## An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications, and transfers

(work in progress)

Jean-Philippe Doyon <sup>1</sup> Celine Scornavacca<sup>2</sup> Gergely J. Szöllősi<sup>3</sup> Vincent Ranwez<sup>4</sup> Vincent Berry<sup>1</sup>

LIRMM, CNRS - Univ. Montpellier 2, France.
 Center for Bioinformatics (ZBIT), Tuebingen Univ., Germany.
 LBBE, CNRS - Univ. Lyon 1, France.
 ISEM, CNRS - Univ. Montpellier 2, France.

Bertinoro meeting May 2010

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### Inferring the Tree of Life

Is there at all a Tree of Life?



- Evidence that vertical signal from the early stages of life is still visible in current genomes [DAUBIN ET AL 03, KURLAND ET AL 03, ...]
- The NCBI tree contains 64% unresolved nodes
- Only using single-copy genes  $\rightarrow$  "Tree of 1%" [BAPTESTE ET AL 07]
- A reasonable and assessed ToL might only be obtained if we are able to extract some signal from multigene families

 Model of reconciliation

Algorithm

Experimental results

Conclusion

#### Gene sequence evolution

Example of a multigene family tree G:



Gene sequences are submitted to complex macro-evolutionary events

Model of reconciliation

Algorithm

Experimental results

Conclusion

### Gene trees can (often) differ from species trees

Reconciliation approach: explicitly accounts for (some) macro events:



Reconciliation methods apply to multigene family trees G, explaining incongruence with S by postulating macro events

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### Motivation(s) for reconciliation



- Identify orthologous sequences: transfers of functional annotation
- · Part of a complex process to infer more accurate species trees

• ...

Model of reconciliation

Algorithm

Experimental results

Conclusion

### A parsimony approach to reconcile trees

Preference goes to the most parsimonious history in the number of non-speciation events (avoiding unnecessary events).



most probable history

unlikely history

Model of reconciliation

Algorithm

Experimental results

Conclusion

### A parsimony approach to reconcile trees

#### Remarks

- This criterion is reasonable when assuming speciation is the dominant event (over transfers, duplications and losses)
- In this case, parsimony might be thought of as an approximation of the maximum likelihood criterion to estimate the gene history
- Parsimony is less refined than the maximum likelihood criterion, but is faster to compute

Genomic databases contain several dozen thousand gene families: reconciliation needs to be as fast as possible to cope with NGS technology [PENEL ET AL 09, ...]

Model of reconciliation

Algorithm

Experimental results

Conclusion

### The MPR problem

The Most Parsimonious Reconciliation problem

- Consider a set of macro events affecting genes:
  - Speciation (S)
  - Duplication (D)
  - Transfer  $(\mathbb{T})$
  - Loss (L)
- Give a cost to each considered event:
- Find a reconciliation that min. the overall cost of  $\mathbb{S}$ ,  $\mathbb{D}$ ,  $\mathbb{T}$ ,  $\mathbb{L}$  events.

#### Transfers

- must occur between concomitant species;
- ...thus impose chronological constraints between edges of S.

Model of reconciliation

Algorithm

Experimental results

Conclusion

### The MPR problem

#### Transfers

- must occur between concomitant species;
- thus some combination of T between edges of S are inconsistent.
- makes hard life for solving MPR : general pbm is NP-complete [HL0



Model of reconciliation

Algorithm

Experimental results

Conclusion

### The MPR problem

#### Transfers

- must occur between concomitant species;
- thus some combination of T between edges of S are inconsistent.
- makes hard life for solving MPR : general pbm is NP-complete [HL04]



Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Previous approaches & models**

#### Species Graph SG

[Gorecki]

- Transfers are allowed along inserted horizontal branches.
- Given SG, MPR is solved in  $O(|SG|^3 \cdot |G|)$ .
- Computing an optimal SG is hard to solve.

#### Some existing reconciliation models

- Map each node of G onto a node of S, but this is not sufficient for Transfer + Loss. [VERNOT ET AL 08]
- May have pbms to directly account for ⊥ [HALLETT & LAGERGREN 04]
- Handle transfers locally, which can lead to time inconsistent
  reconciliations (Tarzan & Jane software)
  [MERKLE ET AL 05-10]

#### Dating the species tree S

LAGERGREN'S GROUP 09-10, LYUBETSKY ET AL 09, MERKLE ET AL 05-10]

- Transfers between two concomitant branches (Local Consistency)
- Subdividing S into an S' tree with time slices (Global Consistency)
- $\rightarrow$  algorithms in  $O(|S|^4 \cdot |G|^4)$  and  $O(|S'|^3 \cdot |G|)$

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Previous approaches & models**

#### Species Graph SG

[Gorecki]

- Transfers are allowed along inserted horizontal branches.
- Given SG, MPR is solved in  $O(|SG|^3 \cdot |G|)$ .
- Computing an optimal SG is hard to solve.

#### Some existing reconciliation models

- Map each node of G onto a node of S, but this is not sufficient for Transfer + Loss. [VERNOT ET AL 08]
- May have pbms to directly account for ⊥ [HALLETT & LAGERGREN 04]
- Handle transfers locally, which can lead to time inconsistent
  reconciliations (Tarzan & Jane software)
  [MERKLE ET AL 05-10]

### Dating the species tree S

[LAGERGREN'S GROUP 09-10, LYUBETSKY ET AL 09, MERKLE ET AL 05-10]

- Transfers between two concomitant branches (Local Consistency)
- Subdividing S into an S' tree with time slices (Global Consistency)
- $\rightarrow$  algorithms in  $O(|S|^4 \cdot |G|^4)$  and  $O(|S'|^3 \cdot |G|)$

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Our contribution**

For reconciling G and S while accounting for duplications ( $\mathbb{D}$ ), losses ( $\mathbb{L}$ ), transfers ( $\mathbb{T}$ ), and speciations ( $\mathbb{S}$ ):

### An efficient model for $\mathbb{DTLS}$ reconciliation

- Considering a species tree *S* that is dated (function  $\theta : V(S) \rightarrow \mathbb{R}$ ).
- Relying on 6 basic cases, each one being fast to investigate
- Considering loss events in combination with other events (SL and TL),

### A dynamic programming algorithm

- exact: solves MPR under the above model
- fast: runs in time O(|S'| · |G|) where S' is a subdivision of S in a small number of time slices.

### Experimental results for the relevance of parsimony

Is parsimony relevant to infer the evolutionary scenario of a gene family?

Model of reconciliation

Algorithm

Experimental results

Conclusion

### An Efficient Reconciliation Model

#### Dealing with the dated species tree S

- We add new internal vertices to partition the tree into successive sets of concomitant edges. Thus, we obtain a subdivision *S*<sup>'</sup> of the dated species tree *S*.
- In other words, time is discretized into time slices [GORBUNOV ET AL 09, LIBESKIND-HADAS 09, TOFIGH ET AL 10].
- Compared to an accurate ML model, a smaller number of time slices is used.
- We renumber the time stamps. Harmless as parsimony is not able to estimate precise dates within a given time interval.



Model of reconciliation

Algorithm

Experimental results

Conclusion

### An Efficient Reconciliation Model

#### Dealing with the dated species tree S

- We add new internal vertices to partition the tree into successive sets of concomitant edges. Thus, we obtain a subdivision *S*' of the dated species tree *S*.
- In other words, time is discretized into time slices [GORBUNOV ET AL 09, LIBESKIND-HADAS 09, TOFIGH ET AL 10].
- Compared to an accurate ML model, a smaller number of time slices is used.
- We renumber the time stamps. Harmless as parsimony is not able to estimate precise dates within a given time interval.



Model of reconciliation

Algorithm

Experimental results

Conclusion

### An Efficient Reconciliation Model

#### Dealing with the dated species tree S

- We add new internal vertices to partition the tree into successive sets of concomitant edges. Thus, we obtain a subdivision *S*' of the dated species tree *S*.
- In other words, time is discretized into time slices [GORBUNOV ET AL 09, LIBESKIND-HADAS 09, TOFIGH ET AL 10].
- Compared to an accurate ML model, a smaller number of time slices is used.
- We renumber the time stamps. Harmless as parsimony is not able to estimate precise dates within a given time interval.



Introduction	
000000000	)

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### A reconciliation between G and S

- plunges G into the subdivision S', which
  - defines a map α : E(G) → 2<sup>E(S')</sup> assigning each edge of G onto an ordered sequence of branches of S' having non-decreasing time stamps.
  - induces events that label internal nodes of  $G(\mathbb{S},\mathbb{T},\mathbb{D})$  and time slices/stamps for these nodes.
  - induces a completed gene tree  $G^{\circ}$  where  $\mathbb{L}$  leaves have been attached to G.



Model of reconciliation

Algorithm

Experimental results

Conclusion

### Atomic events considered by the model

The model distinguishes 6 events or combinations thereof:



Note: for simplicity, losses are considered in combination with other events

Model of reconciliation

Algorithm

Experimental results

Conclusion

### Properties of the model in a parsimony framework

Implications due to the parsimony criterion:

#### Property 1

 $\mathbb{SL}$  and  $\mathbb{TL}$  are parsimonious event associations.

This means that when a single G edge goes through a speciation node s of S', there is no need to examine scenarios on the children of s that are not seen in G.

#### Property 2

Each edge of *G* never follows several  $\mathbb{TL}$  in a row at a same time slice *t* in *S'* (i.e. any  $\mathbb{TL}$  event is followed by a different event).

This allows dynamic programming to make progress without looking at more than two concomitant branches of S' at once (progress is done either in time, or in gene tree nodes).

Model of reconciliation

Algorithm •00000 Experimental results

Conclusion

### **Dynamic Programming Algorithm**

#### Input

- Dated species tree  $(S, \theta)$
- Gene tree G (could be dated also)
- Respective costs τ, δ, λ (and σ) for DTL (and S) events

#### Output

- A Most Parsimonious Reconciliation α between G and (S, θ)
- Indicating where in S, DTL events occured

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

#### Principle

The DP algorithm processes edges  $(u_p, u)$  of *G* in postorder from leaves to the root, and when considering an  $(u_p, u)$ , it successively considers *S* branches slice by slice, also from the leaves to the root



- 1: for all edge  $(u_p, u)$  of G following a bottom-up traversal do
- 2: for all time t of S' in backward time order do
- 3: for all branch  $(x_p, x)$  of S' located at time t do
- 4: ...

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

The DP algorithm is based on the previous parsimony model.

Recall: six (combinations of) events can explain the presence of a gene edge  $(u_p, u)$  inside an *S'* branch  $(x_p, x)$ : {S, D, Ø, SL, T, TL}

#### Remark

The complexity of existing parsimony algorithms is hampered by the cost of examining possible transfers between branches of S'

#### Idea 1

In our model, two TL cannot happen in a row in a same time slice (Prop. 1). Hence, for a pair (  $(u_{\rho}, u), (x_{\rho}, x)$  ) we can examine the TL senario, after all 5 others have been examined.

This is good, as predicting a  $\mathbb{TL}$  event doesn't make the DP progress in the examination of *G* nodes, but then you're guaranteed to make progress in *t*.

Then we can examine TL events separately from the others (afterwards)

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

First (regular) version of the DP algorithm:

1: 2:	<ol> <li>for all edge (u<sub>p</sub>, u) of G following a bottom-up traversal do</li> <li>for all time t of S' in backward time order do</li> </ol>			
3:	for all branch $(x_p, x)$ of S' located at time t do			
4:	for all $\mathbb{E} \in \{\mathbb{S}, \mathbb{D}, \varnothing, \mathbb{SL}\}$ do			
5:	Compute $Cost_{\mathbb{E}}(u, x)$			
6:	for all branch $(y_p, y) \neq (x_p, x)$ of S' located at time t do			
7:	$Cost_{\mathbb{T}}(u, x) \leftarrow min \{Cost(u_1, x) + Cost(u_2, y) + \tau,$			
8:	$Cost(u_2, x) + Cost(u_1, y) + \tau,$			
9:	$Cost_{\mathbb{T}}(u, x)$ for previous $(y_{\rho}, y)$ tubes }			
10:	$Cost_{\mathbb{S}}(u, x) \leftarrow \min_{\mathbb{E}} \{Cost_{\mathbb{E}}(u, x)\} \text{ for } \mathbb{E} \in \{\mathbb{S}, \mathbb{D}, \emptyset, \mathbb{SL}, \mathbb{T}\}$			
11:	for all branch $(x_p, x)$ of S' located at time t do			
12:	for all branch $(y_p, y) \neq (x_p, x)$ of S' located at time t do			
13:	$Cost_{\mathbb{TL}}(u, x) \leftarrow min \{ Cost_{5}(u, y) + \tau + \lambda, \}$			
14:	$Cost_{TL}(u, x)$ for previous $(y_{\rho}, y)$ tubes }			
	$Cost(u, x) \leftarrow min\{Cost_{\mathbb{TL}}(u, x), Cost_{5}(x, u)\}$			

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

- The first idea allows the DP algorithm to avoid considering combinations of transfers in a row (i.e. combinations of 3 concomitant branches of *G* [GORBUNOV ET AL 09]
- Hence basically to win an O(|S'|) factor.

Now

#### Idea 2

When trying to place  $(u_p, u)$  inside tube  $(x_p, x)$  while involving a transfer ( $\mathbb{T}$  or  $\mathbb{TL}$ ), we have to examine all potential landing places  $(y_p, y)$  at time *t* in *S*'.

However, the best landing place is independent of the donor branch  $(x_p, x)$ ). Hence we can factorize the computation of the best receiver.

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

- The first idea allows the DP algorithm to avoid considering combinations of transfers in a row (i.e. combinations of 3 concomitant branches of *G* [GORBUNOV ET AL 09]
- Hence basically to win an O(|S'|) factor.



#### Idea 2

When trying to place  $(u_{\rho}, u)$  inside tube  $(x_{\rho}, x)$  while involving a transfer ( $\mathbb{T}$  or  $\mathbb{TL}$ ), we have to examine all potential landing places  $(y_{\rho}, y)$  at time *t* in *S*'.

However, the best landing place is independent of the donor branch  $(x_p, x)$ ). Hence we can factorize the computation of the best receiver.

In	tr	od	uc	tio	n		
0	0	00	00	00	00	00	

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

- 1: for all edge  $(u_p, u)$  of G following a bottom-up traversal do
- 2: for all time t of S' in backward time order do
- 3: Compute the (two) best receiver(s)  $y^*$  for  $(u, u_1)$  and  $(u, u_2)$  at time t
- 4: for all branch  $(x_p, x)$  of S' located at time t do
- 5: Compute  $\mathbb{E} \in \{\mathbb{S}, \mathbb{D}, \emptyset, \mathbb{SL}\}$  that minimizes  $Cost_{\mathbb{E}}(u, x)$
- 6:  $Cost_{\mathbb{E}}(u, x) \leftarrow min\{Cost_{\mathbb{E}}(u, x), Cost(u_i, y^*) + Cost(u_j, x) + \tau\}$
- 7: Compute the (two) best receiver(s),  $y^*$  at time t in term of  $Cost_5(u, y^*)$
- 8: for all branch  $(x_p, x)$  of S' located at time t do
- 9:  $Cost(u, x) \leftarrow min\{Cost_5(u, x), Cost_5(u, y^*) + \tau + \lambda\}$

10: return Cost(r(G), r(S))

#### heorem

- 1. Cost(r(G), r(S)) is the cost of a Most Parsimonious Reconciliation
- **2.** The algorithm runs in  $\Theta(|G| \cdot |S'|)$
- 3. Same optimization applies to ML version of the DP algorithm [SZÖLLÖZI ET AL IN PREP.]

Intro	oduo	ctior	
00	000	000	000

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Dynamic Programming Algorithm**

- 1: for all edge  $(u_p, u)$  of G following a bottom-up traversal do
- 2: for all time t of S' in backward time order do
- 3: Compute the (two) best receiver(s)  $y^*$  for  $(u, u_1)$  and  $(u, u_2)$  at time t
- 4: for all branch  $(x_p, x)$  of S' located at time t do
- 5: Compute  $\mathbb{E} \in \{\mathbb{S}, \mathbb{D}, \emptyset, \mathbb{SL}\}$  that minimizes  $Cost_{\mathbb{E}}(u, x)$
- 6:  $Cost_5(u, x) \leftarrow min\{Cost_{\mathbb{E}}(u, x), Cost(u_i, y^*) + Cost(u_j, x) + \tau\}$
- 7: Compute the (two) best receiver(s),  $y^*$  at time t in term of  $Cost_5(u, y^*)$
- 8: for all branch  $(x_p, x)$  of S' located at time t do
- 9:  $Cost(u, x) \leftarrow min\{Cost_5(u, x), Cost_5(u, y^*) + \tau + \lambda\}$

10: return Cost(r(G), r(S))

#### Theorem

- 1. Cost(r(G), r(S)) is the cost of a Most Parsimonious Reconciliation
- **2.** The algorithm runs in  $\Theta(|G| \cdot |S'|)$
- 3. Same optimization applies to ML version of the DP algorithm [SZÖLLÖZI ET AL IN PREP.]

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Simulations**

#### Species tree generation (software from A. Rambaut):

The following programs are now no longer supported This may mean they no longer run on newer computers or the analyses h by other software	ave been superceded
Phylogen	
Description:	Current version: v1.1
Further information and downloads	

# Simulation of genomic evolutionary events (DTLS) with rates typical of Archaea [CSÜROS & MIKLÖS 09]

### Streamlining and Large Ancestral Genomes in Archaea Inferred with a Phylogenetic Birth-and-Death Model

#### Miklós Csűrös\* and István Miklós†

\*Department of Computer Science and Operations Research, University of Montréal, Montréal, Canada; and †Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary

Homologous genes originate from a common ancestor through vertical inheritance, duplication, or broizontal gene transfer. Entire benotogi families spawned by a single ancestraf gene can be identified across multiple genomes based on protein sequence similarity. The sequences, however, do not always reveal conclusively the history of large families. To study the evolution of complete gene reperiories, we propose here a mathematical framework that does not rely on resolved gene family histories. We show that so-called phylogenetic profiles, formed by family sizes across multiple genomes, are sufficient to infer principal evolutionary trends. The main novely in our approach is an efficient algorithm to compute the likehood of a phylogenetic profile in a model of brith-ad-each processes calling on a phylogenetic

Introduction	
000000000	С

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Datasets**

#### Details of the simulation process:

- 10 species trees on 100 species (standard birth and death process, with b/d ratio = 1.25), subsequently made ultrametric (dates).
- DTL events generated according to a Poisson process running from the root to the tips, with loss rate  $r_{\lambda}$ , transfer rate  $r_{\tau}$  and duplication rate  $r_{\delta}$  applied to extant species at time *t* (obtained gene trees have between 59 and 93 leaves).

#### First dataset - ds1

Designed to simulate a relatively large time scale, comparable to an archaean or bacterial phylum:

- fixed loss rate  $r_{\lambda} = 0.7$
- fixed tree height h = 1
- 11 values for  $r_{\tau}$  and  $r_{\delta}$  varying in [0.01, 0.35]
- 5 gene trees per species tree in each of these 11  $\times$  11 conditions, leading to 6,050 gene trees.

Introduction	
0000000000	

Model of reconciliation

Algorithm

Experimental results

Conclusion

### **Datasets**

#### Second dataset - ds<sub>2</sub>

Designed to investigate the behavior of parsimony on different phylogenetic scales:

- four different tree heights *h* ∈ [0.2, 0.4, 0.8, 1.6]
- fixed ratio  $r_{\lambda}/(r_{\delta}+r_{\tau}+r_{\lambda})=0.7$

[CSUROS AND MIKLOS]

- varying the relative importance of transfers vs duplications: 11 values for  $r_{\tau} \in [0, 0.3]$  each time choosing  $r_{\delta} = 0.3 r_{\tau}$
- 20 gene trees per species tree in each of these 4  $\times$  11 conditions, leading to 8,800 gene trees.

#### Running the method:

- Crude costs are given to the DP for each macro event  $\mathbb{E}$ :  $1/\overline{r_{\mathbb{E}}}$
- only one (arbitrary) MPR solution is considered

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### **Measures**

- Running time: between 1.09s and 1.38s depending on  $\mathbb{D}$  and  $\mathbb{T}$  rates.
- Relative "over cost" of the real scenario α<sub>R</sub> compared to the MP one α<sub>P</sub>: due to homoplasy, parsimony can propose less events than really occurred.
- Accuracy Type I and II errors of  $\alpha_P$ 
  - we account for the correct tagging of G nodes
  - as well as for the branches of S involved in the events

Model of reconciliation

Algorithm

Experimental results

Conclusion

### Efficiency of parsimony according to costs



#### Over cost of real scenarios compared to MPR

- Small for all  $\mathbb D$  and  $\mathbb T$  rates
- Increases with the height of the gene trees
- Parsimony might be considered as a credible criterion to estimate reconciliations
   Great!

Model of reconciliation	Algorithm	Experimental results
0000	000000	0000000000

Conclusion

### Accuracy of parsimony to retrieve $\mathbb{D}$ events





- Reasonnably few forgotten duplications (due to homoplasy and non-unicity of the MPR?)
- \*Very\* few false positives

Not bad!

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### Accuracy of parsimony to retrieve $\mathbb{T}$ events



(from previous picture, some  $\mathbb{D}$  events taken for  $\mathbb{T}$  events)

Huh huh... :(

Model of reconciliation

Algorithm

Experimental results

Conclusion

#### Accuracy of parsimony to retrieve T events

A variant: looking only at the tagging of nodes in G, i.e. independently of the branches of S concerned. This is relevant for orthology prediction, with various applications such as function prediction.





Less dark picture, but still not satisfactory.

Model of reconciliation

Algorithm

Experimental results

### **Running times**

#### Comparison with an implementation of [Gorbunov et al 09] algorithm



Conclusion

#### Relationship between the MP and ML criteria



cost of MP reconciliation

### Further analyses required

#### To sum up:

- DP is much faster than previous implementations we had (from dozens minutes to less than 2 sec) [GORBUNOV ET AL 09]
- Computed parsimony cost fits nicely with real cost.
- Few duplications not recovered and almost no incorrect duplication predicted
- Transfers less correctly predicted on average ( $\approx 20 30\%$  errors)
- How do the error levels vary depending on the parts of *S* where events occur? (ancient vs recent events).
- Why are  $\mathbb{D}$  events better prediced than  $\mathbb{T}$  events?
  - Importance of the non-unicity of the MPR (not yet accounted for)?
  - Are relative costs for macro-events given to the DP too crude?
  - Mathematically more stringent definition of T (3 vs 2 coordintated to determine the event)?
  - Homoplasy: combinations of D + L interpreted as T events?

Model of reconciliation

Algorithm

Experimental results

Conclusion

### SylvX - a GUI to deal with reconciliations

We're currently designing a reconciliation viewer/editor allowing:

- Automatic computation and manual modifications of reconciliations
- Various graphical operators & analyses: re-rooting, zooming, filtering events



Model of reconciliation

Algorithm

Experimental results

Conclusion

### SylvX - a GUI to deal with reconciliations

We're currently designing a reconciliation viewer/editor allowing:

- Reusable interface to plug other reconciliation methods (ML, ...)
- Graphical computer-aided comparison of reconciliations (same *G* and *S*, varying *G*, varying *S*).



Model of reconciliation

Algorithm

Experimental results

Conclusion

### What next?

- Enumerating and counting the parsimonious reconciliations (dig for common events among MPRs)
- Deeper look at links btw MPR and ML reconciliations [DOYON ET AL 09]
- Estimating the rates of events
  - examining published results for different groups of species. This will allow us to propose reconciliations of real gene trees (can already be done this for Archaea).
  - · designing a method to estimate these rates from the data
- Allowing for polytomous G and S (as in Notung) [VERNOT ET AL 08]
- Using synteny information: to detect segmental events + account for local methodological problems in gene trees
- Mixing  $\mathbb{DTLS}$  with other reasons for incongruence btw G and S
- Consider other macro-events explicitely: gene conversion, deep coalescence, ...

Model of reconciliation

Algorithm

Experimental results

Conclusion

### Acknowledgment



Phylogenomics: integrated algorithms and visualizations for analyzing the evolution of life

#### http://www.lirmm.fr/phylariane/

#### Thanks

- Jean-Philippe Doyon, LIRMM, Montpellier (FR)
- Celine Scornavacca, Univ. Tuebingen (GE)
- Vincent Ranwez, ISE-M, Montpellier (FR)
- Gergely Szöllözi Eric Tannier & Vincent Daubin, L
- Vassily Lyubetsky and Kostya Gorbunov, Moscow
- · Jens and David for the invitation

#### Funding

Phylariane ANR project, Région LR, CNRS, ...