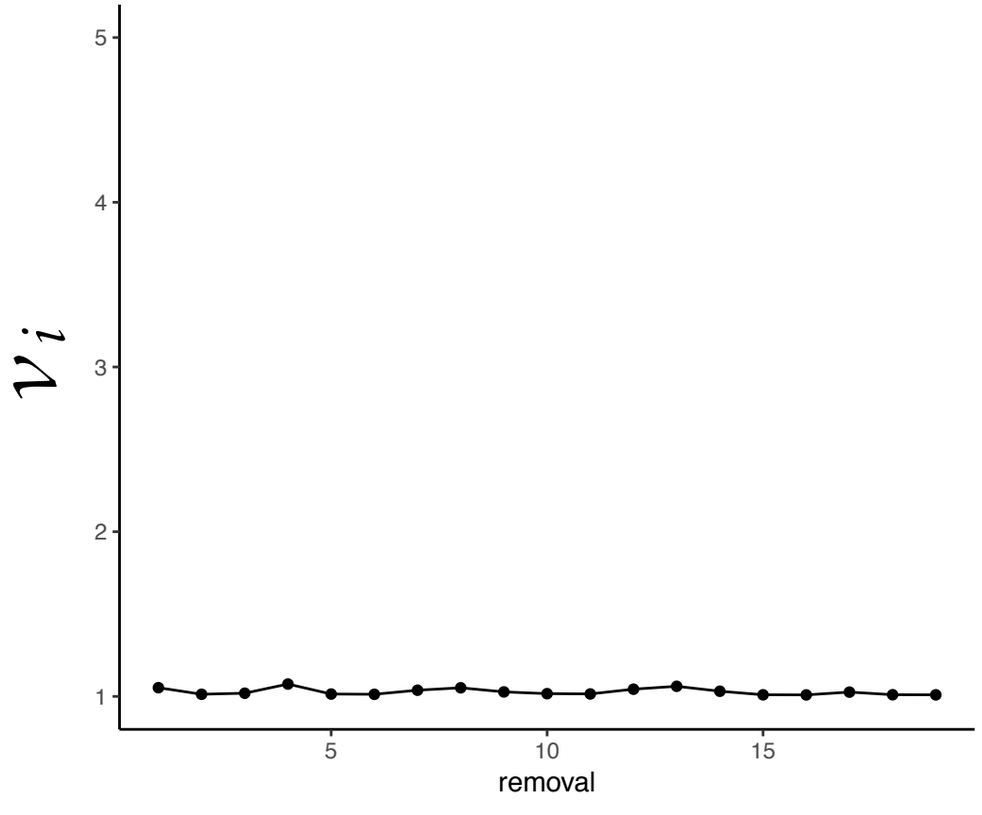
We have a polynomial time solution using dynamic programming

Uyen Mai

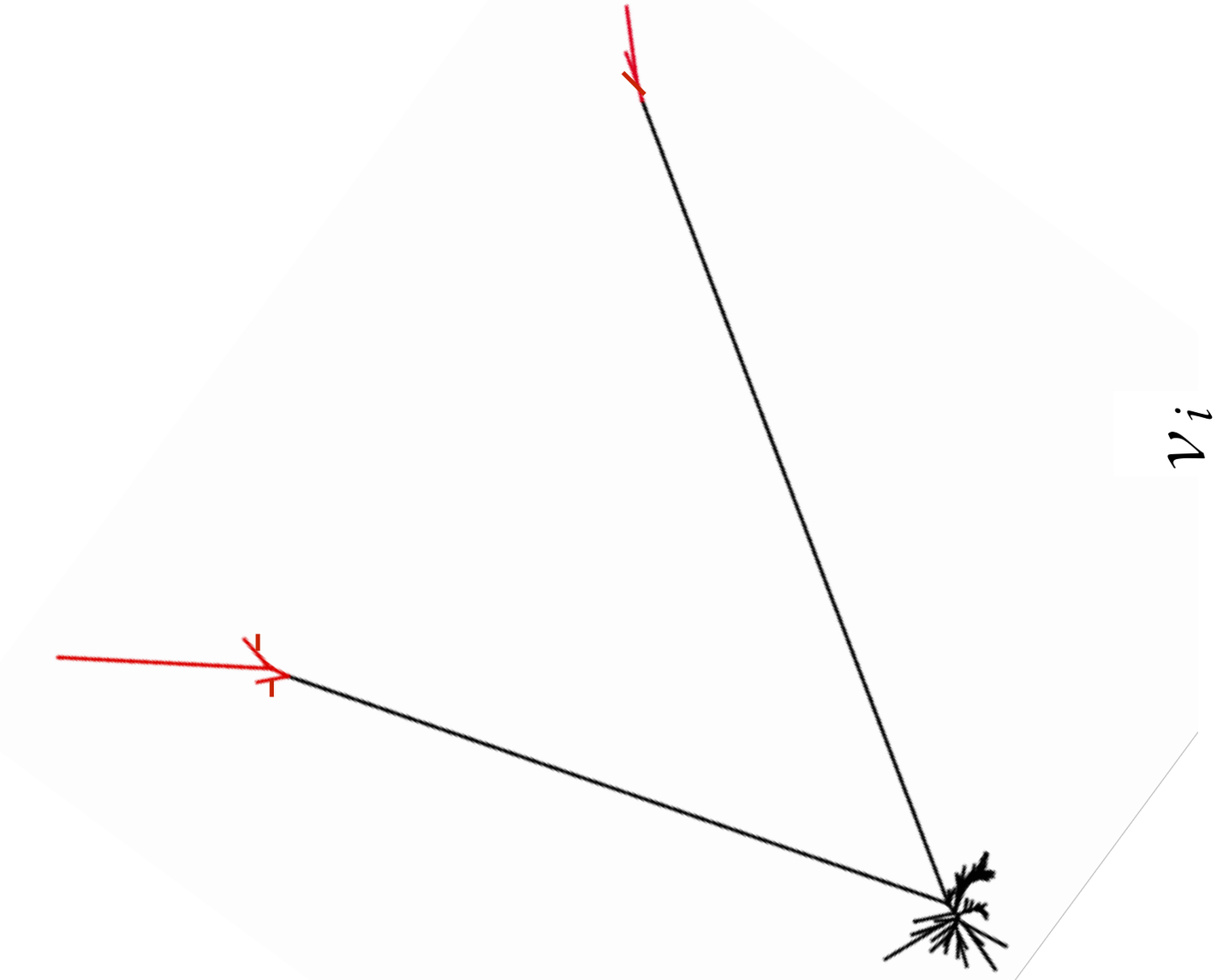
What to remove?

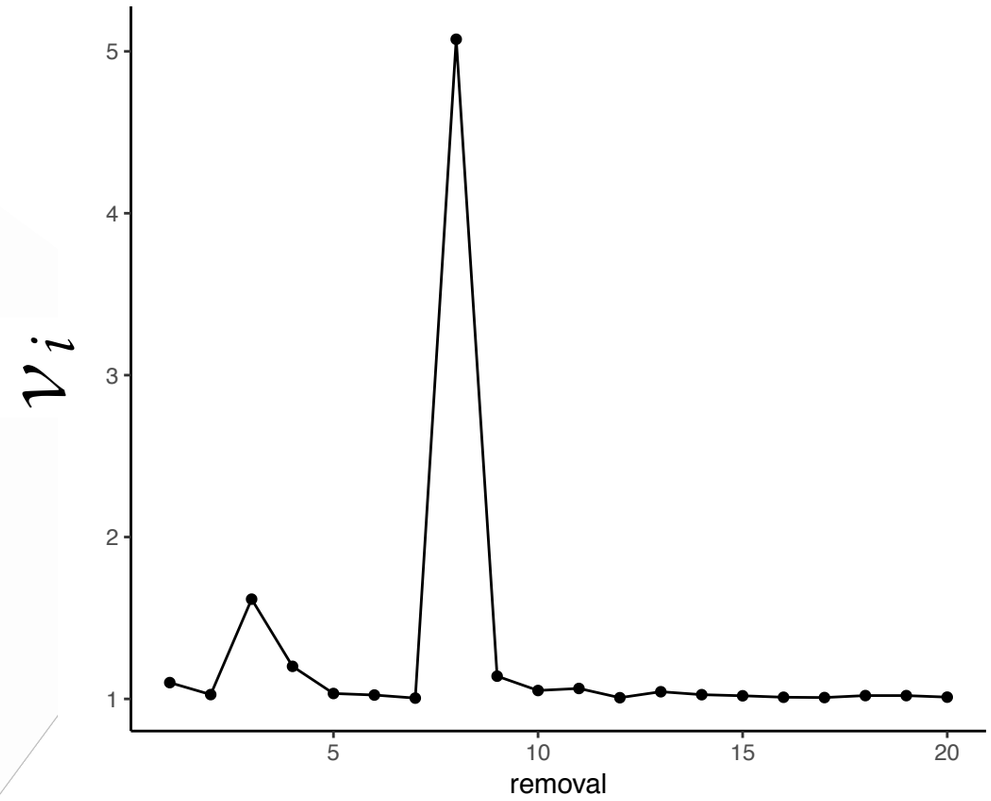
Let v_i = $\frac{\text{the diameter after } i\text{-}1 \text{ removals}}{\text{the diameter after } i \text{ removals}}$





What to remove?





Running Time

- k-shrink can be solved in $O(k^2h+n)$

  where h = the tree height

  - by default, we set $k=O(n^{0.5})$

- Fast: processes a tree of n=203,452 leaves with k=2255 in 28 mins

How much data?

- In biology, data is not free. Fundamental questions:
  - How much data is enough?
  - What's the best strategy to obtain data?

How much data?

Sebastien Roch

Shubhanshu Shekhar

- In biology, data is not free. Fundamental questions:
  - How much data is enough?
  - What's the best strategy to obtain data?
- Theoretical answers: sample complexity
  - For ASTRAL, we have bounds on the number of genes needed

[Shekhar et al, TCBB, 2017]

How much data?

Sebastien Roch

Shubhanshu Shekhar

- In biology, data is not free. Fundamental questions:
  - How much data is enough?
  - What's the best strategy to obtain data?
- Theoretical answers: sample complexity
  - For ASTRAL, we have bounds on the number of genes needed

[Shekhar et al, TCBB, 2017]

- Practical:
  - For ASTRAL, more genes are better than more or individuals

[Rabiee and Mirarab, MPE, 2018]

  - For phylogenetic placement, we showed a very low coverage of genome (too little for assembly) is more than enough

[Sarmashghi, et. al., biorxiv, 2018]

Code Optimization

- All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)
- What's the best measure of performance?
  - FLOPS?

Code Optimization

- All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

- What's the best measure of performance?
  - FLOPS?
  - POPS ~ PPPD?

Code Optimization

- All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

- What's the best measure of performance?

  - FLOPS?

  - POPS ~ PPPD?

    - Programs Optimized Per Student ~ Papers Published Per Dissertation

Code Optimization

- All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

- What's the best measure of performance?
  - FLOPS?
  - POPS ~ PPPD?
    - Programs Optimized Per Student ~ Papers Published Per Dissertation
  - Ease of development matters in fast moving fields

More generally …

- Analyzing large datasets requires method development:

- Scalability:
  hardware + software + better algorithms

- Attention to accuracy and how properties of big data (e.g. data noisiness) affect accuracy

Chao Zhang

Uyen Mai

Tandy Warnow

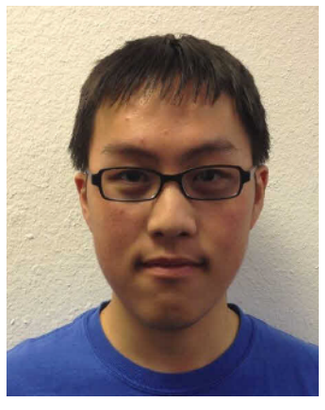
Sebastien Roch

James B. Whitfield

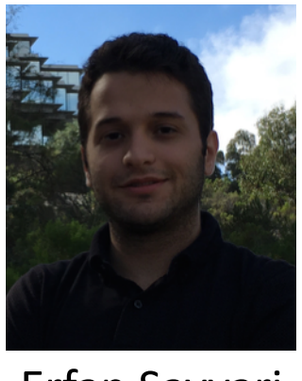
S.M. Bayzid

Théo Zimmermann

Maryam Rabiee



John Yin



Erfan Sayyari

Shubhanshu Shekhar