Joint reconstruction of ancestral gene orders and extant scaffolding

Cedric Chauve
Department of Mathematics
Simon Fraser University

Methodological issues from previous work

- We deal with fragmented extant genome assemblies that make a precise analysis of genome rearrangements history difficult.
  This motivated the creation of an *Anopheles* genome assemblies improvement group

- Analysis based on duplication-free methods, i.e. relying on one-to-one orthologous genes disregard more than half of the gene complement.

- Idea 1: joint ancestral genomes reconstruction / extant genomes scaffolding, by combining species-tree aware genome rearrangement parsimony and sequencing data.
  - RACA (Kim *et al*, PNAS (2013)), GOS-ASM (Aganezov *et al,* ISBRA (2016))
  - Different approach of traditional comparative scaffolding

- Idea 2: using reconciled gene family trees for the full gene complement with a joint gene evolution / genome evolution analysis method.
  Gene trees as an unused resource for genome rearrangements analysis.
  - El-Mabrouk & Sankoff (2000), Chauve *et al* (2013)

Available data

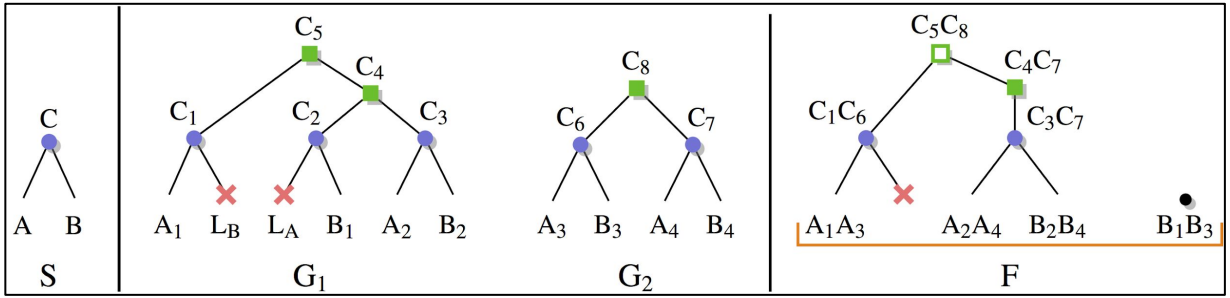
- We have imperfect (i.e. fragmented) genomes assemblies for our mosquito genomes.
- We have annotated genes and their positions along the fragments of the assemblies.
- We have a clustering of genes into homologous gene families. This clustering is imperfect but we rely on it.
- For each gene family we have a gene tree that describes the evolution of the genes in the family.
- We have a species tree that describes the evolution of the set of mosquito species we consider. We assume the phylogeny of the Gambiae complex is resolved.

Our goal is to combine all this data in order to be able to perform a thorough genome rearrangement study of this dataset, including the reconstruction of ancestral gene orders and to study the evolution along the species tree.

DeCo*: The DeCo algorithms family

The early ancestor: DeCo (Berard *et al*, Bioinformatics (2012)):

- ❑ Parsimony analysis of gene adjacency evolution in a phylogenetic context.
- ❑ Input: a species tree, two reconciled gene trees, list of (extant or ancestral) adjacencies
- ❑ Output: an adjacency evolution forest
- ❑ Optimization criterion: minimizing the cost of adjacencies gains / breaks (parsimony)



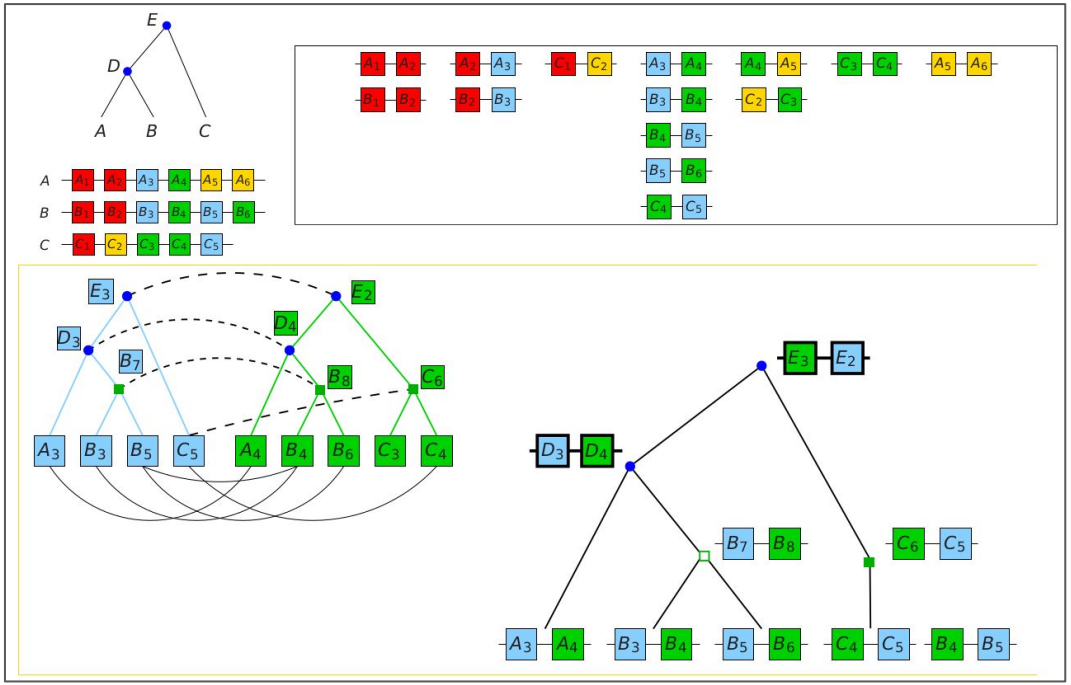
Technical features:

- ❑ Quadratic time Dynamic Programming (DP) algorithm (Sankoff-Rousseau type)
- ❑ Generalization of the SCJ SPP algorithm in a more general evolutionary model but limited to a single adjacency.
- ❑ Process adjacencies independently: the resulting ancestral gene orders can contain conflicts

DeCo*: The DeCo algorithms family

The early ancestor: DeCo (Berard *et al* (2012)).

- ❑ Parsimony analysis of gene adjacency evolution in a phylogenetic context.
- ❑ Computes a single optimal solution in a model where reconciled gene trees account only for gene duplication and gene loss.



DeCo*: The DeCo algorithms family

Technical features:

- ❑ Quadratic time Dynamic Programming (DP) algorithm (Sankoff-Rousseau type)
- ❑ Generalization of the SCJ SPP algorithm in a more general evolutionary model but limited to a single adjacency.

Case 5. $E(v^1)= Spec$ and $E(v^2)= Spec$.

Suppose without loss of generality that $S(ca(v^1))=S(ca(v^2))$ and $S(cb(v^1))=S(cb(v^2))$.

$$c_1(v^1,v^2)=\min\begin{cases} c_1(ca(v^1),ca(v^2))+c_1(cb(v^1),cb(v^2)) \\ c_1(ca(v^1),ca(v^2))+c_0(cb(v^1),cb(v^2))+C(Break) \\ c_0(ca(v^1),ca(v^2))+c_1(cb(v^1),cb(v^2))+C(Break) \\ c_0(ca(v^1),ca(v^2))+c_0(cb(v^1),cb(v^2))+2*C(Break) \end{cases}$$

$$c_0(v^1,v^2)=\min\begin{cases} c_0(ca(v^1),ca(v^2))+c_0(cb(v^1),cb(v^2)) \\ c_1(ca(v^1),ca(v^2))+c_0(cb(v^1),cb(v^2))+C(Gain) \\ c_0(ca(v^1),ca(v^2))+c_1(cb(v^1),cb(v^2))+C(Gain) \\ c_1(ca(v^1),ca(v^2))+c_1(cb(v^1),cb(v^2))+2*C(Gain) \end{cases}$$

Case 6. $E(v^1)= GDup$ and $E(v^2)= GDup$.

In this case $c_1\ (v^1,v^2)= min(D1,D2,D12)$ where $D1$ is the cost in the case the v^1 duplication comes first, $D2$ is the cost in the case the v^2 duplication comes first, $D12$ is the cost in the case where the v^1 and v^2 duplications are simultaneous.

$$\mathbf{D1}=\min\begin{cases} c_1(ca(v^1),v^2)+c_0(cb(v^1),v^2) \\ c_0(ca(v^1),v^2)+c_1(cb(v^1),v^2) \\ c_1(ca(v^1),v^2)+c_1(cb(v^1),v^2)+C(Gain) \\ c_0(ca(v^1),v^2)+c_0(cb(v^1),v^2)+C(Break) \end{cases}$$

$$\mathbf{D2}=\min\begin{cases} c_1(v^1,ca(v^2))+c_0(v^1,cb(v^2)) \\ c_0(v^1,ca(v^2))+c_1(v^1,cb(v^2)) \\ c_1(v^1,ca(v^2))+c_1(v^1,cb(v^2))+C(Gain) \\ c_0(v^1,ca(v^2))+c_0(v^1,cb(v^2))+C(Break) \end{cases}$$

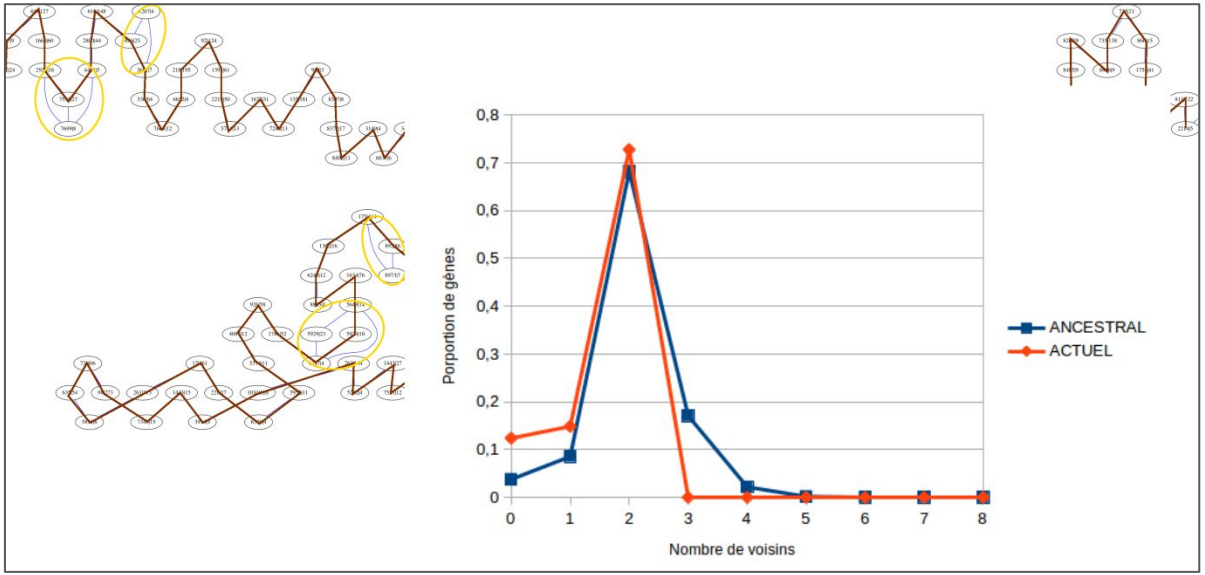
$\mathbf{D12}=\min$ (over all 16 following cases)

$$\begin{cases} (1)\ c_1(ca(v^1),ca(v^2)) + c_1(cb(v^1),cb(v^2)) + \\ \quad c_0(ca(v^1),cb(v^2)) + c_0(cb(v^1),ca(v^2)) \\ (2)\ c_1(ca(v^1),ca(v^2)) + c_1(cb(v^1),cb(v^2)) + \\ \quad c_0(ca(v^1),cb(v^2)) + c_1(cb(v^1),ca(v^2)) + \\ \quad C(Gain) \\ (3)\ c_1(ca(v^1),ca(v^2)) + c_1(cb(v^1),cb(v^2)) + \\ \quad c_1(ca(v^1),cb(v^2)) + c_0(cb(v^1),ca(v^2)) + \\ \quad C(Gain) \\ (4)\ c_1(ca(v^1),ca(v^2)) + c_1(cb(v^1),cb(v^2)) + \\ \quad c_1(ca(v^1),cb(v^2)) + c_1(cb(v^1),ca(v^2)) + \\ \quad 2*C(Gain) \\ (5)\ c_1(ca(v^1),ca(v^2)) + c_0(cb(v^1),cb(v^2)) + \\ \quad c_0(ca(v^1),cb(v^2)) + c_0(cb(v^1),ca(v^2)) + \\ \quad C(Break) \\ (6)\ c_1(ca(v^1),ca(v^2)) + c_0(cb(v^1),cb(v^2)) + \\ \quad c_0(ca(v^1),cb(v^2)) + c_1(cb(v^1),ca(v^2)) + \\ \quad C(Gain) \qquad + C(Break) \end{cases}$$

DeCo*: The DeCo algorithms family

Technical features:

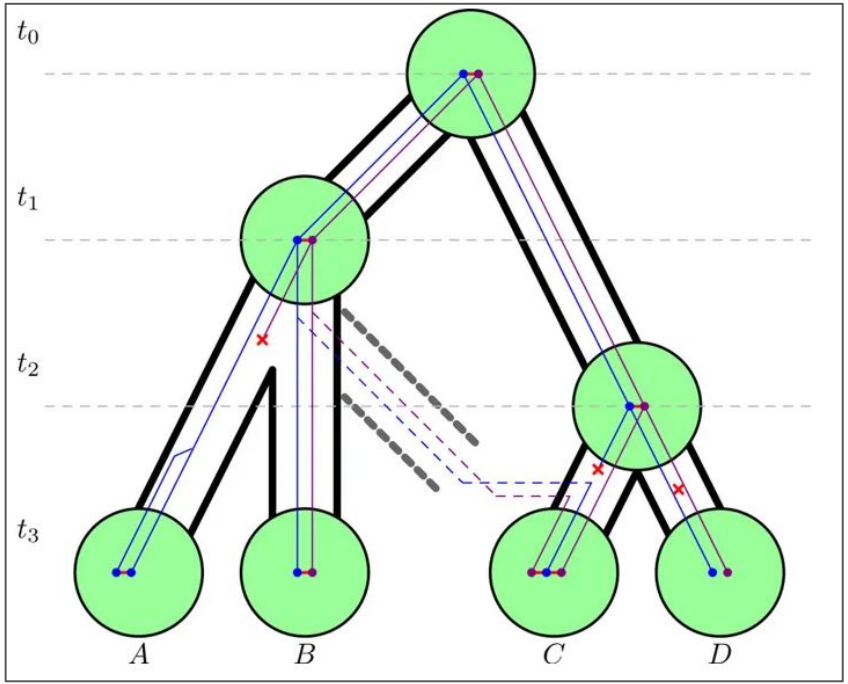
- ❑ Quadratic time Dynamic Programming (DP) algorithm (Sankoff-Rousseau type)
- ❑ Generalization of the SCJ SPP algorithm in a more general evolutionary model but limited to a single adjacency.
- ❑ Process adjacencies independently: the resulting ancestral gene orders can contain conflicts



DeCo*: The DeCo algorithms family

A first extension accounting for Lateral Gene Transfers : DeCoLT (Patterson *et al* (2013)).

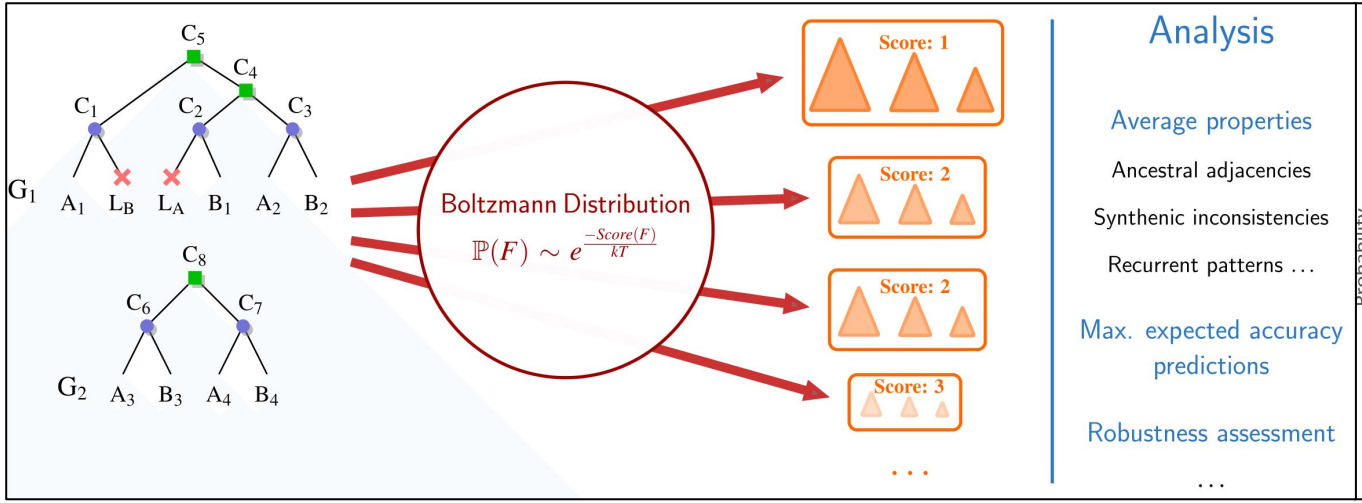
- If the species tree is fully dated, then one can obtain reconciled gene trees accounting for LGT
- The DP algorithm of DeCo can then be extended to account for the HGTs



DeCo*: The DeCo algorithms family

A second extension allowing to sample solutions : DeCLone (Zanetti *et al* (2015)).

- ❑ Use algebraic DP and the Boltzmann-Gibbs probabilistic approach to count/sample solutions
- ❑ Can sample optimal and/or sub-optimal solutions



Analysis

Average properties

Ancestral adjacencies

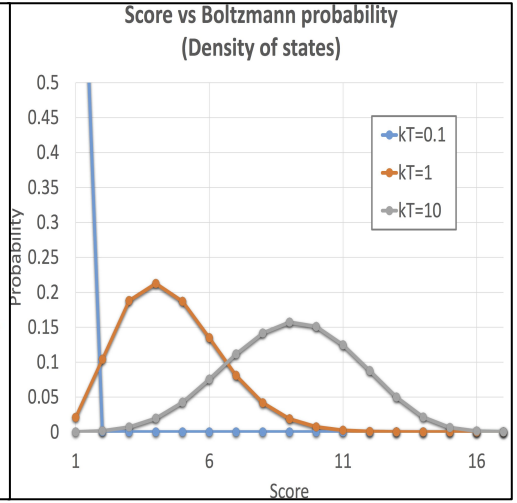
Synthenic inconsistencies

Recurrent patterns . . .

Max. expected accuracy predictions

Robustness assessment

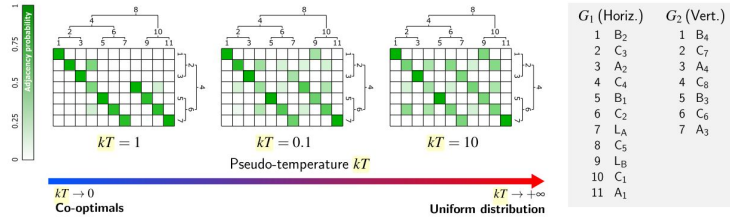
. . .



DeClone: a probabilistic framework

Details:

- ❑ Algebraic Dynamic Programming: one DP scheme fits them all
  - ❑ Sampling solutions under the Boltzmann distribution
  - ❑ Computing exact probabilities



G_1 (Horiz.)		G_2 (Vert.)	
1	B_2	1	B_4
2	C_3	2	C_7
3	A_2	3	A_4
4	C_4	4	C_8
5	B_1	5	B_3
6	C_2	6	C_6
7	L_A	7	A_3
8	C_5		
9	L_B		
10	C_1		
11	A_1		

Figure 3 Temperature dependency of the adjacency matrix. Each value of the (pseudo) temperature parameter kT induces different Boltzmann probabilities for ancestral and extant adjacencies, allowing to probe the continuum between a parsimonious model ($kT \to 0$) and the uniform distribution $kt \to \infty$.

- ❑ Quadratic time and space complexity
- ❑ Implementation is relatively easy but for numerical precision issues

1. $e(g_1) = \text{Extant}$ and $e(g_2) = \text{Extant}$:

$$c_1(g_1, g_2) = \begin{cases} \alpha & \text{if } g_1 g_2 \text{ is an adjacency} \\ 0 & \text{otherwise} \end{cases}$$

$$c_0(g_1, g_2) = \begin{cases} \alpha & \text{if } g_1 g_2 \text{ is not an adjacency} \\ 0 & \text{otherwise} \end{cases}$$

2. $e(g_1) = \text{GLoss}$ and $e(g_2) \in \{\text{Extant}, \text{Spec}, \text{GDup}\}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = \alpha^{\#\text{NonGDup}(g_2)}$$

3. $e(g_1) \in \{\text{Extant}, \text{Spec}, \text{GDup}\}$ and $e(g_2) = \text{GLoss}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = \alpha^{\#\text{NonGDup}(g_1)}$$

4. $e(g_1) = \text{GLoss}$ and $e(g_2) = \text{GLoss}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = \alpha$$

5. $e(g_1) \in \{\text{Extant}, \text{Spec}\}$ and $e(g_2) = \text{GDup}$:

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(g_1, b_{g_2}) \times c_0(g_1, a_{g_2}), & c_0(g_1, b_{g_2}) \times c_1(g_1, a_{g_2}), \\ c_1(g_1, b_{g_2}) \times c_1(g_1, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_0(g_1, b_{g_2}) \times c_0(g_1, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}} \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(g_1, b_{g_2}) \times c_0(g_1, a_{g_2}), & c_0(g_1, b_{g_2}) \times c_1(g_1, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, \\ c_1(g_1, b_{g_2}) \times c_0(g_1, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_1(g_1, b_{g_2}) \times c_1(g_1, a_{g_2}) \times e^{\frac{-2\mathsf{AG}}{kT}} \end{cases}$$

6. $e(g_1) = \text{GDup}$ and $e(g_2) \in \{\text{Extant}, \text{Spec}\}$:

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2), \\ c_1(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_0(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AB}}{kT}} \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2), \\ c_1(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_1(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2) \times e^{\frac{-2\mathsf{AG}}{kT}}. \end{cases}$$

7. $e(g_1) = \text{Spec}$ and $e(g_2) = \text{Spec}$:

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times \alpha, & c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}} \times \alpha, \\ c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}} \times \alpha, & c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-2\mathsf{AB}}{kT}} \times \alpha, \\ c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times \alpha, & c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}} \times \alpha, \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}} \times \alpha, & c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times e^{\frac{-2\mathsf{AB}}{kT}} \times \alpha \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times \alpha, & c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}} \times \alpha, \\ c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}} \times \alpha, & c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-2\mathsf{AG}}{kT}} \times \alpha, \\ c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times \alpha, & c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}} \times \alpha, \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}} \times \alpha, & c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times e^{\frac{-2\mathsf{AG}}{kT}} \times \alpha, \end{cases}$$

8. $e(g_1) = \text{GDup}$ and $e(g_2) = \text{GDup}$:

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2), \\ c_1(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_0(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AB}}{kT}}, \\ c_1(g_1, a_{g_2}) \times c_0(g_1, b_{g_2}), & c_0(g_1, a_{g_2}) \times c_1(g_1, b_{g_2}), \\ c_1(g_1, a_{g_2}) \times c_1(g_1, b_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_0(g_1, a_{g_2}) \times c_0(g_1, b_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}}, \\ c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}), & \\ c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-2\mathsf{AG}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}+\mathsf{AB}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}+\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}+\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}+\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}), & \\ c_0(a_{g_1}, a_{g_2}) \times c_1(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & \\ c_1(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_1(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_1(b_{g_1}, a_{g_2}) \times e^{\frac{-\mathsf{AB}}{kT}}, & \\ c_0(a_{g_1}, a_{g_2}) \times c_0(b_{g_1}, b_{g_2}) \times c_0(a_{g_1}, b_{g_2}) \times c_0(b_{g_1}, a_{g_2}) \times e^{\frac{-2\mathsf{AB}}{kT}}, & \end{cases}$$

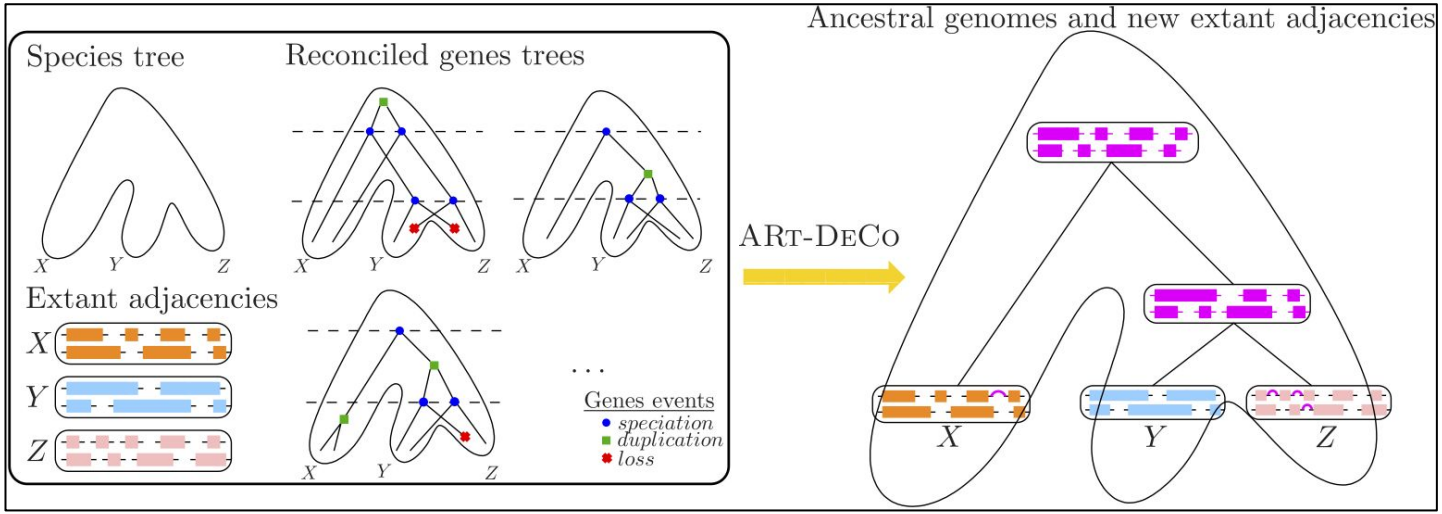
$$c_0(g_1, g_2) = \sum \begin{cases} c_0(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AG}}{kT}}, \\ c_1(a_{g_1}, g_2) \times c_0(b_{g_1}, g_2) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_1(a_{g_1}, g_2) \times c_1(b_{g_1}, g_2) \times e^{\frac{-2\mathsf{AG}}{kT}}, \\ c_0(g_1, a_{g_2}) \times c_0(g_1, b_{g_2}), & c_0(g_1, a_{g_2}) \times c_1(g_1, b_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, \\ c_1(g_1, a_{g_2}) \times c_0(g_1, b_{g_2}) \times e^{\frac{-\mathsf{AG}}{kT}}, & c_1(g_1, a_{g_2}) \times c_1(g_1, b_{g_2}) \times e^{\frac{-2\mathsf{AG}}{kT}}. \end{cases}$$

Zanetti, Ponty & Chauve, BMC Bioinformatics (2015)

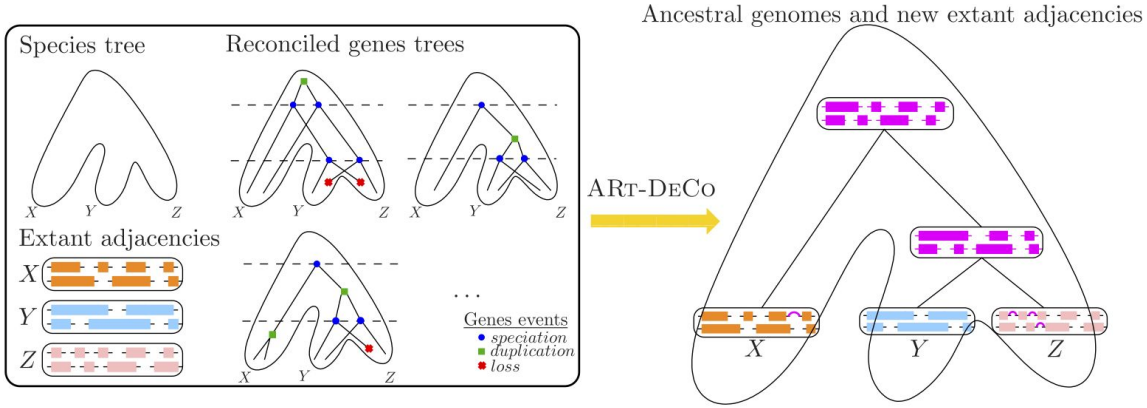
DeCo*: The DeCo algorithms family

A third extension allowing to refine extant assemblies : ArtDeCo - ADseq (Anselmetti *et al* (2015, 2018)).

- Idea: including in the model the gain of extant adjacencies
- Such adjacencies are scaffolding adjacencies that improve the extant assemblies
- This can be done by modifying only the base case of the DP scheme (Art-DeCo)
- The cost of creating a scaffolding adjacency can also be weighted by some sequencing signal based on sequencing data (ADseq).



ARt-DeCo / AD-Seq: Assembly Recovery through Detection of Co-evolution



- Base case of the DP DeCo algorithm:
  0 / 1 for the presence / absence of an adjacency
- ARt-DeCo: parsimony-supported scaffolding (extant) adjacencies
  log-probability of a potential scaffolding adjacency based on the expected karyotype
- ADseq: integrating sequencing data to handle species-specific adjacency breaks
  scaffolding probability based on paired reads and a scaffolding software
  (BESST, Sahlin, Chikhi & Arvestad, Bioinformatics (2016))

Anselmetti *et al*, BMC Genomics (2015)
Anselmetti *et al*, BMC Genomics (2018)

AD-Seq: Assembly Recovery through Detection of Co-evolution with Sequencing Data

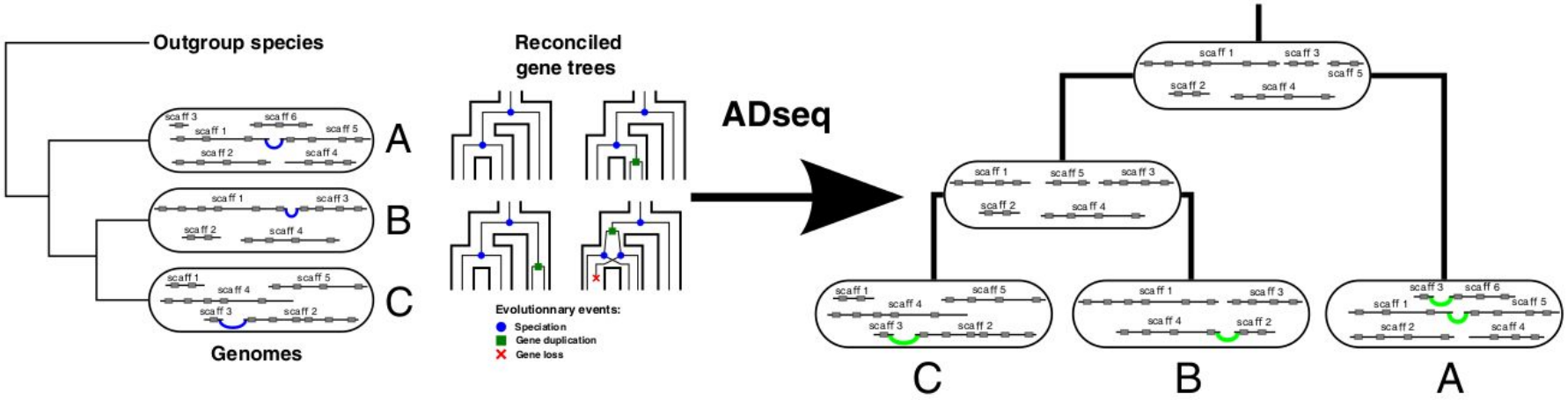
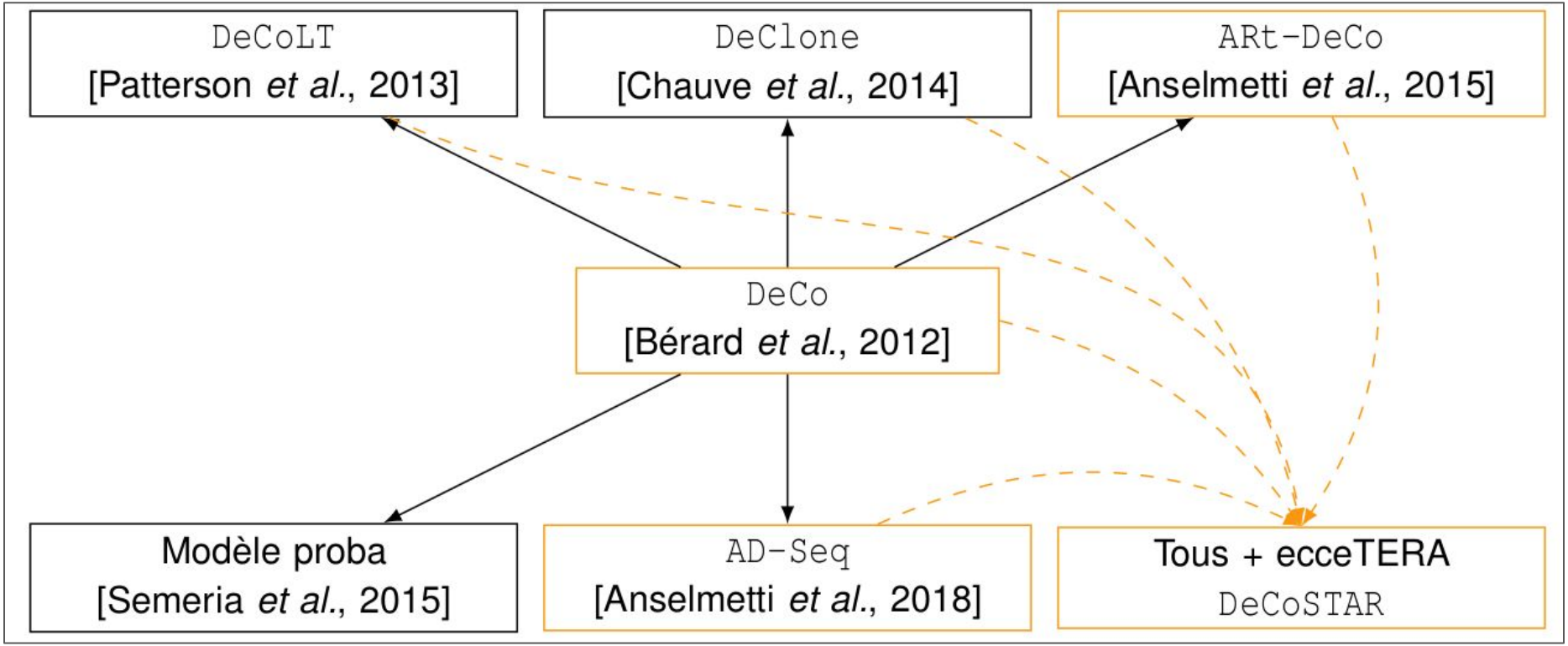


Fig. 1 Input and output of ADSEQ. (Left) Input data: (1) a species tree with extant genomes (A, B and C) containing observed adjacencies (black link) and scaffolding gene adjacencies with a *prior* score (blue link); each grey box represents a gene. (2) reconciled gene trees representing evolutionary histories of gene families annotated by evolutionary events. (Right) Typical output: gene order across ancestral and extant genomes including new extant gene adjacencies with a *posterior* score (green link) between genes located at fragments extremities in the initial genome assemblies

Anselmetti *et al*, BMC Genomics (2018)

DeCo*: The DeCo algorithms family



DeCoSTAR: Duchemin et al, GBE (2017)

Let’s put all that together

- We have imperfect (i.e. fragmented) genomes assemblies for our mosquito genomes.
- We have annotated genes and their positions along ghe fragments of the assemblies.
- We have a clustering of genes into homologous gene families. This clustering is imperfect but we rely on it.
- For each gene family we have a gene tree that describes the evolution of the genes in the family.
- We have a species tree that describes the evolution of the set of mosquito species we consider. We assume the phylogeny of the Gambiae complex is resolved.

We use the AD-Seq tool to jointly reconstruct ancestral gene orders and improve the scaffolding of extant genomes.

ProfileNJ: improved reconciled gene trees

Pitfall: Poor quality gene trees will impact negatively the whole pipeline.
Gene trees obtained through automatic sequence-based pipelines are error-prone.

Solution:ProfileNJ, efficient gene tree correction guided by species and synteny evolution.

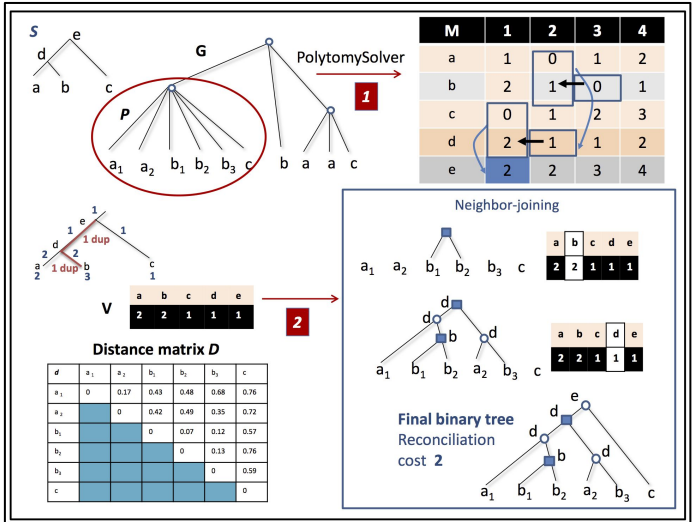


Figure 1. A species tree S and a multifurcated gene tree G. Each leaf x_i or x of G represents a gene belonging to genome x present as a leaf in S. Step (1) of ProfileNJ is PolytomySolver, which resolves each polytomy P of G independently. A dynamic programming table M is constructed. Step (2) of ProfileNJ takes as input a count vector V, here resulting from the backtracking path related by rectangles and arrows in table M, and a distance matrix d for the considered genes. A Neighbor joining (NJ) based procedure computes the gene tree in agreement with V that best reflects the distance matrix. The final completely refined tree is given bottom right. Duplication nodes are indicated by squares.

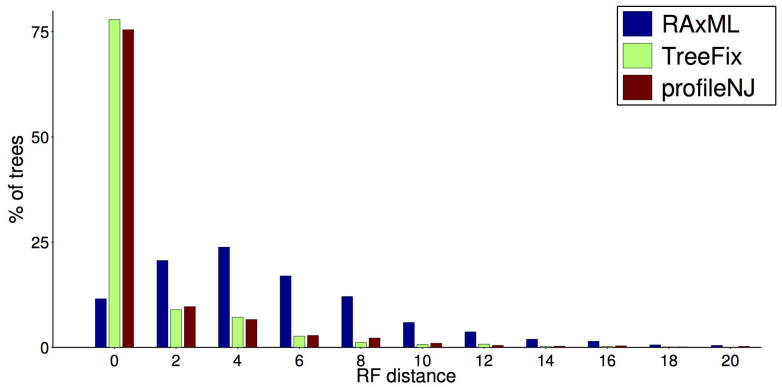


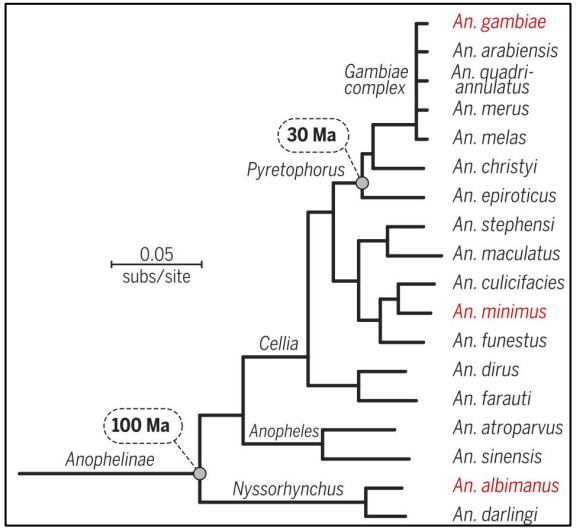
Figure 2. Topology accuracy of RAxML, TreeFix and ProfileNJ trees, mesured by RF distance with the true tree, on $\sim$ 2500 simulated trees from the fungal dataset. We use a sample of trees simulated under four different DL rate : $(1r_D$ - $1r_L)$, $(2r_D$ - $2r_L)$, $(4r_D$ - $4r_L)$ and $(4r_D$ - $1r_L)$. Percentage of reconstructed trees (y-axis) with a given RF distance (x-axis) to the true tree. TreeFix and ProfileNJ have a similar reconstruction accuracy (75% of trees match the true trees) while the input trees (RAxML) have the lowest accuracy. The graph is cut on the right, but contains more than 99% of the data.

Nouhati *et al*, PLoS One (2016)

Experiments: data and preprocessing

Input:

- 18 *Anopheles* genomes
- Fragmentation: from 6 to 8,600 contigs / scaffolds, mean 1,899, standard deviation 2,021
- 11,534 gene families trees with >= 3 genes and no overlapping genes: 172,585 genes
- HiSeq 2000 jump libraries with 3-5kb insert (contaminated with paired-end reads) for 13 genomes
- Scaffolding adjacencies / scores from BESST (Sahlin *et al*, (2016), customized “relaxed” version)

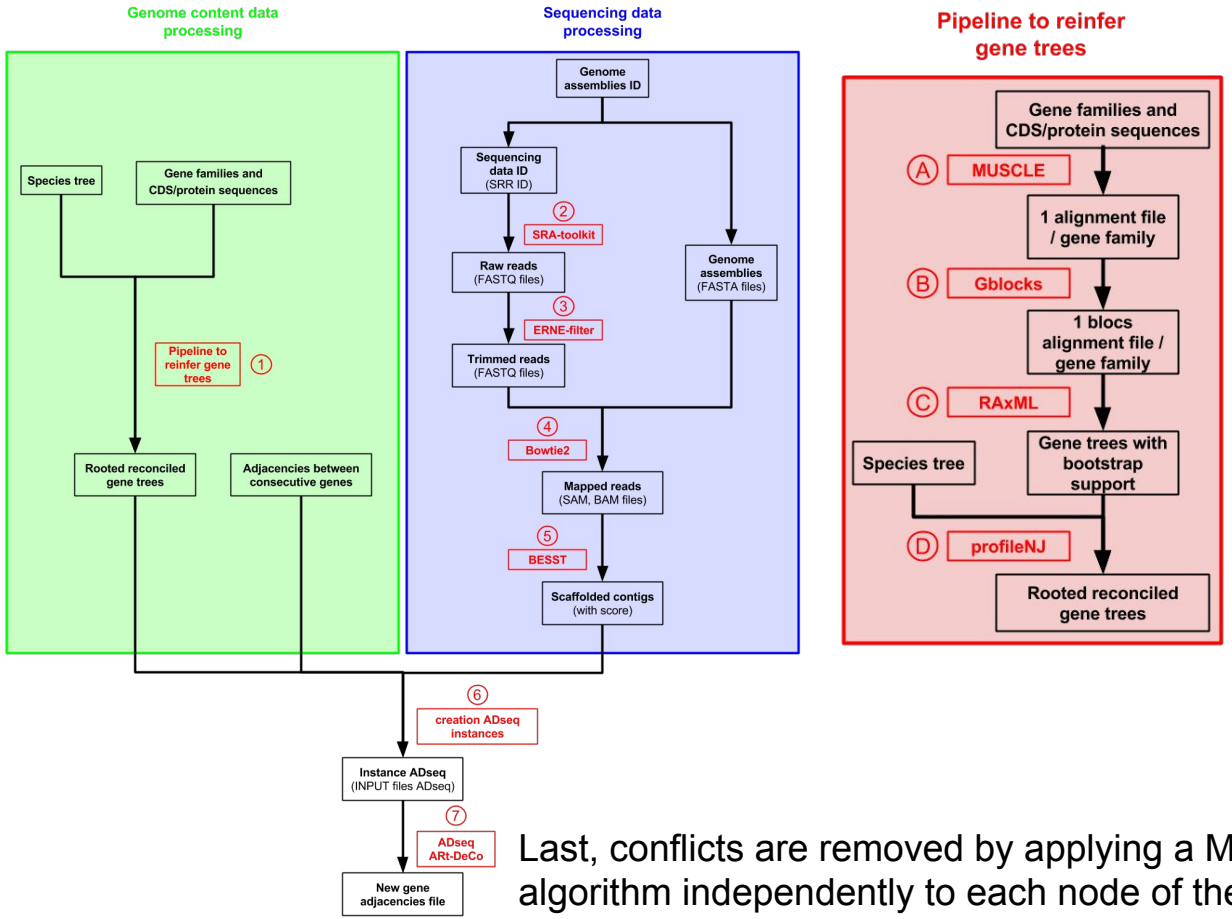


The *Anopheles* assemblies improvement group:
Igor Sharakhov, Nora Besansky, Max Alekseyev, Scott Emrich, Matthew Hahn, Adam Phillippy, Robert Waterhouse, Cedric Chauve

Integrating various comparative scaffolding methods to improve *Anopheles* genome assemblies.

Minisymposium, International Congress of Entomology 2016 (Orlando, Sept. 29, 2016).

Experiments: pipeline overview



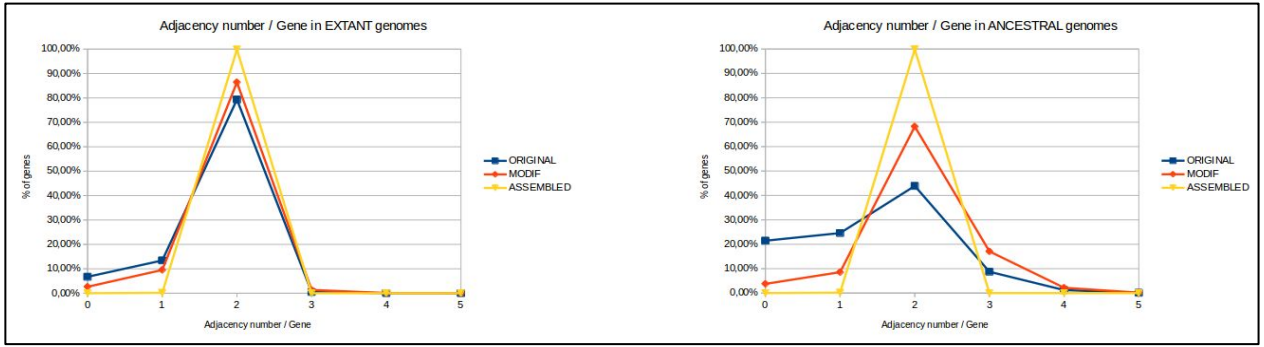
Last, conflicts are removed by applying a MWM algorithm independently to each node of the tree.

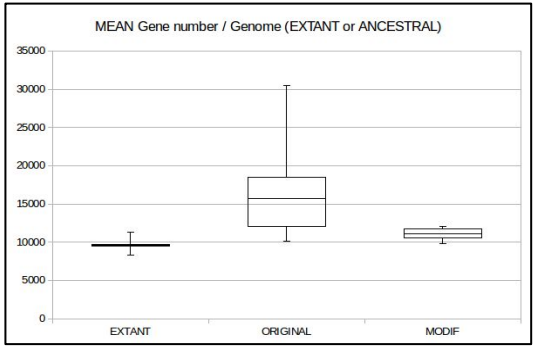
Anselmetti *et al*, BMC Genomics (2018)

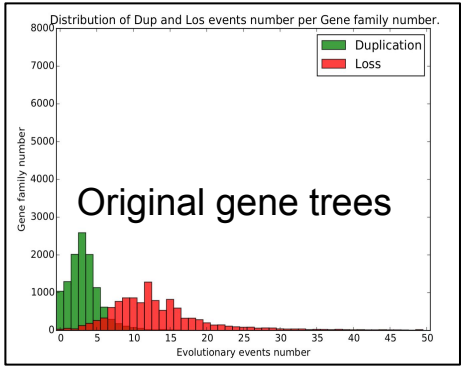
ProfileNJ results

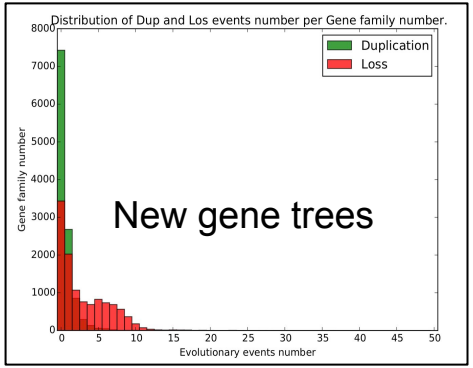
Improved gene trees:

- ❑ More realistic gene numbers for ancestral species
- ❑ Improved scaffolding adjacencies









The corrected gene trees lead to a larger number of new scaffolding adjacencies than the original trees (~22,000 vs ~10,000).

Anselmetti *et al*, BMC Genomics (2018)

AD-Seq: results on artificially fragmented genomes (principled fragmentation)

Simulation algorithm:

- De-novo assembly from raw Illumina reads (two runs: full reads set / half of the reads)
- Assembly software: Minia (conservative de Bruijn graph contig assembler, Chikhi & Rizk, WABI (2012))
- Mapping Minia contigs onto reference contigs to detect true adjacencies
- Merging Minia contigs that align uniquely on the same gene
- Computing scaffolding adjacencies with ADseq

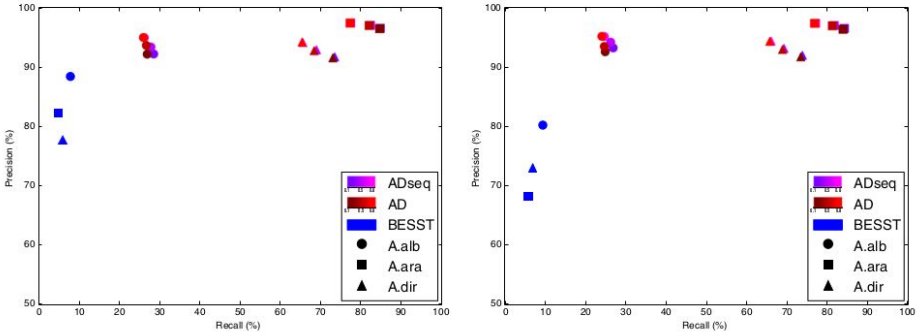


Fig. 2 Precision and recall statistics for scaffolding adjacencies on three artificially fragmented genomes (A.alb: *A. albimanus*, A.ara: *A. arabiensis* and A.dir: *A. dirus*). **Left graph:** results with 50% of reads. **Right graph:** results with all reads. The different methods results are plotted with the precision on the Y axis and the recall on the X axis. For ADSEQ and AD, results for three different adjacency support thresholds (0.1, 0.5 and 0.8) before genome linearization are plotted and represented with a color gradient. Note: A True Positive (TP) adjacency requires the proper orientation of both genes

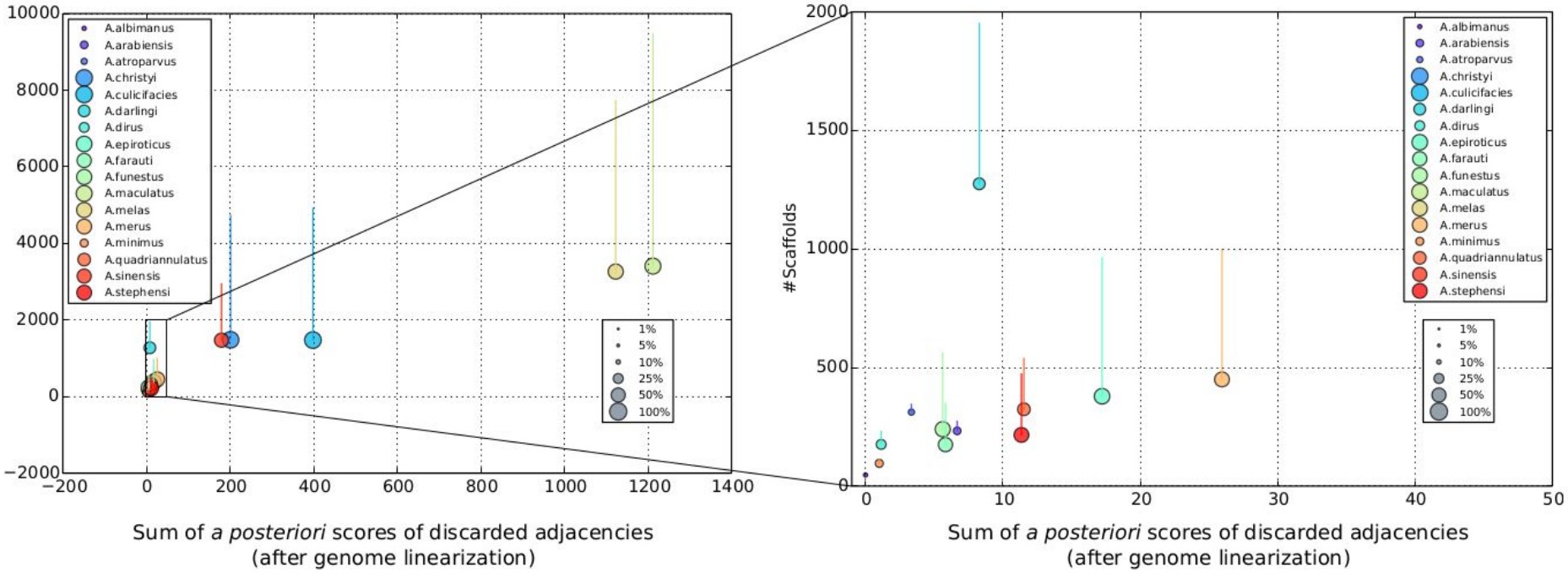
Anselmetti *et al*, BMC Genomics (2018)

AD-Seq / DeClone: extant genomes results

Species	Genes	Fragments	Adj: P>0.5 / BESST	Adj: P>0.9 / BESST	Adj: P>0.99 / BESST
A. arabiensis	9,742	237	105 / 19	71 / 12	66 / 12
A. quadriannulatus	9,725	480	290 / 0	245 / 0	234 / 0
A. melas	11,345	7,141	6,096 / 1,340	4,619 / 1,149	4,573 / 1,141
A. merus	9,745	961	710 / 385	612 / 330	588 / 317
A. christyi	8,456	4,595	3,561 / 728	3,201 / 698	3,153 / 693
A. epiroticus	9,495	925	710 / 307	610 / 276	583 / 264
A. stephensi	9,640	458	330 / 0	286 / 0	277 / 0
A. maculatus	9,553	8,600	6,390 / 540	5,755 / 507	5,718 / 507
A. culicifacies	10,364	4,596	4,384 / 1,667	3,809 / 1,556	3,767 / 1,546
A. minimus	9,420	102	34 / 6	30 / 6	27 / 5
A. funestus	9,655	534	397 / 97	354 / 87	324 / 87
A. dirus	9,513	213	82 / 11	72 / 10	71 / 10
A. farauti	9,545	331	202 / 0	182 / 0	179 / 0
A. atroparvus	9,547	314	52 / 14	44 / 12	42 / 12
A. sinensis	10,015	2,753	2,054 / 652	1,659 / 564	1,634 / 554
A. albimanus	8,712	49	0 / 0	0 / 0	0 / 0
A. darlingi	8,250	1,891	803 / 0	730 / 0	709 / 0

Anselmetti *et al*, BMC Genomics (2018)

AD-Seq / DeClone: extant genomes results



- The more a genome is fragmented, the more its scaffolding is improved

Anselmetti *et al*, BMC Genomics (2018)

AD-Seq / DeClone: extant genomes results

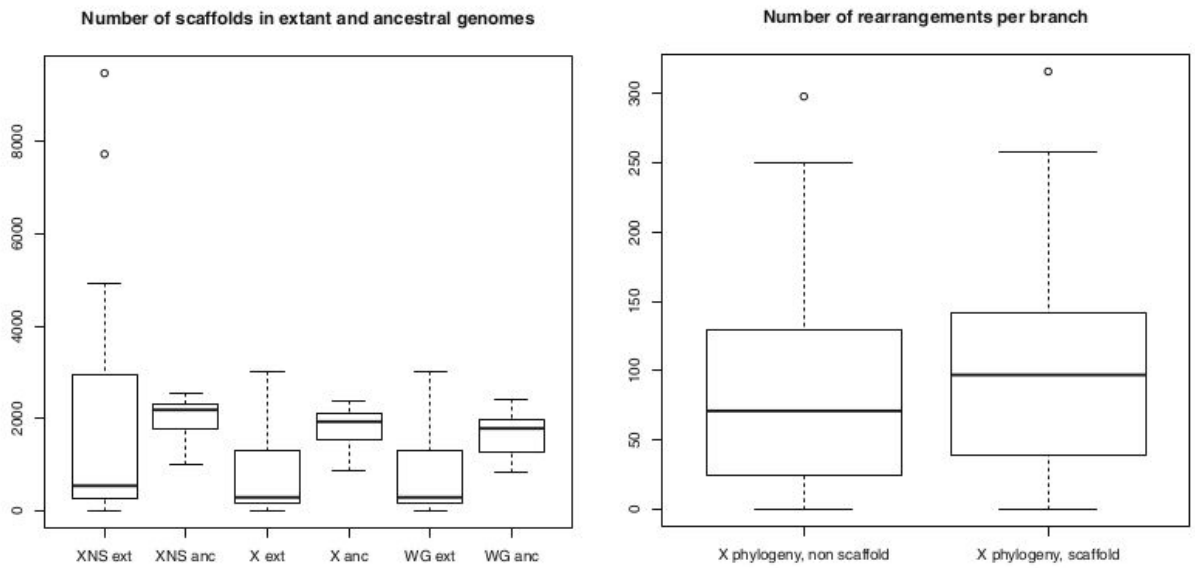
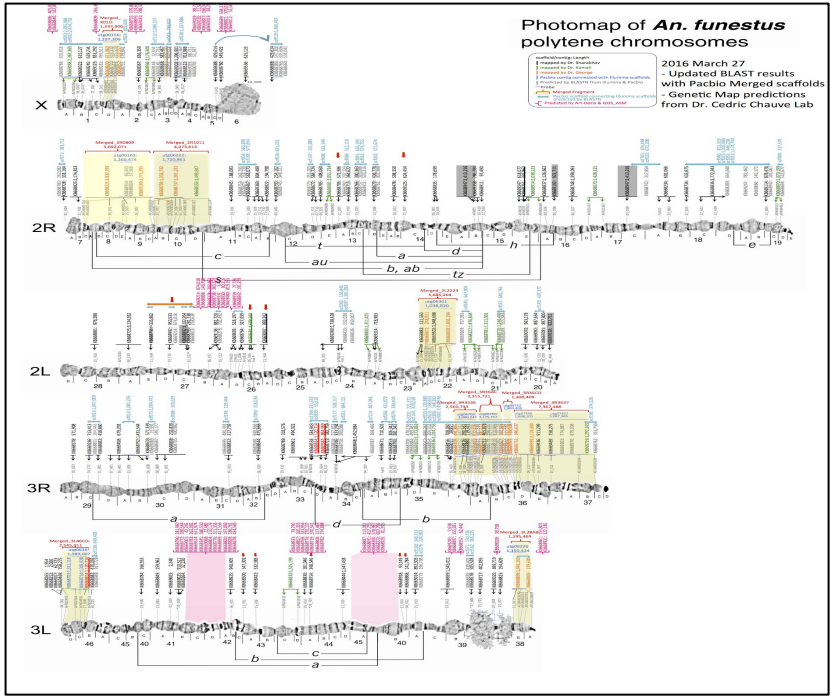


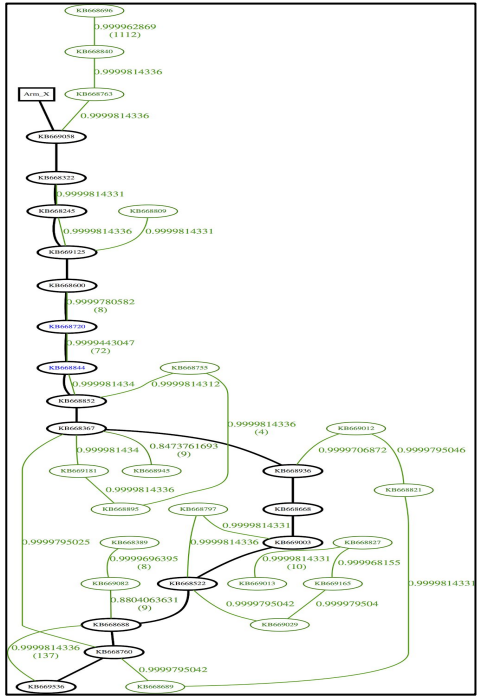
Fig. 4 Left: Number of segments in extant and ancestral genomes, according to three runs of ADSEQ in three different conditions. In the first run, we turn off the scaffolding mode on the X phylogeny, that is, it only constructs ancestral segments. The first column "XNS ext" thus describes the initial assembly, and "XNS anc" the assembly of ancestral genomes when reconstructed without extant scaffolding. In the second and third runs, the scaffolding mode was turned on, and run with the X phylogeny ("X ext" and "X anc") and the WG phylogeny ("WG ext" and "WG anc"). **Right:** Number of rearrangements over all branches of the X phylogeny, with and without the scaffolding mode

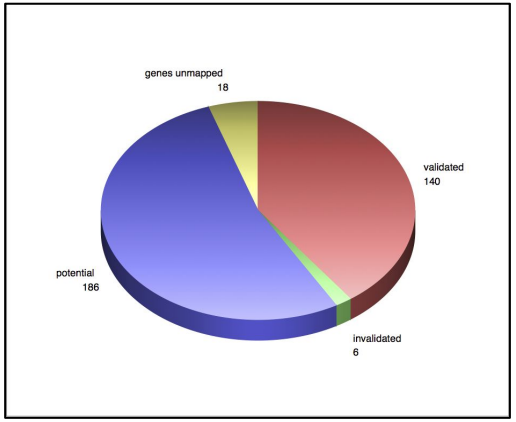
Anselmetti *et al*, BMC Genomics (2018)

Improving the *An. funestus* genetic map

- ❑ Initial assembly: 534 scaffolds containing at least one selected gene.
- ❑ ADseq: 350 scaffolding extant adjacencies predicted.
- ❑ 324 with a Boltzmann probability > 0.99 and only 87 supported by BESST.
- ❑ High agreement with a preliminary genetic map and a set of 3,773 preliminary PacBio scaffolds





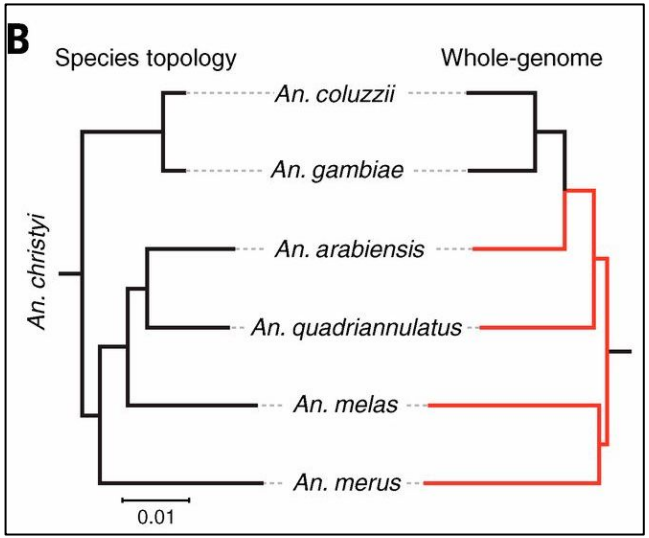


Unpublished

Different modes of evolution for autosomes and the X chromosome

Goal: Now we have improved extant assemblies and ancestral gene orders. We study evolution.

- ❑ Analysis of gene movement and genome rearrangement rates in *Anopheles* genomes to refine the hypothesis of differentiated evolutionary modes between the X chromosome and the autosomes.
- ❑ Accounting for the uncertainty in the phylogeny of the Gambiae complex



Methodological issues:

1. How to decide if a gene is an X-gene or an autosomal gene with highly fragmented genome assemblies?
   It is obvious with one-to-one orthologous genes, not with gene families subject to gene duplication.

2. Same question for genome segments (contigs / CARs).

3. Accounting for introgression.

Anselmetti *et al*, BMC Genomics (2018)

Differentiated evolution of autosomes and the X chromosome: gene transferts

Assigning genes / CARs to the categories X / autosomal.

1. Gene: Detecting *A. gambiae* orthologs using gene trees
   Computing a proba. dist. from their location
   Ex: 2 orthologs on X and 1 on arm 2L: *P(X)=0.66* and *P(2L)=0.33.*

2. Contigs / scaffolds: averaging the probability of the genes it contains.
   New adjacencies: only 6 (out of 23,405) inconsistencies joining two CARs with different proba. dist.

3. Assign to each extant gene the proba. dist. of its contig / scaffold.
   Confirmation of the accuracy of gene assignment by looking at an improved assembly of *An. albimanus*.

4. Repeat for ancestral CARs.

Results: few gene moves: 469 aut-2-X, 429 X-2-aut

Anselmetti *et al*, BMC Genomics (2018)

Differentiated evolution of autosomes and the X chromosome: genome rearrangements

Estimating genome rearrangements rates:

1. Assign each gene family to a category X / autosomal as described before
2. Analyze ADseq parsimonious evolutionary scenarios (~19,000 instances with both phylogenies) for adjacencies between pairs of autosomal genes or pairs of X genes.

Results indicate a compartmentalized evolution

1. Genes events on autosomes support the WG phylogeny: less duplications.
2. Syntenic conflicts support the WG phylogeny (6319 vs. 7665).
3. Genome rearrangements support slightly the WG phylogeny vs. the X phylogeny.

Table 1 Numbers of inferred rearrangements and duplications in the X chromosome and in the autosomes, according to the phylogeny (X or WG) used as a parameter of ADSEQ

Event	X phylogeny		WG phylogeny	
	X chr.	Autosomes	X chr.	Autosomes
Duplications	604	5857	606	5553
Rearrangements	415	2949	416	2760

But no clear support for a differentiated rearrangement rate or a phylogeny.

Anselmetti *et al*, BMC Genomics (2018)

Summary of the AD-Seq analysis

Starting point:

- ❑ A “classical” large-scale data set of general interest, a large multi-groups project
- ❑ Fragmented genome assemblies (1 fully assembled genome)
- ❑ Limited but non-trivial evolution: 100M years, introgression, uncertain species phylogeny

Methodology:

- ❑ DeCo*, a family of algorithms for synteny evolution in a parsimonious framework
- ❑ Joint phylogenetic scaffolding / ancestral reconstruction using reconciled gene trees
  An attempt to integrate evolutionary analysis at various levels (genes, genomes) and sequencing data.

Anopheles resources/results:

- ❑ Improved gene trees
- ❑ Improved genomes assemblies (*A. funestus* as an illustration)
- ❑ Ancestral genomes maps
- ❑ Confirmation of different evolutionary regime X/autosomes, but question mark about a highly different rearrangement rate or a specific phylogeny.

The next step: assembly improvement consortium

- The work done using AD-Seq was done by our group only.
- The *Anopheles* assembly improvement group included three groups working on comparative scaffolding (our group AD-Seq, Max Alekseyev group GOS-ASM, Rob Waterhouse group OrthoStitch).
- Each group developed its own method, that filtered different amount of data and relied on different principles.

Our goal was to combine the results of the three groups in order to provide more robust improved *Anopheles* genomes assemblies.

Evolutionary superscaffolding and chromosome anchoring to improve *Anopheles* genome assemblies.
BMC Biology (2020)

Method

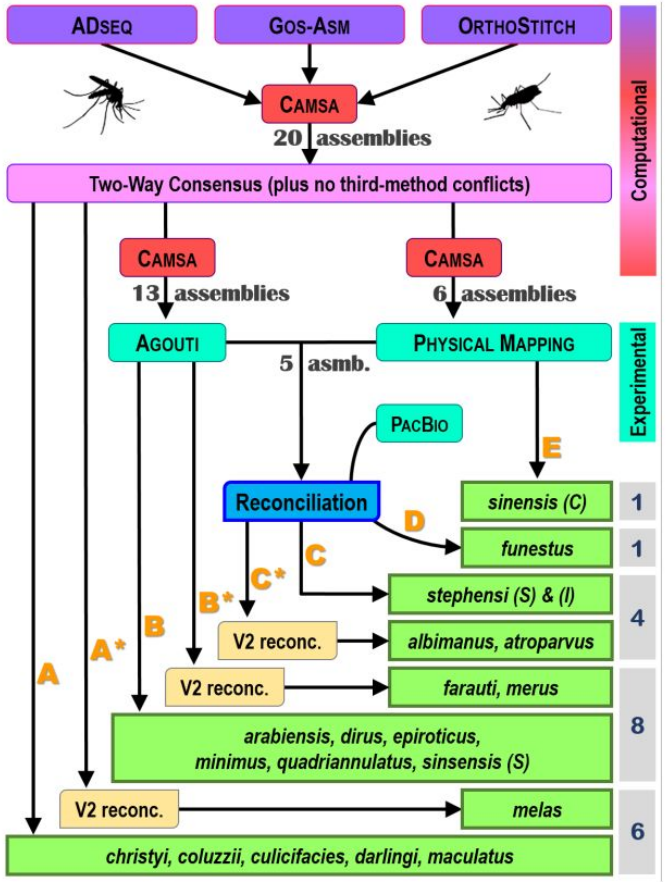


Figure S1. Workflows applied to upgrade the 20 anopheline assemblies

A: two-way synteny only. B: two-way synteny and AGOUTI. C: two-way synteny, AGOUTI, and physical mapping data. D: two-way synteny, AGOUTI, physical mapping data, and PacBio sequencing data. E: two-way synteny and physical mapping data. Asterisks (*) indicate additional reconciliation with version 2 assemblies (V2 reconc.) for a subset of species.

Results

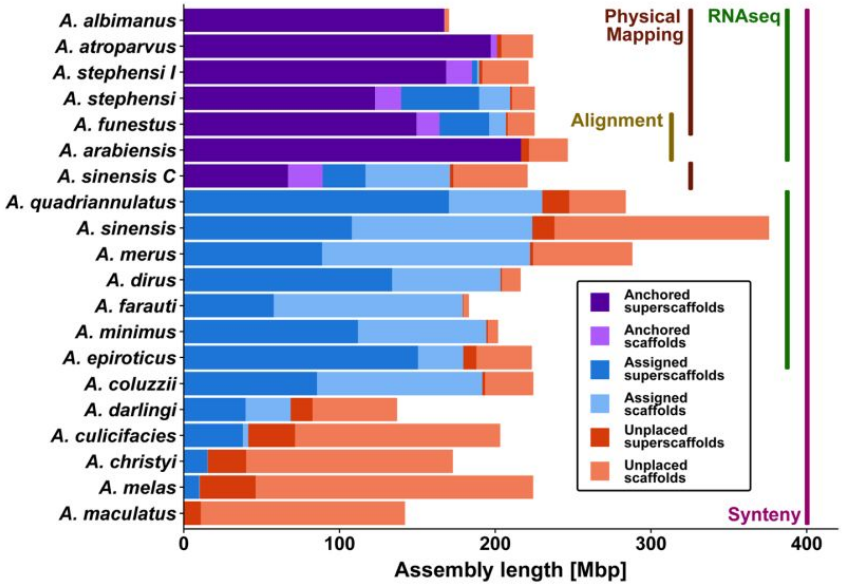
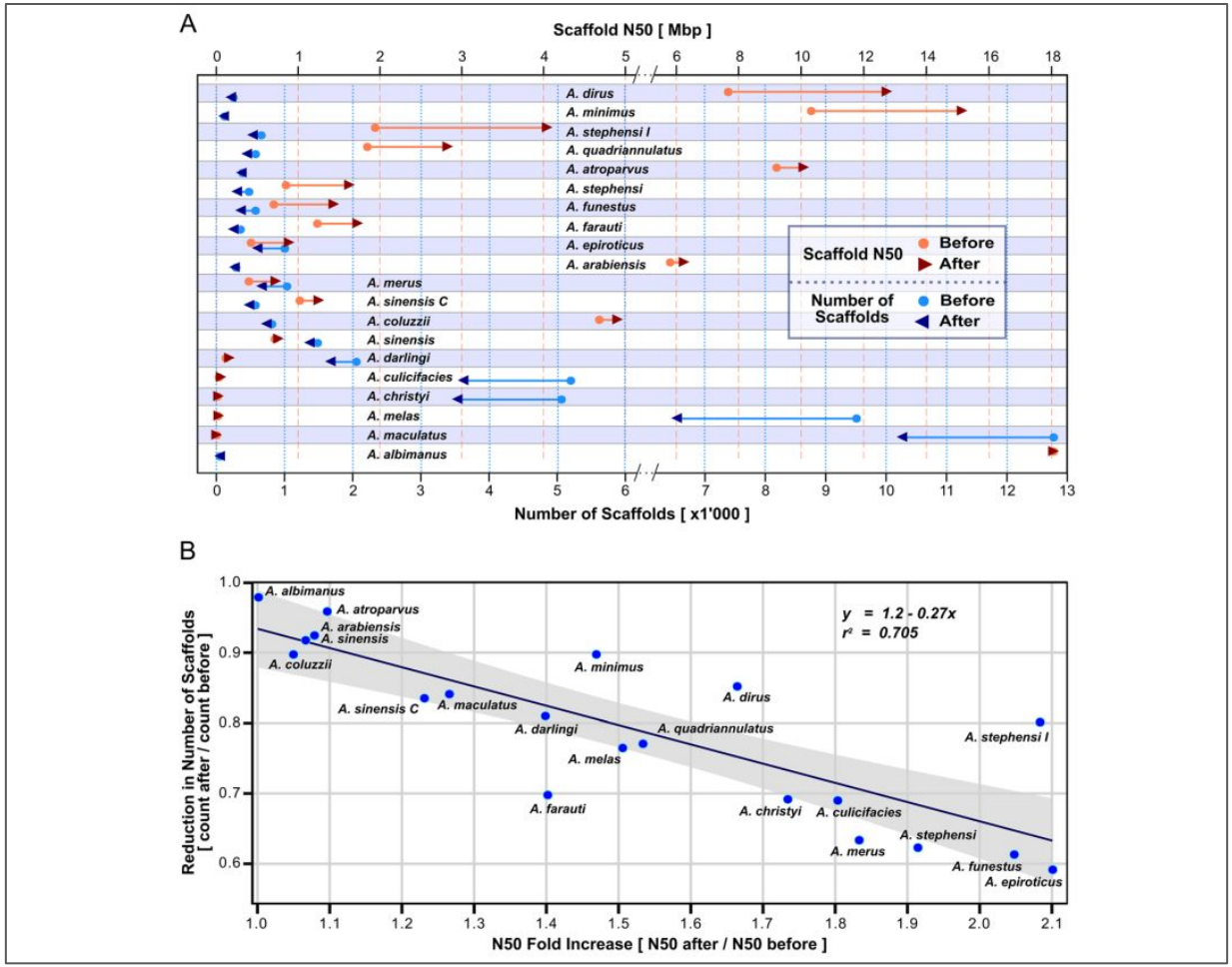
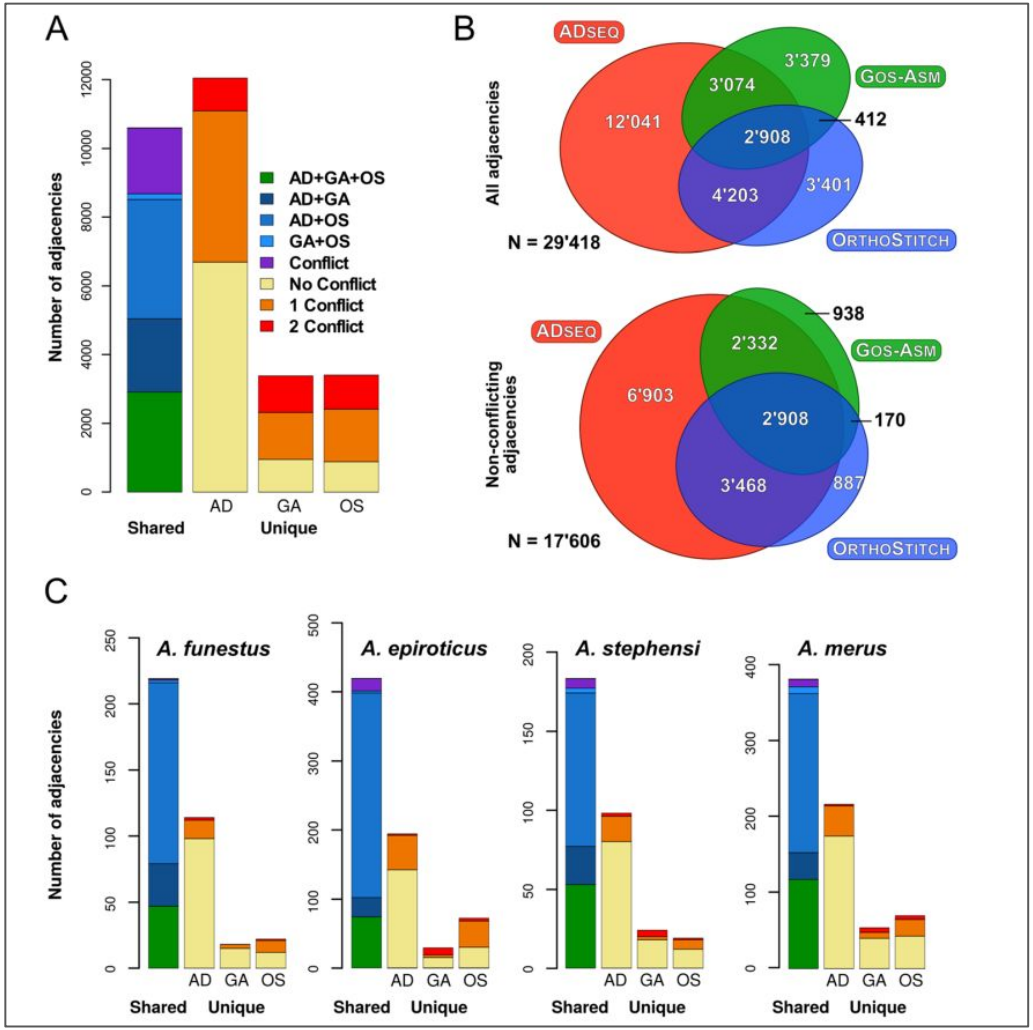


Fig. 1 Genomic spans of scaffolds and superscaffolds with and without chromosome anchoring or arm assignments for 20 improved *Anopheles* assemblies. Consensus gene synteny-based methods were employed across the 21-assembly input dataset (also including *Anopheles gambiae*) to delineate scaffold adjacencies and build new superscaffolded assemblies with improved contiguities. These were integrated with results from additional complementary approaches for subsets of the anophelines including transcriptome (RNAseq) and genome sequencing data, whole genome alignments, and chromosome anchoring data from physical mapping of probes. Chromosome mapping data for 7 assemblies enabled anchoring of superscaffolds and scaffolds to their chromosomal locations (purple colours). Enumerating shared orthologues further enabled the assignment of non-anchored superscaffolds and scaffolds to chromosome arms (blue colours). Unplaced superscaffolds and scaffolds (orange colours) still comprise the majority of the least contiguous input assemblies, but they make up only a small proportion of the assemblies for which the available data allowed for substantial improvements to assembly contiguity and/or anchoring and/or arm assignments. Results for two strains are shown for *Anopheles sinensis*, SINENSIS and Chinese (C), and *Anopheles stephensi*, SDA-500 and Indian (I)

Results


A
Scaffold N50 [Mbp]
0 1 2 3 4 5 6 8 10 12 14 16 18
A. dirus
A. minimus
A. stephensi I
A. quadriannulatus
A. atroparvus
A. stephensi
A. funestus
A. farauti
A. epiroticus
A. arabiensis
A. merus
A. sinensis C
A. coluzzii
A. sinensis
A. darlingi
A. culicifacies
A. christyi
A. melas
A. maculatus
A. albimanus
Scaffold N50
Before
After
Number of Scaffolds
Before
After
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Number of Scaffolds [x1'000]
B
Reduction in Number of Scaffolds [count after / count before]
1.0 0.9 0.8 0.7 0.6
y = 1.2 - 0.27x
r² = 0.705
A. albimanus
A. atroparvus
A. arabiensis
A. sinensis
A. coluzzii
A. minimus
A. dirus
A. sinensis C
A. maculatus
A. darlingi
A. quadriannulatus
A. stephensi I
A. melas
A. farauti
A. christyi
A. culicifacies
A. stephensi
A. merus
A. funestus
A. epiroticus
1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1
N50 Fold Increase [N50 after / N50 before]

Results


A
Number of adjacencies
0
2000
4000
6000
8000
10000
12000
AD+GA+OS
AD+GA
AD+OS
GA+OS
Conflict
No Conflict
1 Conflict
2 Conflict
AD
GA
OS
Shared
Unique
B
All adjacencies
ADSEQ
GOS-ASM
ORTHOSTITCH
12'041
3'074
3'379
2'908
412
4'203
3'401
N = 29'418
Non-conflicting adjacencies
ADSEQ
GOS-ASM
ORTHOSTITCH
6'903
2'332
938
2'908
170
3'468
887
N = 17'606
C
Number of adjacencies
A. funestus
A. epiroticus
A. stephensi
A. merus
AD
GA
OS
Shared
Unique

Summary of the consortium work

Starting point:

- A “classical” large-scale data set of general interest, a large multi-groups project
- Fragmented genome assemblies (1 fully assembled genome)
- Limited but non-trivial evolution: 100M years, introgression, uncertain species phylogeny

Methodology:

- Three synteny-based scaffolding methods, combined through a stringent consensus algorithm
- Additional data (RNA-seq, PacBio, genome maps)

Results:

- Significantly improved genome assemblies, made available to the community through VectorBase
- The three synteny methods do generate quite different results