On the inference of complex phylogenetic networks with SnappNet

Charles-Elie Rabier

Vincent Berry, Marnus Stoltz
João D. Santos, Jean-Christophe Glaszmann
Fabio Pardi and Céline Scornavacca

Genome Harvest / KIM Data & Life Sciences
ISEM, Institut des Sciences de l'Evolution de Montpellier
IMAG, Institut Montpelliérain Alexander Grothendieck
LIRMM, Laboratoire d'informatique, de Robotique et de Microélectronique
UMR AGAP, Amélioration Génétique et adaptation des plantes, CIRAD

Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

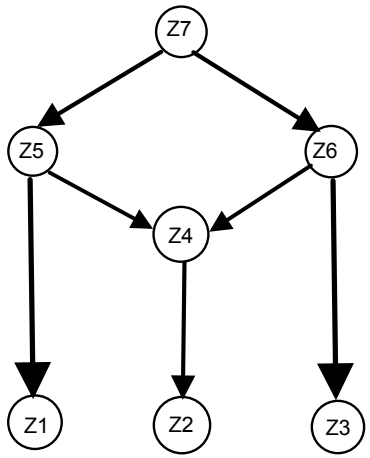
Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

Phylogenetic networks

Phylogenetic networks are Directed Acyclic Graphs (DAG) that allow us to detect :

- hybridizations (e.g. plants)
- introgressions (e.g. plants and animals)
- horizontal gene transfer (e.g. bacteria)

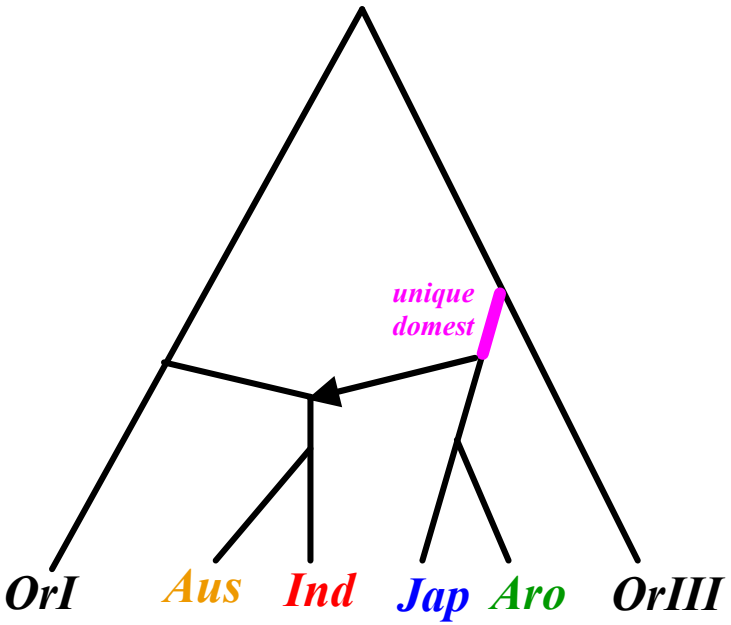


Key points :

- Edge length = evolutionary time
- Dependencies between nodes
- Reticulation nodes have 2 parents and represent reticulation events
- Our goal is to obtain a distribution of phylogenetic networks (uncertainty on clades)

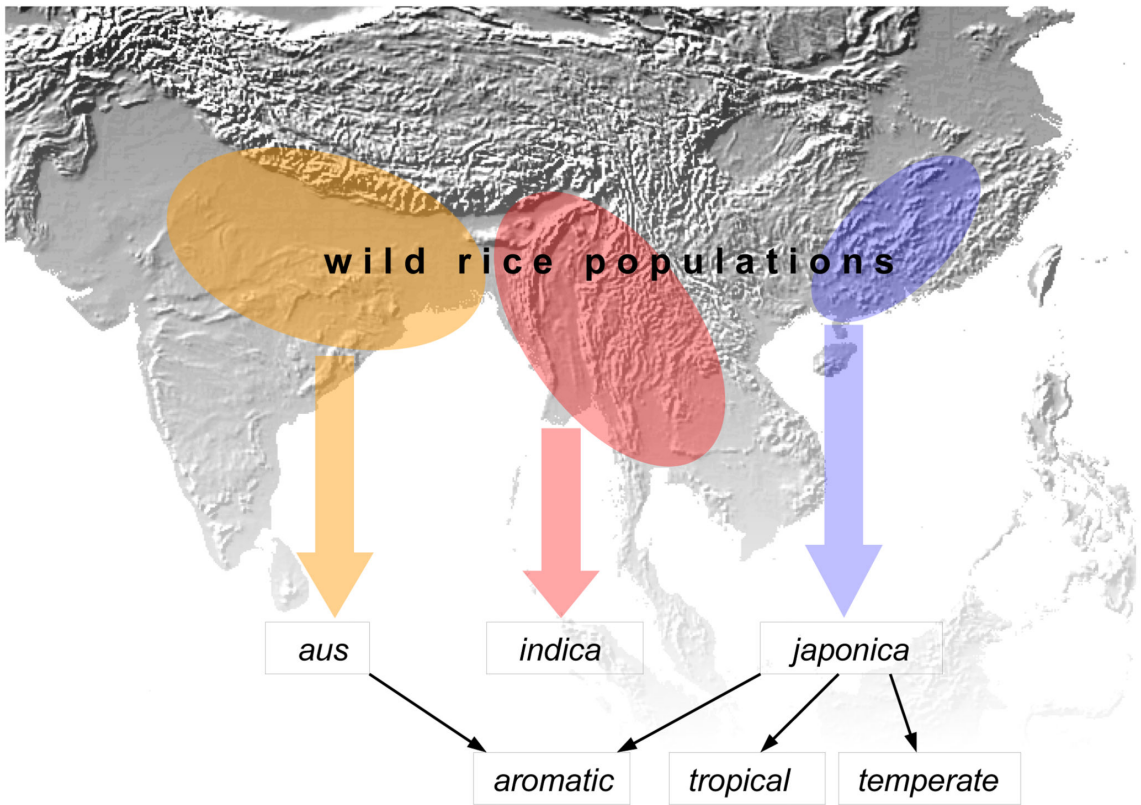
A few scenarios on the rice domestication process

- Huang et al. (Nature, 2012) : japonica domesticated from a wild form in the south of China, and crossed to a wild form in the south East of Asia, generating indica



A few scenarios on the rice domestication process

- Civan et al. (Nature Plants, 2015) : indica, japonica and aus domesticated separately in different locations in Asia

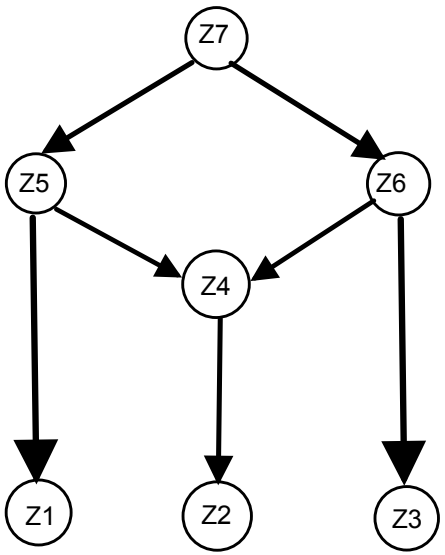


Our methodological approach

We consider a model, that handles incomplete lineage sorting, and that considers explicitely mutations and hybridization.

$\Rightarrow$ Phylogenetic network inference in a rich Bayesian framework

SNAPPNET = Generalization of SNAPP (Bryant et al. 2012) to networks



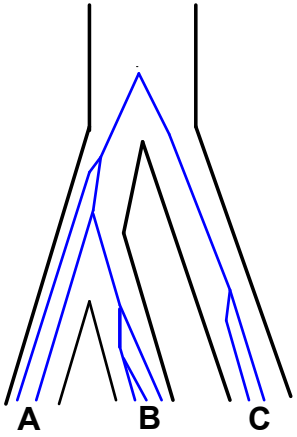
Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

SNAPP software for the Bayesian inference of species trees (Bryant et al. 2012, MBE)

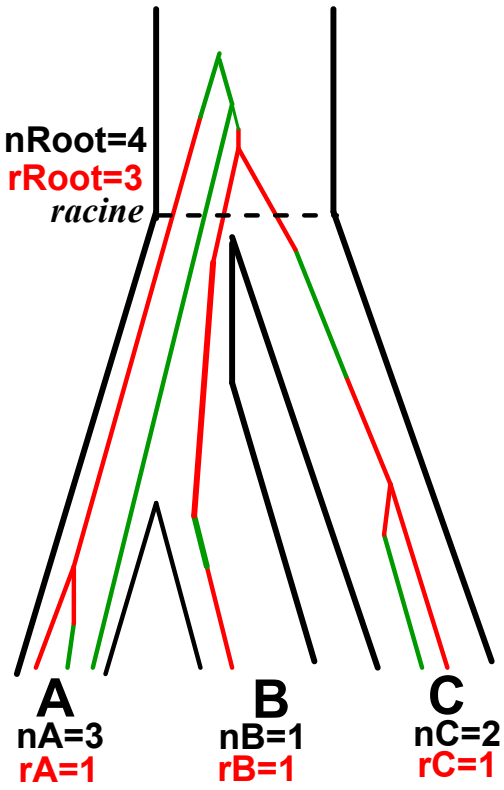
- Independent biallelic markers (SNPs) given the species tree

- Locus tree model (backward)
  - Coalescent process evolving inside a species tree (MultiSpecies Coalescent)
  - Process that allows the discordance between locus trees and species trees (incomplete lineage sorting)



Mutations happen over time

- SNP data model (forward)
  - mutation (red $\leftrightarrow$ green) : markov model evolving along the locus tree branches
  - u : mutation rate red $\rightarrow$ green
  - v : mutation rate green $\rightarrow$ red



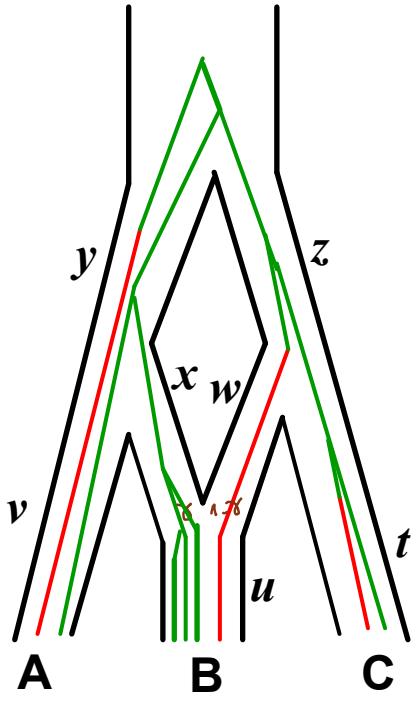
- Random variables : rRoot, nRoot, rIntNode, nIntNode, rA, rB, rC
- nA, nB, nC are not random
- Data=(rA, rB, rC)
- Likelihood : $\mathbb{P}(Data \mid S)$ with S species tree

Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

Network context

- Locus tree model (backward) :
  - Coalescent process
  - Nakhleh's model at the reticulation node
    $\Rightarrow$ Multispecies Network Coalescent
- SNP data model (forward)



- Random Variable : rRoot, nRoot, rIntNode, nIntNode, rA, rB, rC
- nA, nB, nC are not random
- Data=(rA, rB, rC)
- Likelihood : $\mathbb{P}(Data \mid N)$ with N network

SNAPPNet : a new Bayesian method for inferring networks

- N : phylogenetic network (topology, branch lengths, population sizes, inheritance probability)
- X_i : data for locus i
- G_i : locus tree for locus i
- m loci

$$\mathbb{P}(N|X_1,\ldots,X_m) \propto \left(\prod_{i=1}^{m}\int_{\psi}\mathbb{P}(X_i|G_i)\mathbb{P}(G_i|S)dG_i\right)P(N)$$
$$\propto \mathbb{P}\left(Data \mid N\right)P(N)$$

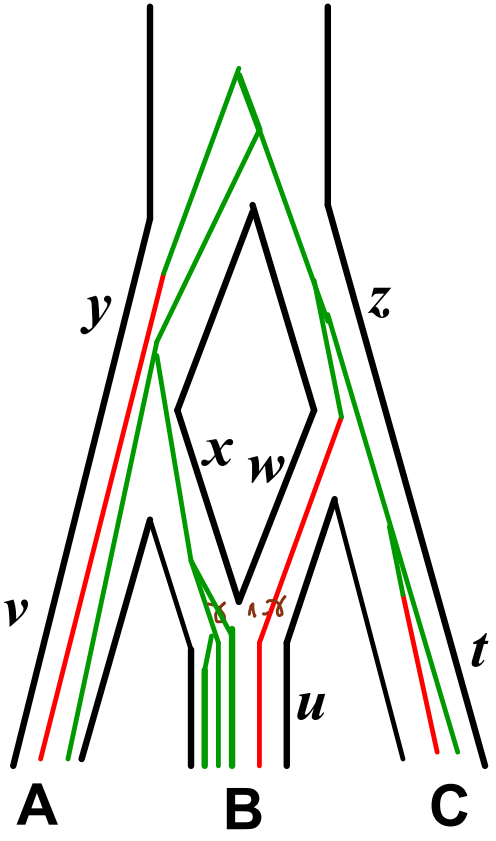
SNAPPNET integrates over all locus trees (generalization of SNAPP, Bryant et al. MBE 2012), using new algorithms dedicated to networks

Computation of the *prior* P(N) by the birth hybridization process of Zhang et al. (MBE 2018)

$\Rightarrow$ Markov Chain Monte Carlo (MCMC) in order to sample from the posterior distribution $\mathbb{P}(N|X_1,\ldots,X_m)$

Implemented within BEAST

Specificities of phylogenetic networks



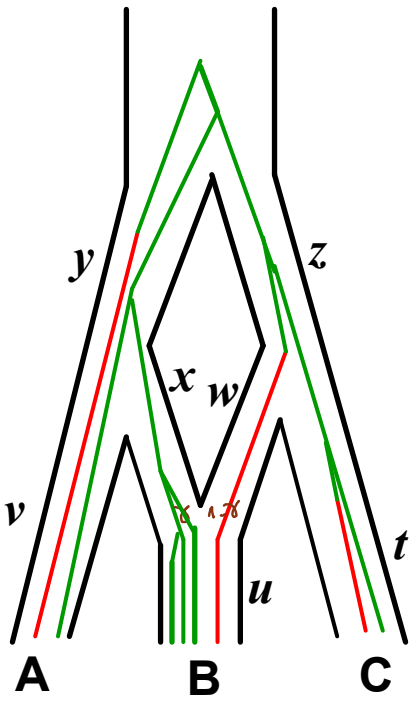
$Data_z$: red/green percentages in species below branch z

$Data_y$: red/green percentages in species below branch y

$Data_{zT}$ and $Data_{yT}$ are not independent ...

$Data_{zT}$ and $Data_{yT}$ contain the red and green alleles of the hybrid species

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

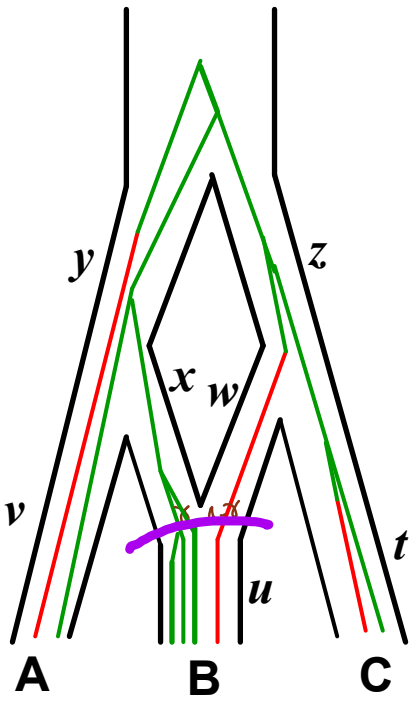
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

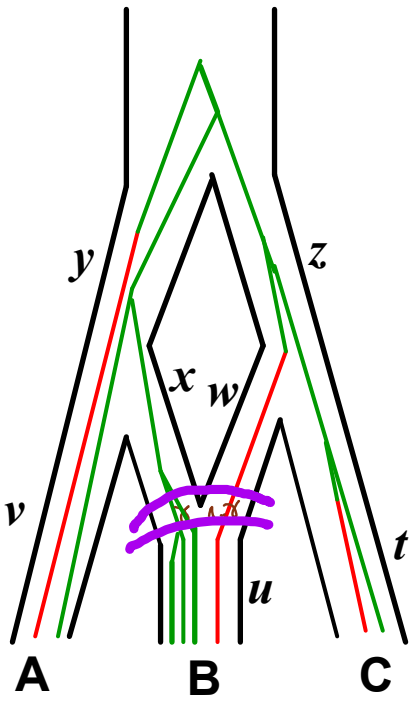
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

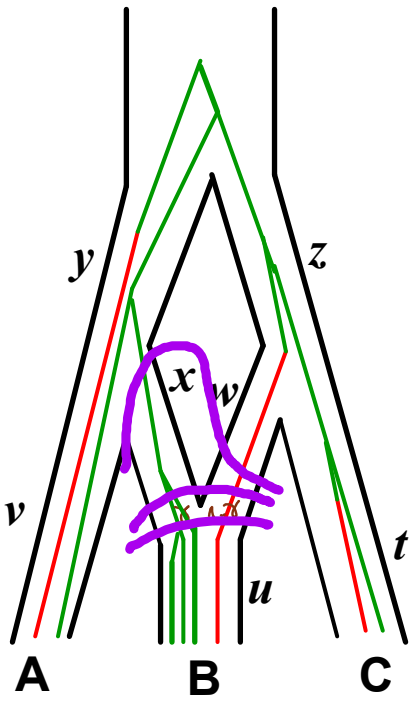
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

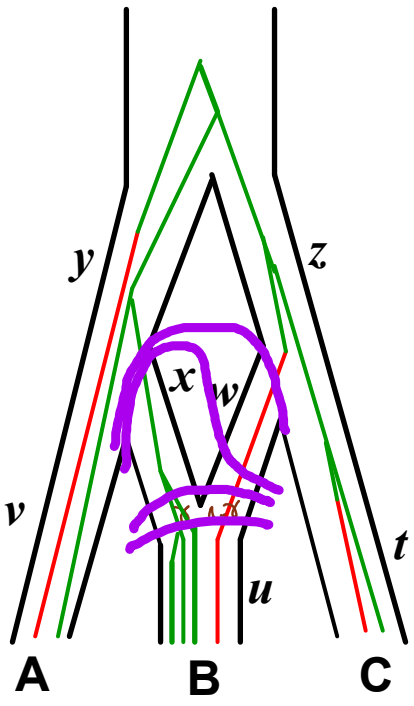
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

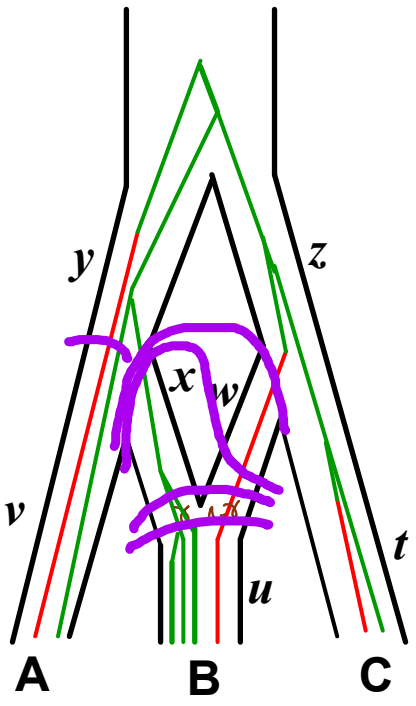
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

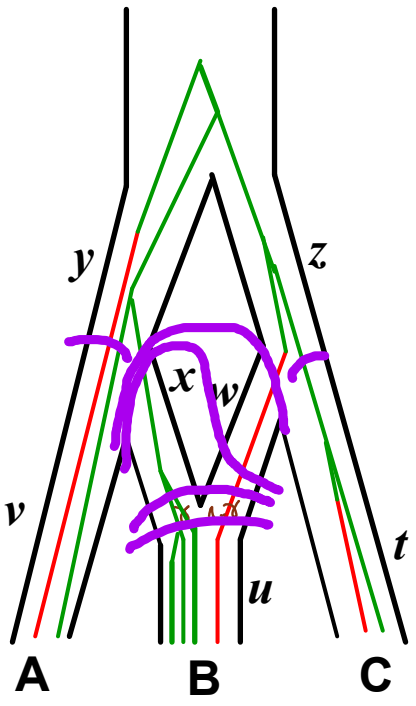
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

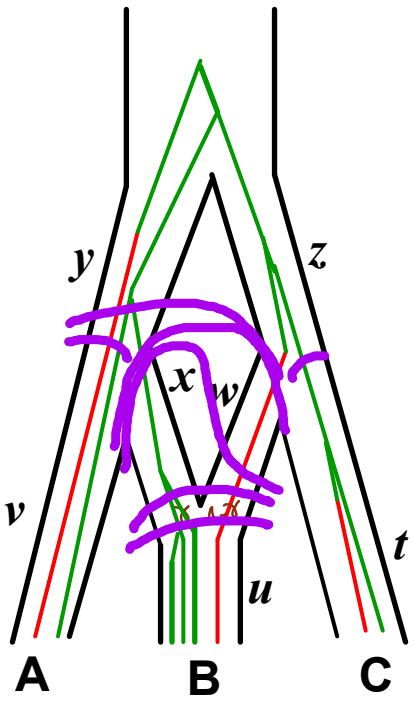
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

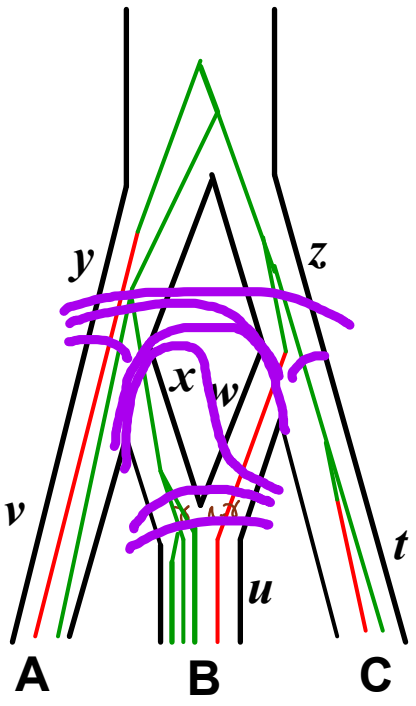
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

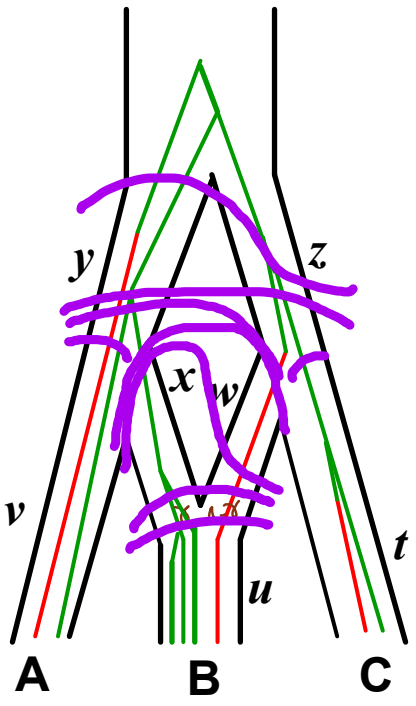
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

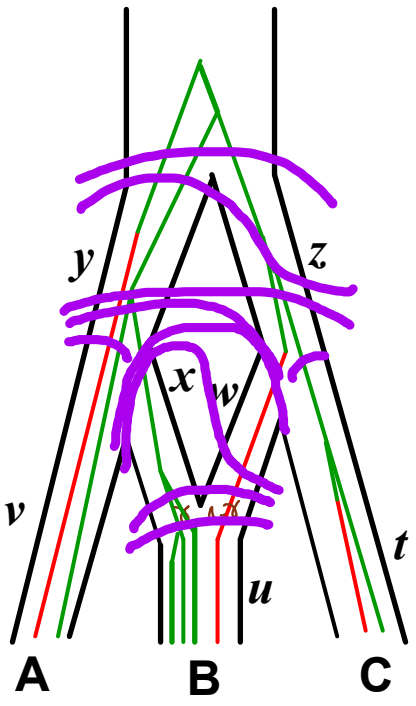
(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Our algorithm based on joint distributions



Quantities computed consecutively

(1) $\mathbb{P}\left(Data_{uT} \mid n_{uT}, r_{uT}\right)$

(2) $\mathbb{P}\left(Data_{xB}\, Data_{wB} \mid n_{xB}, r_{xB}, n_{wB}, r_{wB}\right)$

(3) $\mathbb{P}\left(Data_{xT}\, Data_{wB} \mid n_{xT}, r_{xT}, n_{wB}, r_{wB}\right)$

(4) $\mathbb{P}\left(Data_{xT}\, Data_{wT} \mid n_{xT}, r_{xT}, n_{wT}, r_{wT}\right)$

(5) $\mathbb{P}\left(Data_{vT} \mid n_{vT}, r_{vT}\right)$

(6) $\mathbb{P}\left(Data_{tT} \mid n_{tT}, r_{tT}\right)$

(7) $\mathbb{P}\left(Data_{yB}\, Data_{wT} \mid n_{yB}, r_{yB}, n_{wT}, r_{wT}\right)$

(8) $\mathbb{P}\left(Data_{yB}\, Data_{zB} \mid n_{yB}, r_{yB}, n_{zB}, r_{zB}\right)$

(9) $\mathbb{P}\left(Data_{yT}\, Data_{zB} \mid n_{yT}, r_{yT}, n_{zB}, r_{zB}\right)$

(10) $\mathbb{P}\left(Data_{yT}\, Data_{zT} \mid n_{yT}, r_{yT}, n_{zT}, r_{zT}\right)$

(11) $\mathbb{P}\left(Data \mid n_{root}, r_{root}\right)$

Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

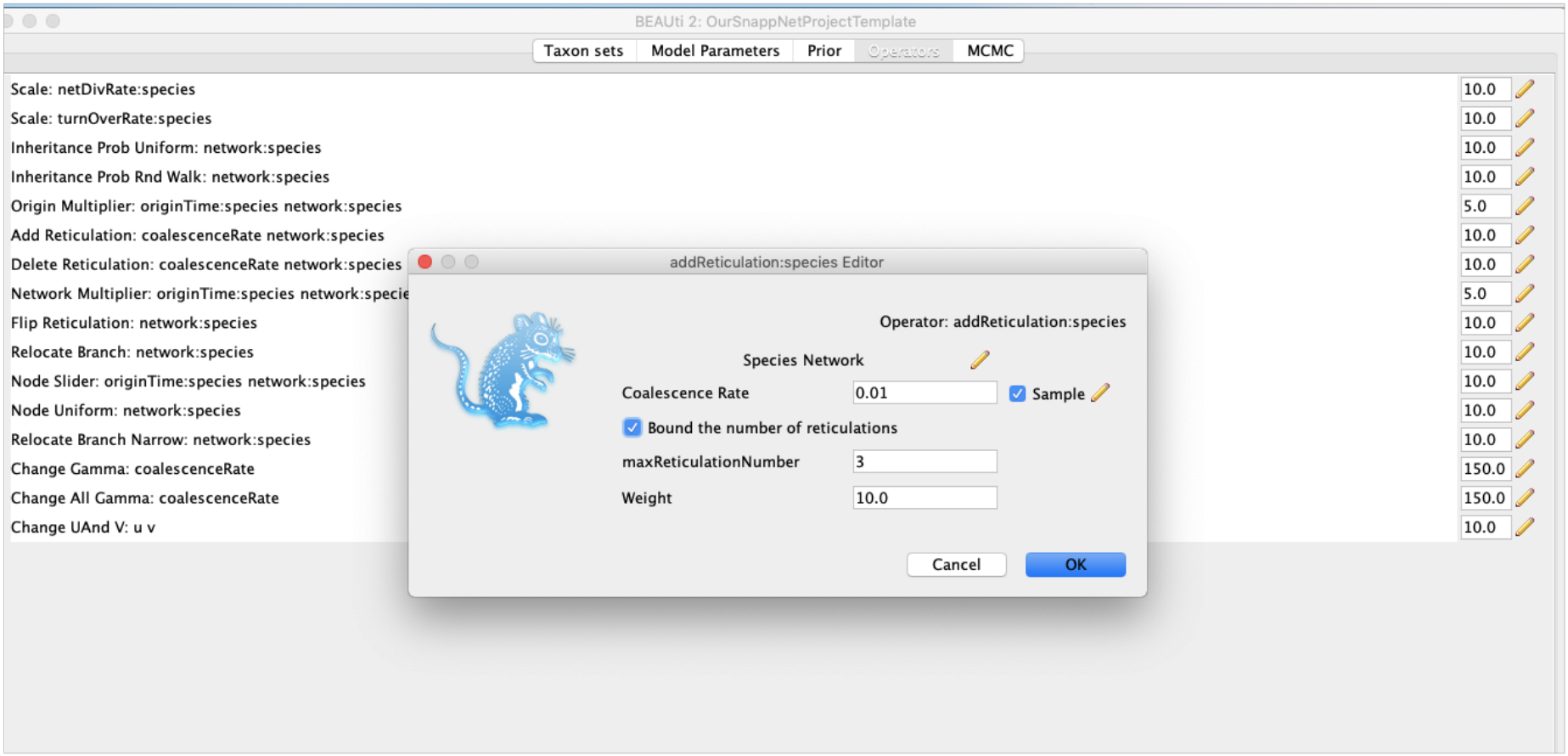
About SNAPPNET's xml (Add On for Beast)

<distribution id="networkPrior"
spec="speciesnetwork.BirthHybridizationModel"
network="@network :species" netDiversification="@netDivRate :species"
turnOver="@turnOverRate :species"/>
<prior id="netDivPrior" name="distribution" x="@netDivRate :species">
<Exponential id="exponential.01" name="distr" mean="10.0"/>
</prior>
<prior id="turnOverPrior" name="distribution" x="@turnOverRate :species">
<Beta id="betadistr.01" name="distr" alpha="1.0" beta="1.0"/>
</prior>

How to get SNAPPNET's xml :

BEAUti — Bayesian Evolutionary Analysis Utility.
This program is used to import data, design the analysis, and generate the BEAST control file.

BEAUti : how to choose the number of reticulations



SNAPPNET available at https ://github.com/rabier/MySnappNet

Roadmap

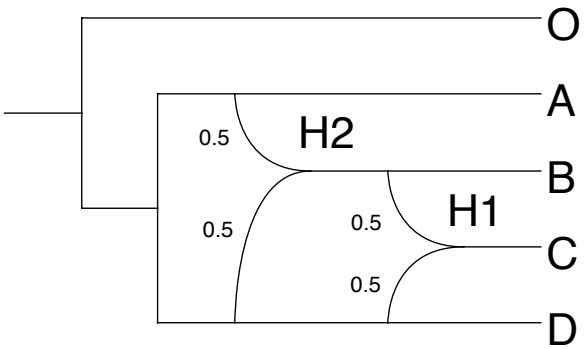
1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

Time required to compute the likelihood $\mathbb{P}(Data \mid N)$

Dataset ID	CPU time	
	SNAPPNET (in minutes)	MCMCBiMarkers (in hours)
1	5.559	35.9354
2	5.6763	34.2433
3	5.7351	32.6519
4	5.446	34.2011
5	5.5996	33.2354

SNAPPNET vs MCMCBiMarkers
(Zhu et al., Plos Comp Biol 2018)

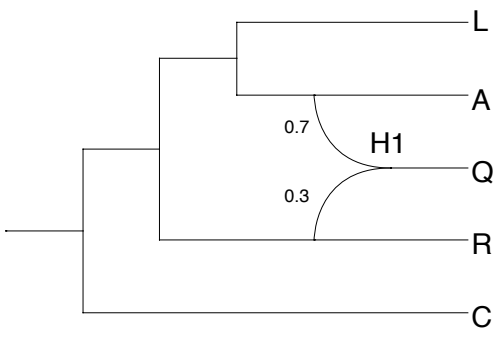
$O(n^8)$ vs $O(n^{12})$



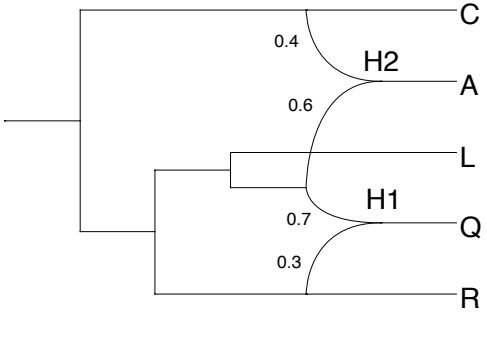
Network C of level 2

SNAPPNET's ability to recover networks A and B taken from Zhu et al (Plos Comput Biol, 2018)

Number of sites	1,000	10,000	100,000
Network A	0%	100%	100%
Network B	0%	81.25%	100%



Network A



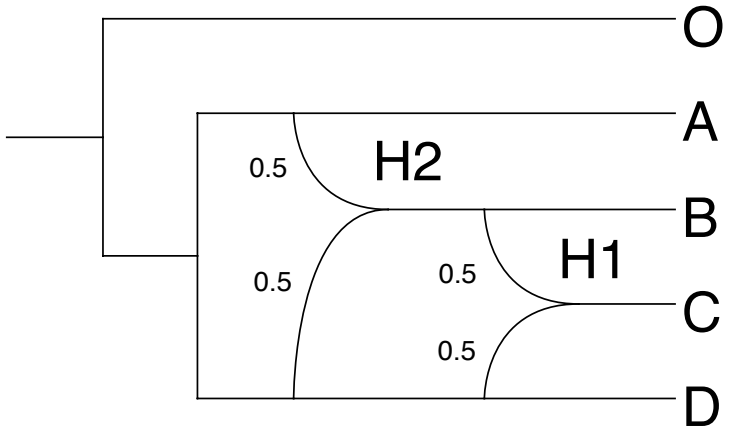
Network B

- 1000 sites : MCMCBiMarkers $>$ SNAPPNET
- $\geq$ 10000 sites : SNAPPNET $\approx$ MCMCBiMarkers
- 10 000 sites are required to infer these networks

Ability to recover the topology of network C

SNAPPNET vs MCMCBiMarkers

	Number of sites		
Number of lineages for B and for C	**1,000**	**10,000**	**100,000**
1	0% 0%	7.87% 4.84%	54.90% 0%
4	0% 0%	50.00% 0%	49.60% 0%



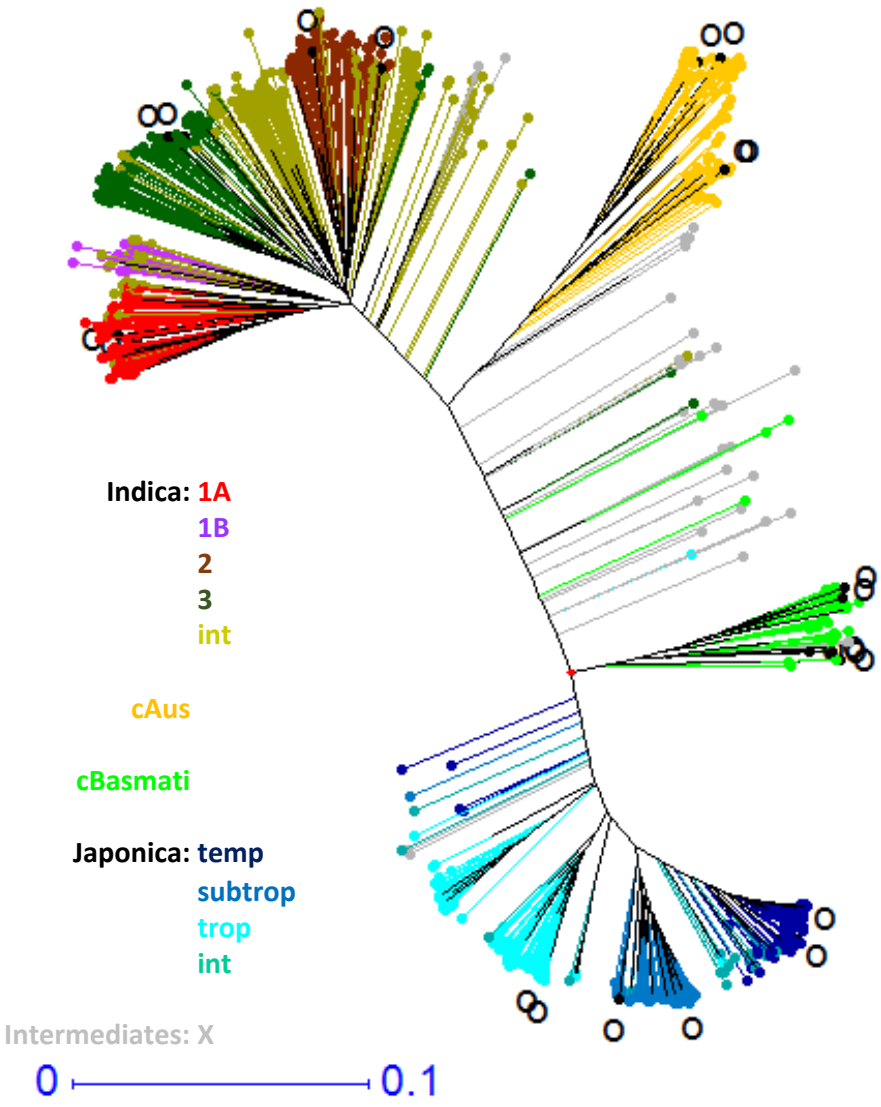
Network C

Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

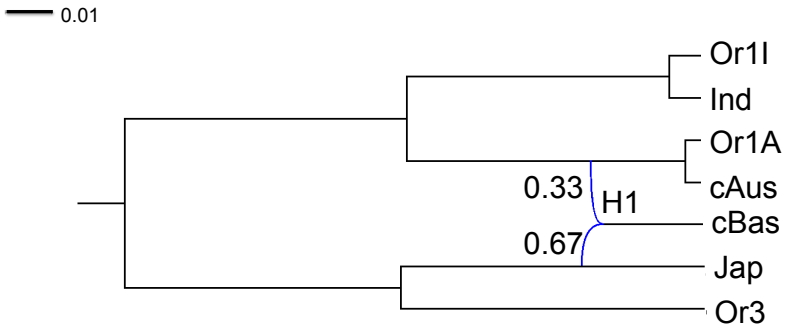
Varieties selected by J.C. Glaszmann

Neighbour joining tree based on Wang et al. (Nature, 2018), 3000 rice varieties (4.8 millions of SNPs)

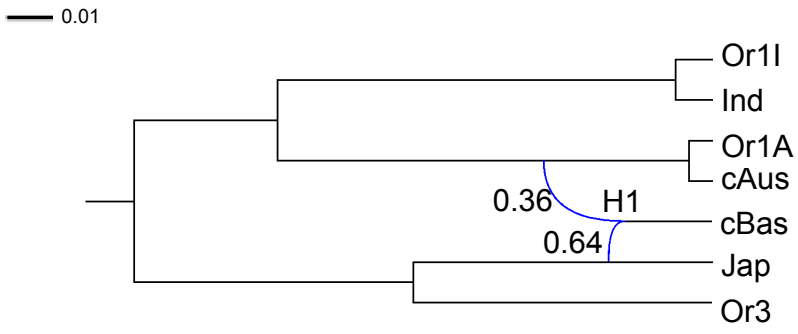


SNAPPNET in the Bayesian MCMC setting

- 2 different samplings of 10K SNPs
- 2 Markov chains per sampling
- Number of reticulations bounded by 2
- 10 millions iterations



First sampling
ESS=844, ESS=1159



Second sampling
ESS=971, ESS=535

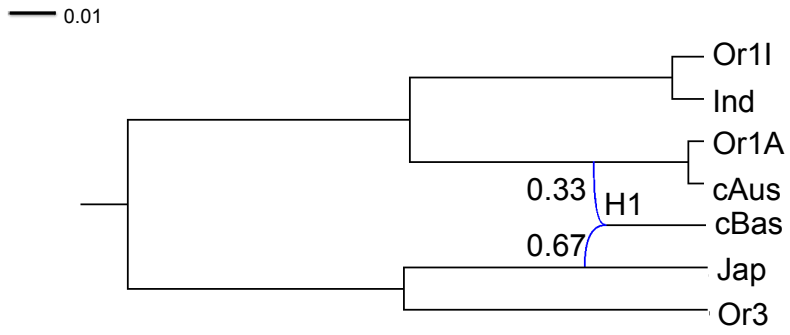
The cultivars are associated with the expected wild forms

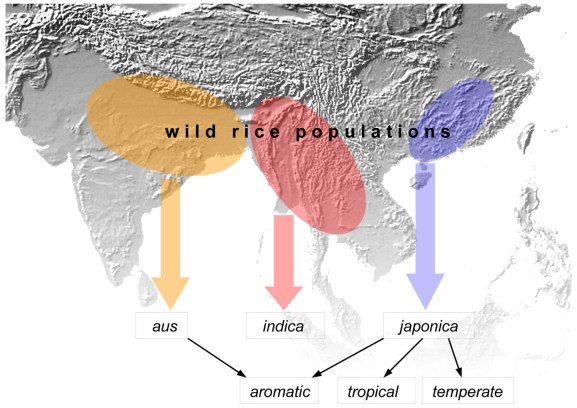
Summary of the different analyses

The inferred networks reveal stable features :

- correspondence between wild subpopulations and cultivated subpopulations
- the early divergence of Japonica, that predates the one between Indica and cAus
- the mobilisation of early Japonica cultivars to combine with the cAus pillar to produce the fourth varietal type cBas
- the indication that this hybridization may have occured before the domestication of cAus

There is an agreement with Civan's thesis, but we observe a few differences ...





Roadmap

1. Introduction
2. Species tree inference with the SNAPP method
3. Network inference
   - Our new method SNAPPNET
     - Algorithm
     - BEAST (Beauti)
   - Comparison SNAPPNET vs MCMCBiMarker
   - Rice real data
4. Conclusion

Conclusion

- SNAPPNET available at https ://github.com/rabier/MySnappNet
- SNAPPNET implemented within the BEAST 2 framework
- Paper : “On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo”, Rabier et al. (PLoS Comput Biol, 2021)
- The gain in speed allows us to consider more complex evolutionary scenarios
- It is also possible to evaluate the likelihoods of evolutionary scenarios

Acknowledgements

Céline Scornavacca
Marnus Stoltz

Vincent Berry
Fabio Pardi

Jean-Christophe Glaszmann
João D. Santos

Jean-Michel Marin

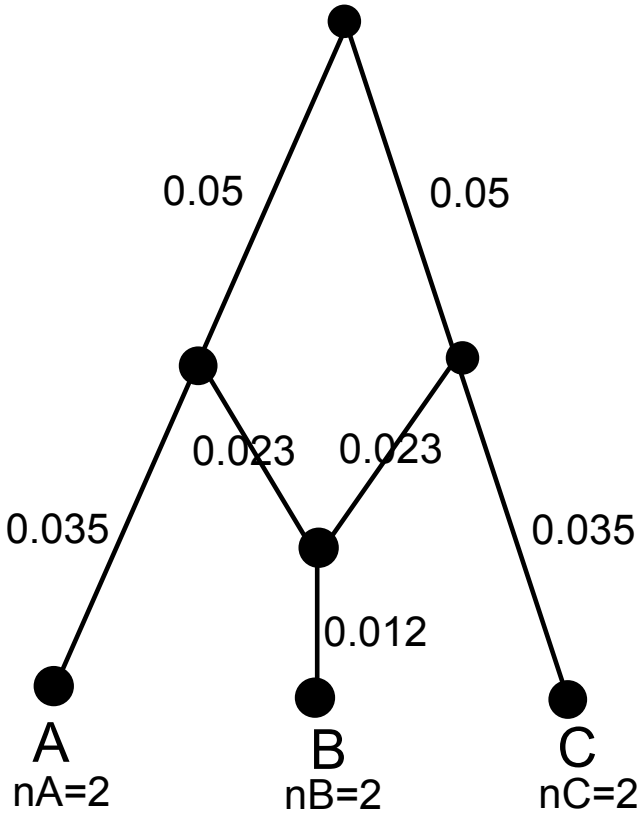
Angélique D'Hont
Manuel Ruiz

Marilyne Summo

References

- Huang et al (Nature, 2012). “A map of rice genome variation reveals the origin of cultivated rice”
- Civan et al (Nature plants, 2015). “Three geographically separate domestications of Asian rice”
- Wang et al (Nature, 2018). “Genomic variation in 3,010 diverse accessions of Asian cultivated rice”
- Bryant et al. (MBE, 2012). “Inferring species trees directly from biallelic genetic markers : bypassing gene trees in a full coalescent analysis”
- Zhang et al. (MBE, 2017). “Bayesian inference of species networks from multilocus sequence data”
- Zhu et al. (PLoS Comput Biol, 2018). “Bayesian inference of phylogenetic networks from bi-allelic genetic markers”

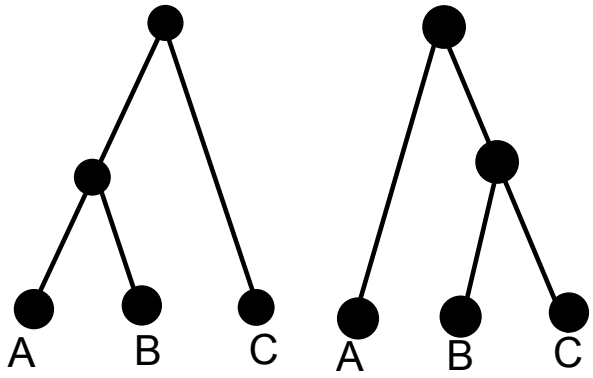
An example on simulated data



- Branch lengths in expected number of mutations per site
- nA=2, nB=2, nC=2
- 1 000 sites or 10 000 sites
- Population sizes θ equal to 0.005 or 0.05
- T : coalescent time for 2 lineages (in mutations par site)
  - if $\theta = 0.005$, then $\mathbb{E}(T) = 0.005/2 = 0.0025$
  - if $\theta = 0.05$, then $\mathbb{E}(T) = 0.005/2 = 0.025$

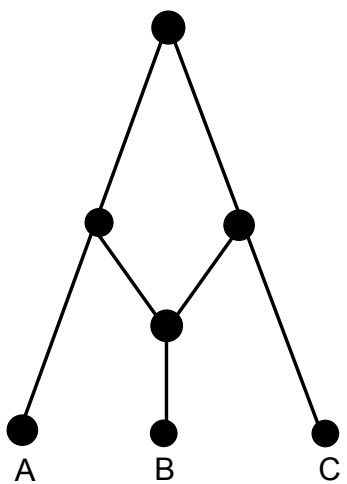
Networks sampled by MCMC

- 1 000 sites, $\theta = 0.005$



51.94% 48.06%

- 10 000 sites, $\theta = 0.005$
- 1 000 sites, $\theta = 0.05$
- 10 000 sites, $\theta = 0.05$



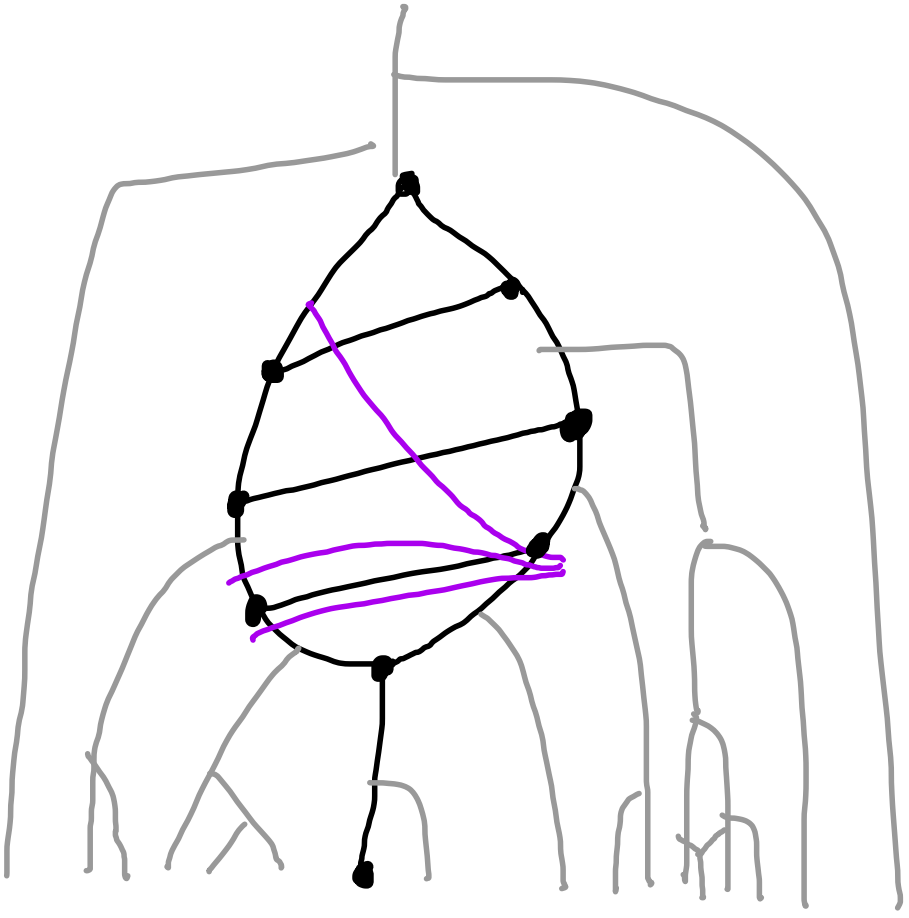
100%

When the population size is larger,
we need more sites to recover the network

We try to minimize the number of branches simultaneously considered in our joint distributions

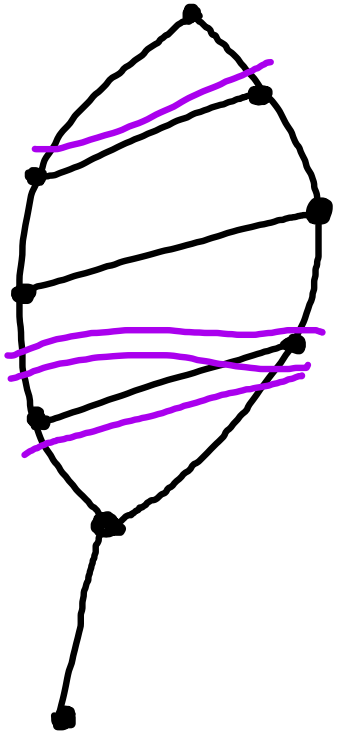
Strategy to avoid

A maximum of 5 branches



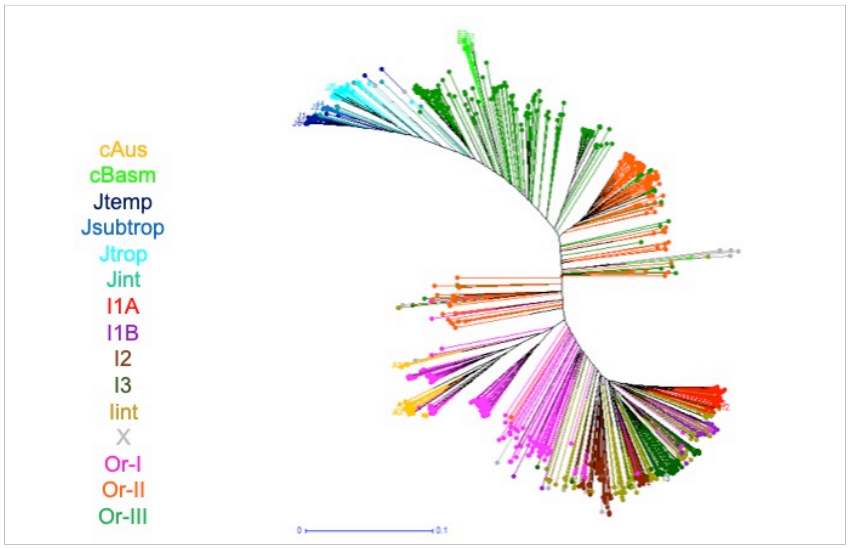
Strategy to adopt

A maximum of 3 branches



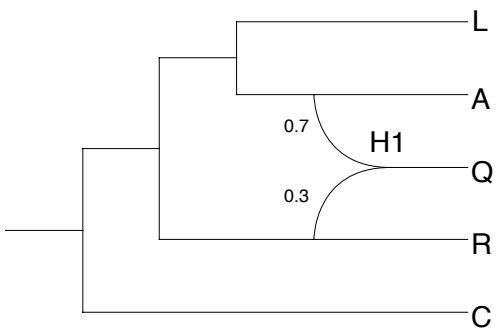
Wild forms added to our datasets

- W1559 : Or1 (Or1I), Thaïland, close to Indica
- W1117 : Or1 (Or1I), India, within Indica
- W1747 : Or1 (Or1A), India, within cAus
- W0574 : Or1 (Or1A), Malaysia, close to cAus
- W3042 : Or3, China, within Japonica
- W3048 : Or3, China, between Japonica and cBasmati
- W3073 : Or3, China, other side of cBasmati

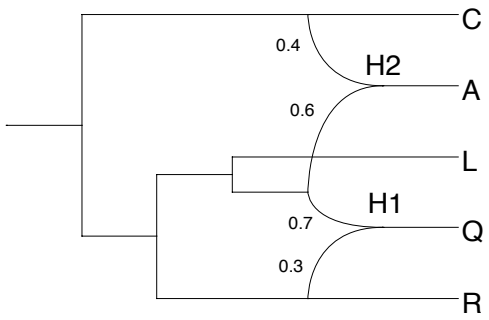


NJ tree (Glaszmann, Wang)

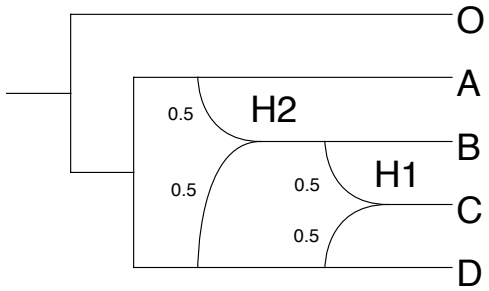
3 networks studied by simulation



Network A



Network B



Network C

Networks A and B taken from Zhu et al (Plos Comput Biol, 2018)

MCMCBiMarkers vs SNAPPNET

- Network A : $O(n^8)$ vs $O(n^6)$
- Network B and C : $O(n^{12})$ vs $O(n^8)$