The infinite sites model of genome evolution

Jian Ma*, Aakrosh Ratan†, Brian J. Raney*, Bernard B. Suh*, Webb Miller†, and **David Haussler***

2008 - PNAS

* Center for Biomolecular Science and Engineering, University of California
† Center for Comparative Genomics and Bioinformatics, Pennsylvania State University

The problem

Recovering the evolutionary history of a set of genomes that are related to an unseen common ancestor genome

Operations: speciation, rearrangements, deletion, insertion, duplication.

Genomes: linear and circular

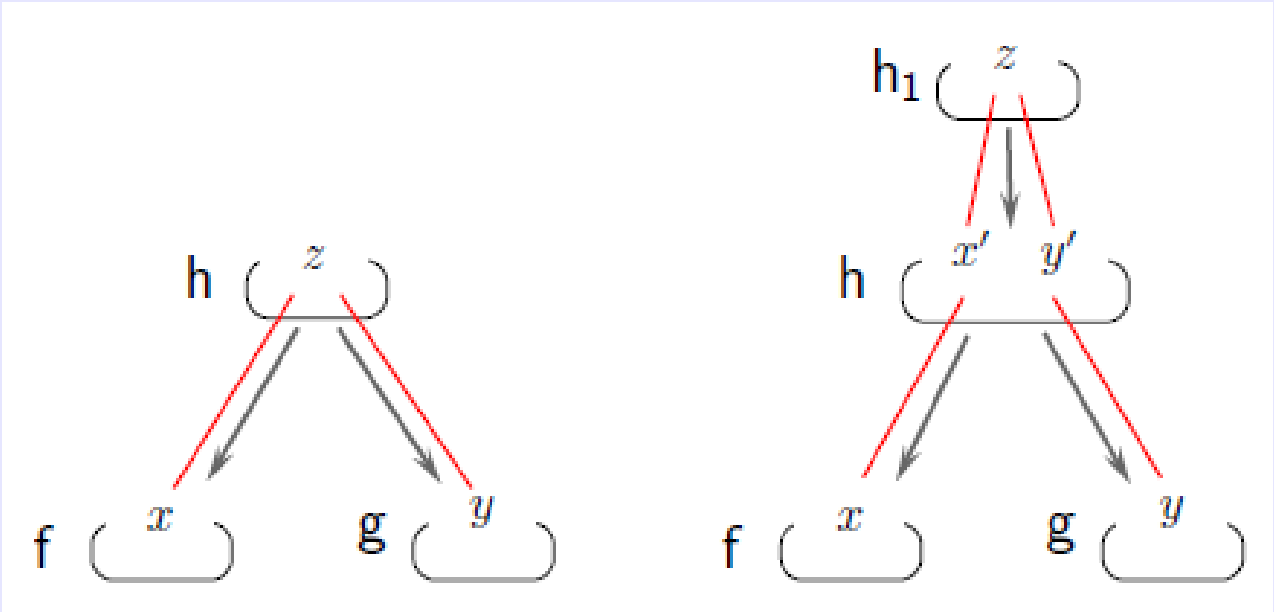
=> Polynomial-time algorithm to find the most **parsimonious** evolutionary history in a specific model.

Model

Infinity sites model: No breakpoint is used twice.

Evolutionary distance: The substitution rate is the same for all sites in a species, but is allowed to vary between species.

Parsimonious: Number of rearrangements, speciation and duplication

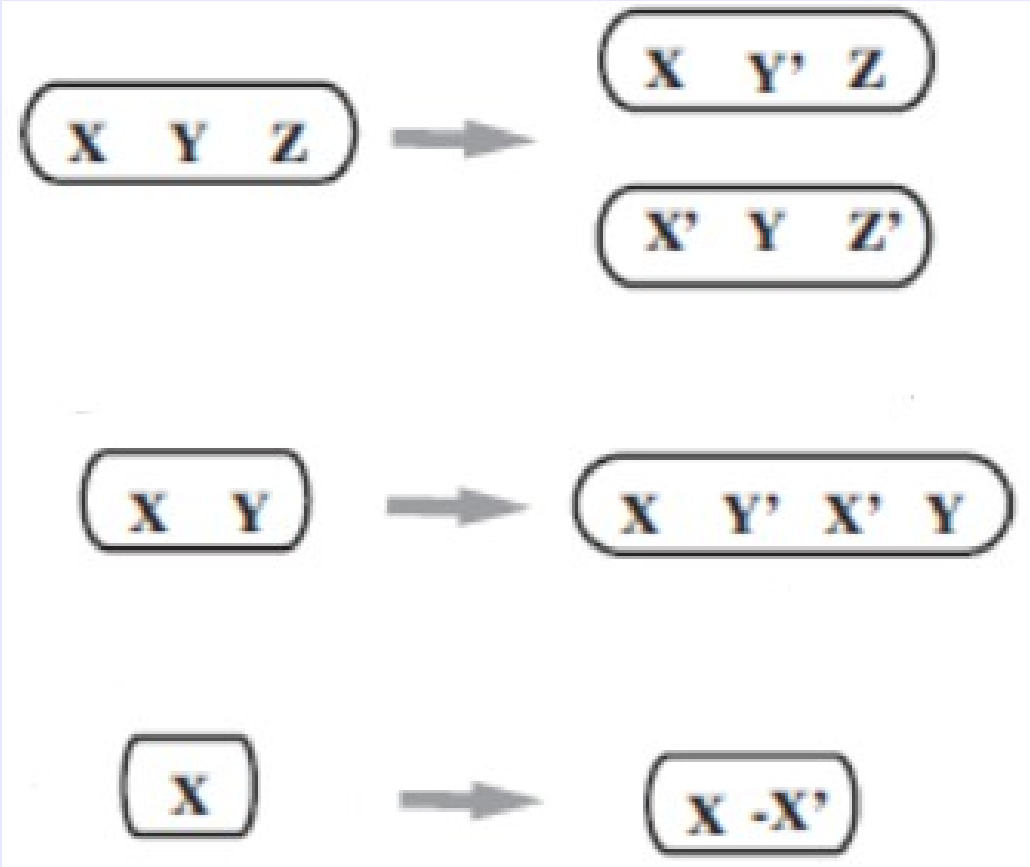


Speciation Duplication then speciation

Some operations

Operations

- Duplications:



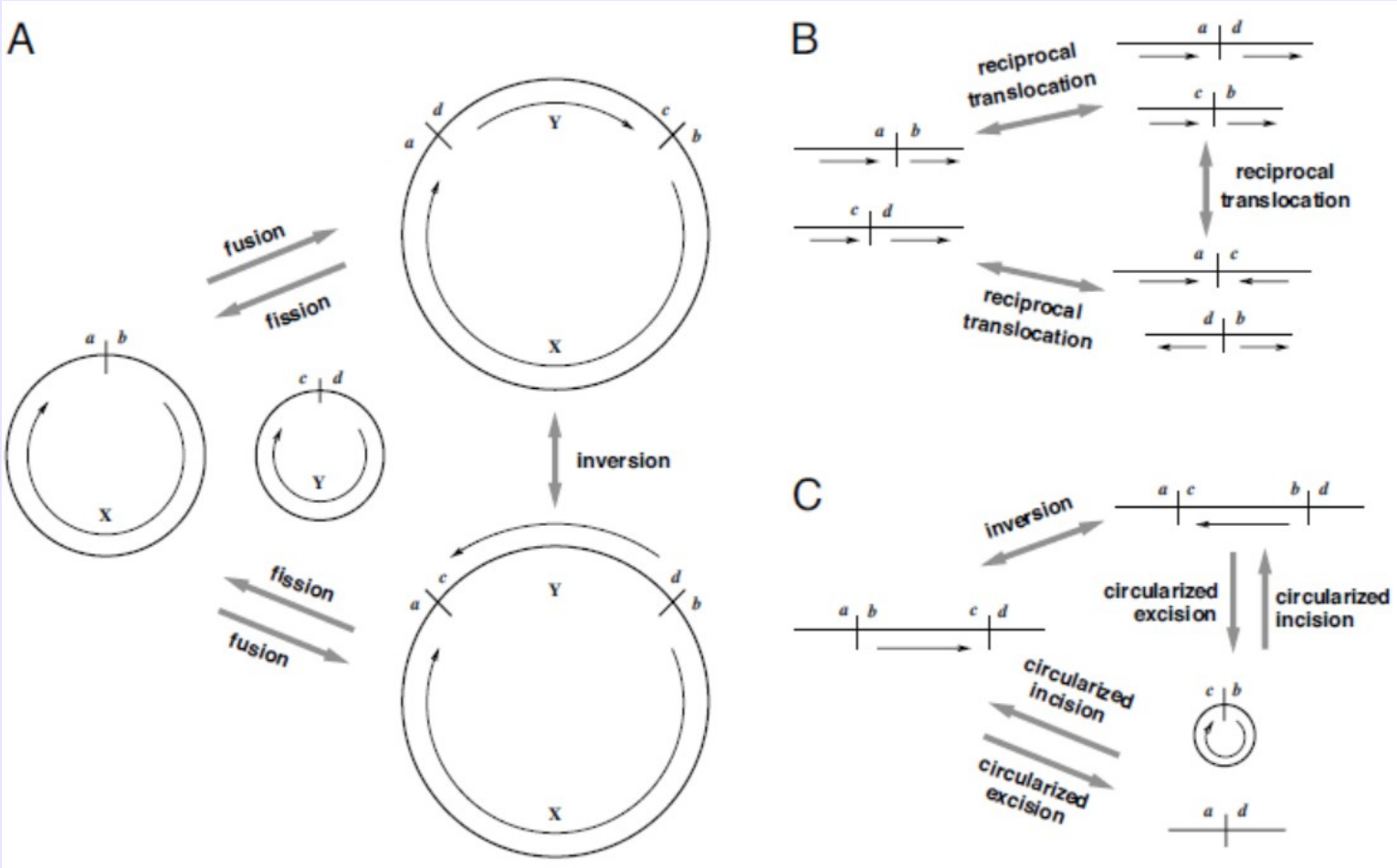
Chromosome duplication

Tandem duplication

Reverse tandem duplication

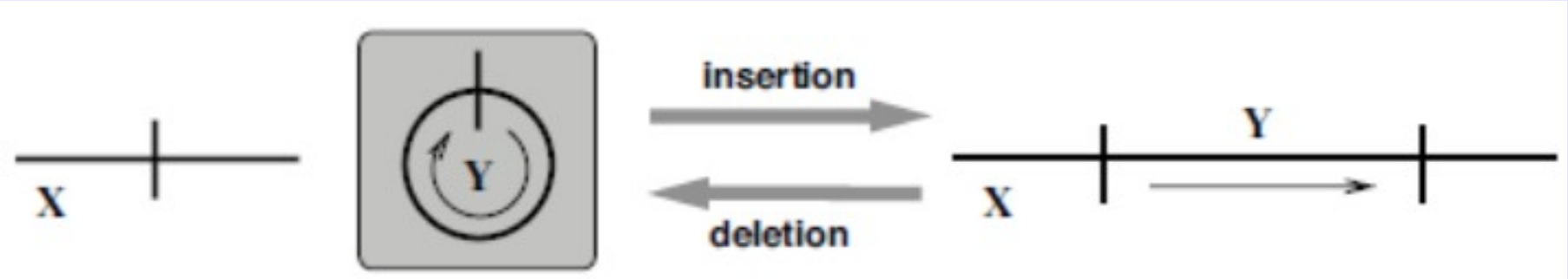
Operations

- Duplications
- Rearrangements:
  - 2-breakpoints (inversion, fusion, reciprocal translocation, ...)

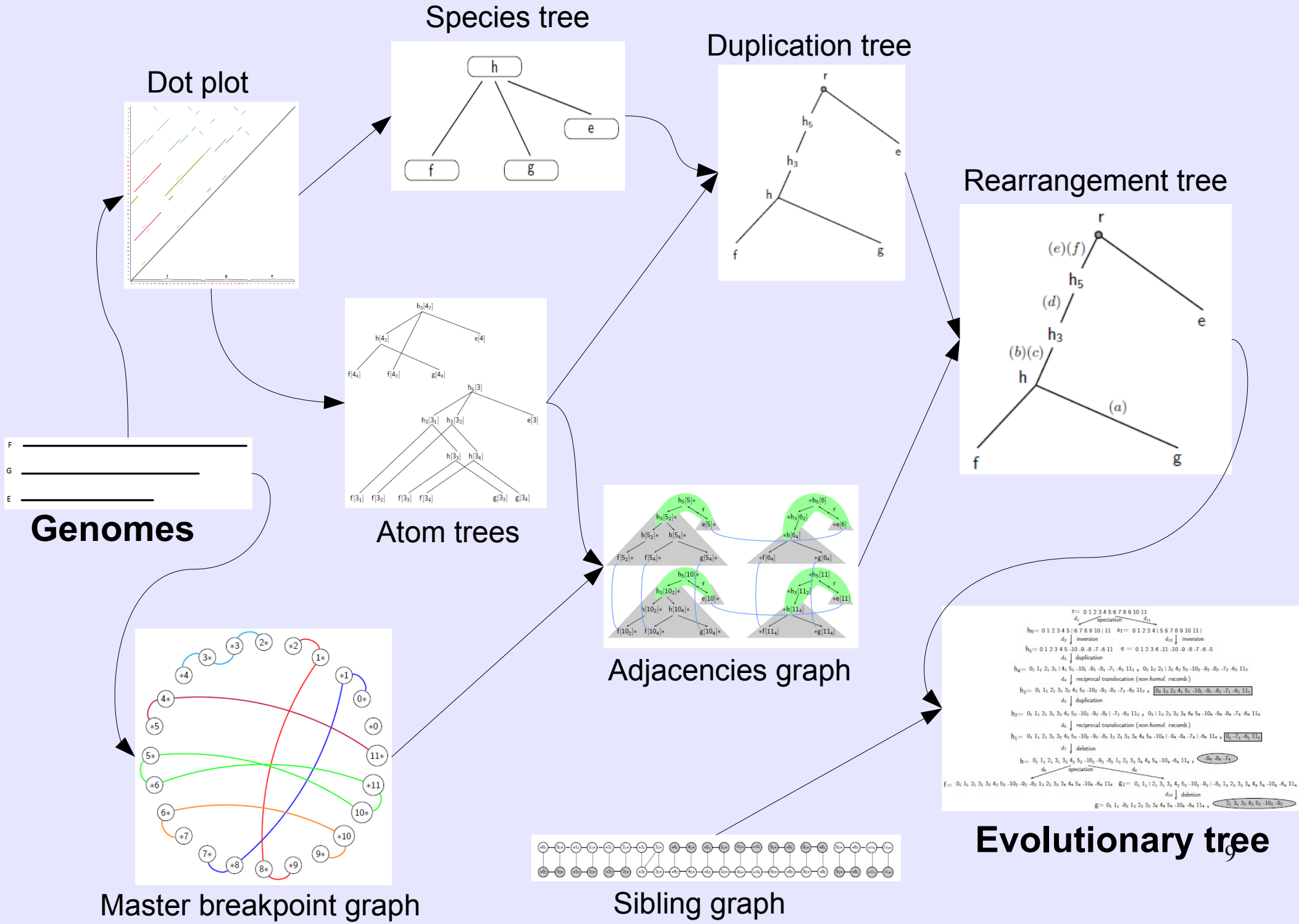


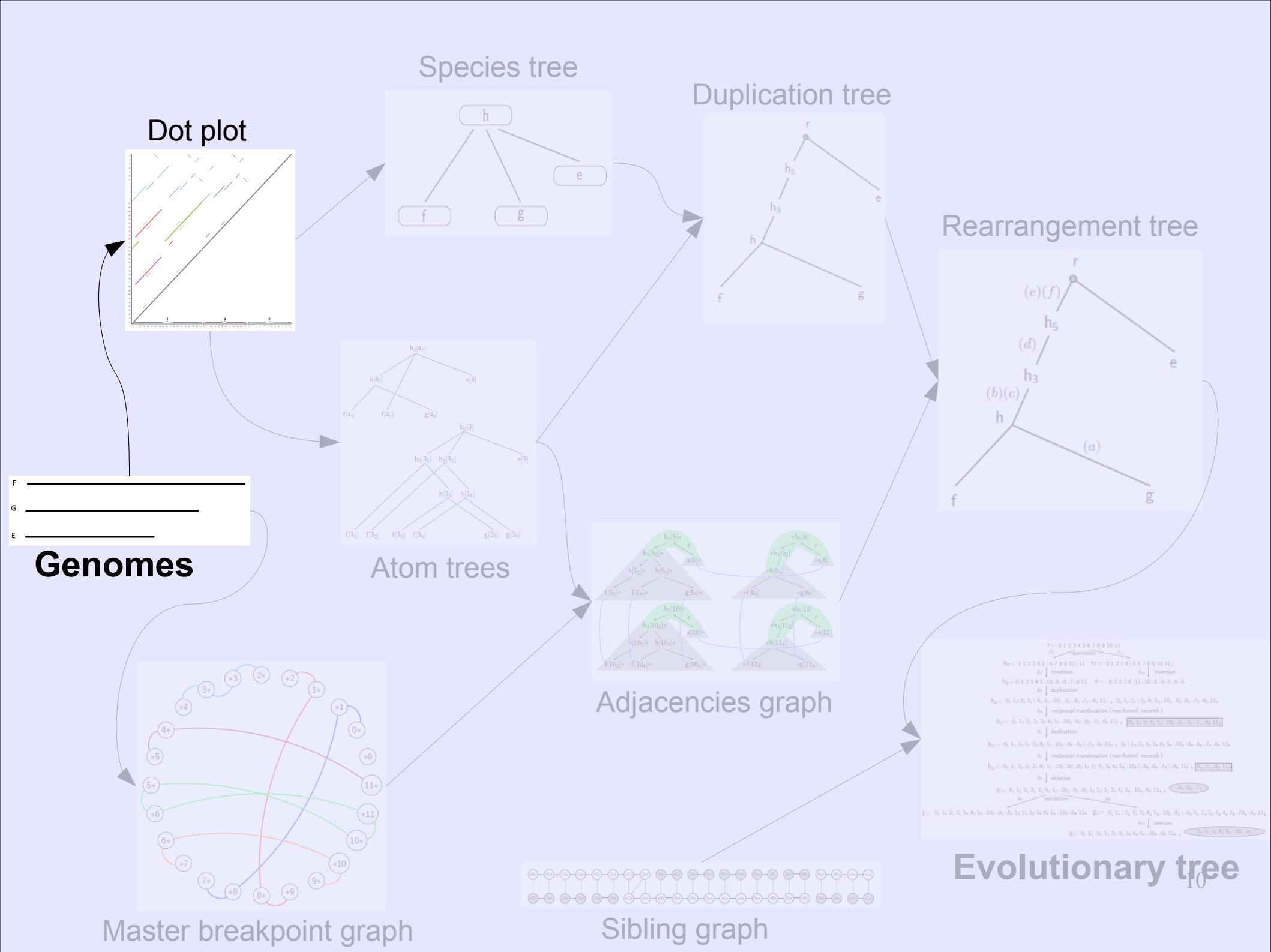
Operations

- Duplications
- Rearrangements
  - 2-breakpoints (inversion, fusion, reciprocal translocation, ...)
  - 3-breakpoints (transposition, transposition with inversion, …)
  - **With duplication**: tandem segmental duplication and duplicative transposition.
- Insertion/Deletion:



Polynomial-time Algorithm


Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree


Dot plot
Species tree
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

Example: **3 genomes** with only one linear chromosome

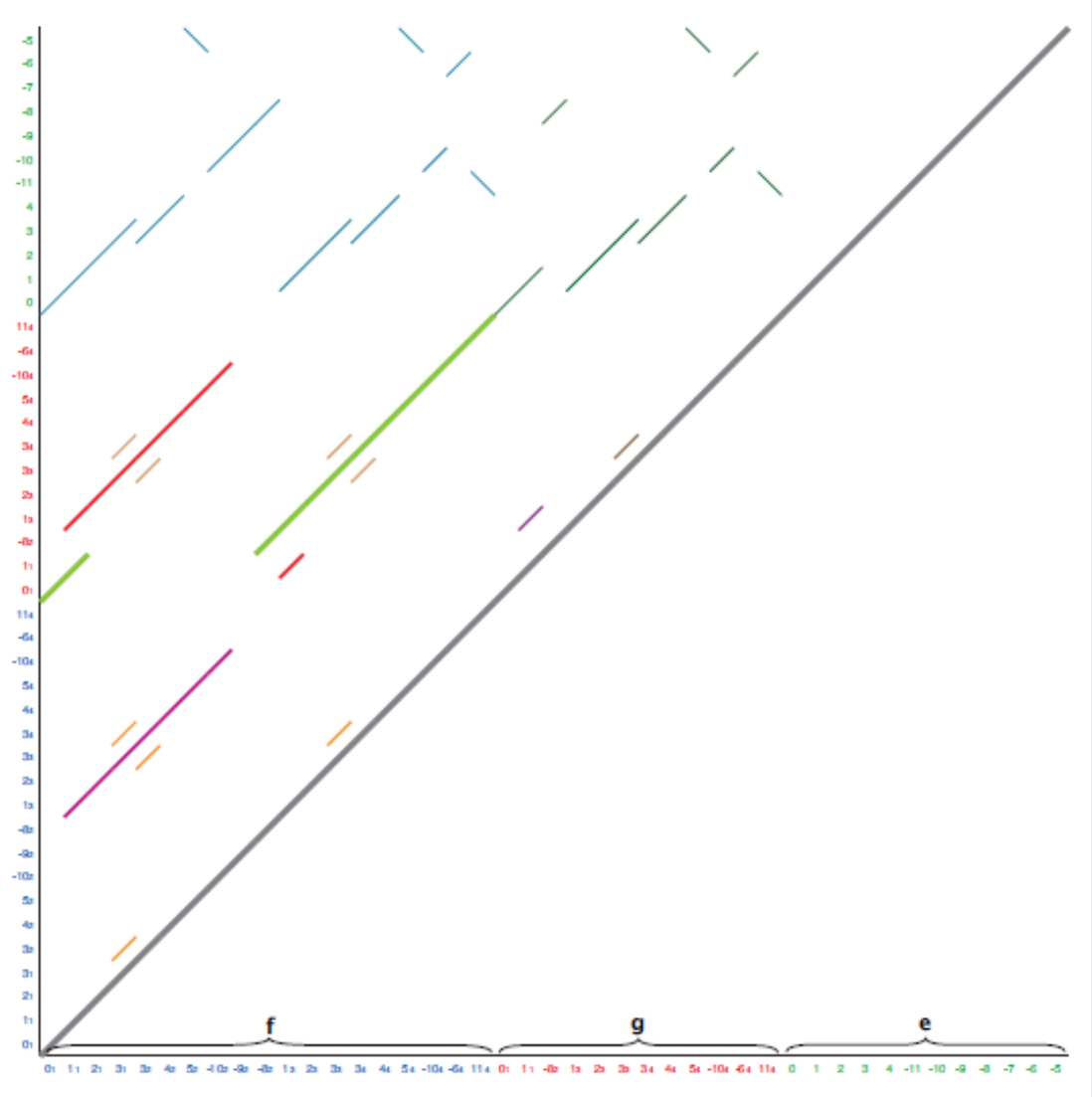
Genome F
TGGCTACTGTAGCCTAGGTATCTATGTT...

Genome G
GCATGCCATTGTAGCCGATCGATATGC...

Genome E
AGTGCGGAGTGCGCGAGTTGAAGTGT...

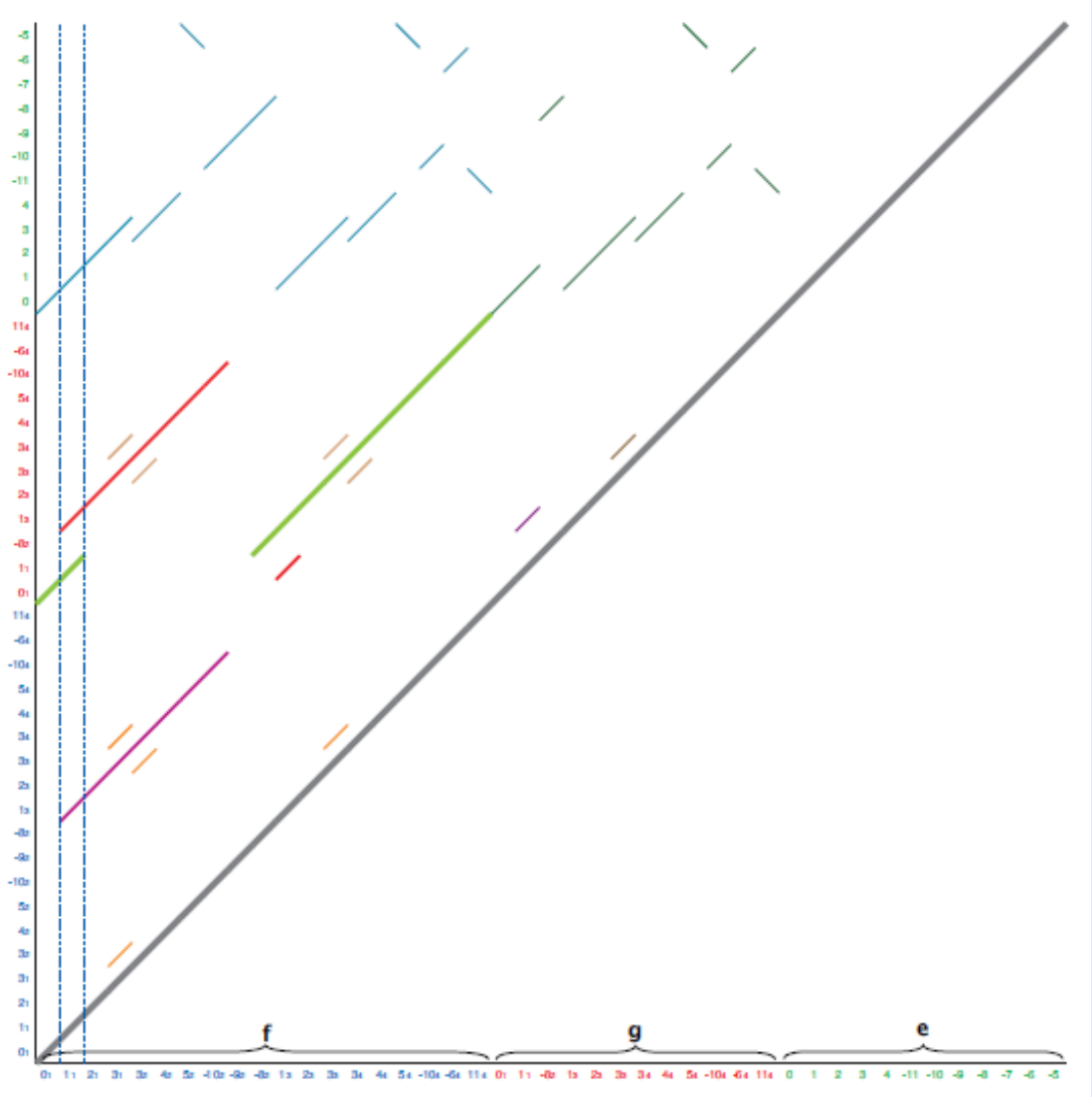
Creation of the dot plot

Aim: Decompose genomes in atoms
Technique: Local alignment



Creation of the dot plot

Aim: Decompose genomes in atoms
Technique: Local alignment



Genome = sequence of atoms

Genome F

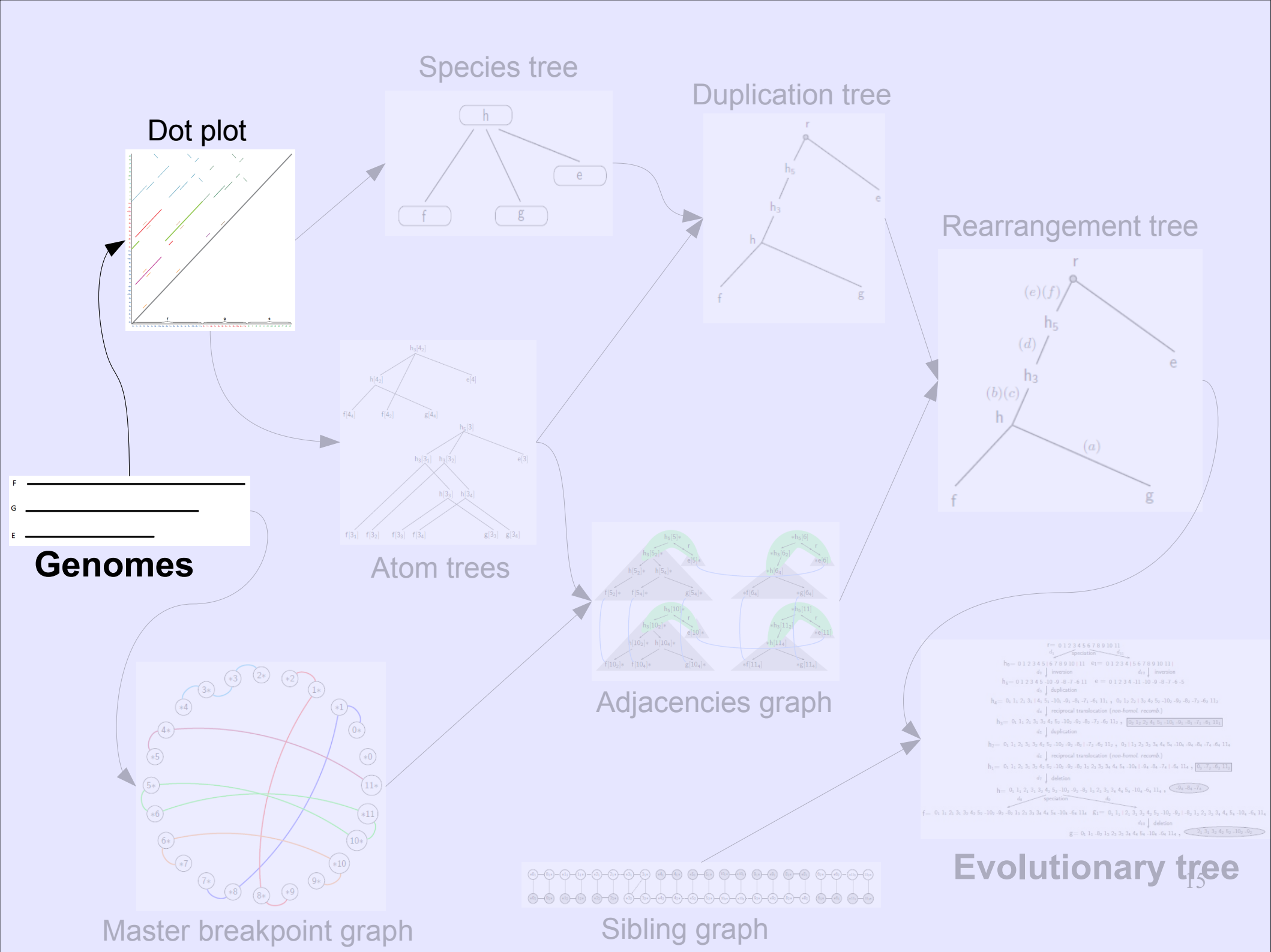
0 1 2 3 3 4 5 -10 -9 -8 1 2 3 3 4 5 -10 -6 11

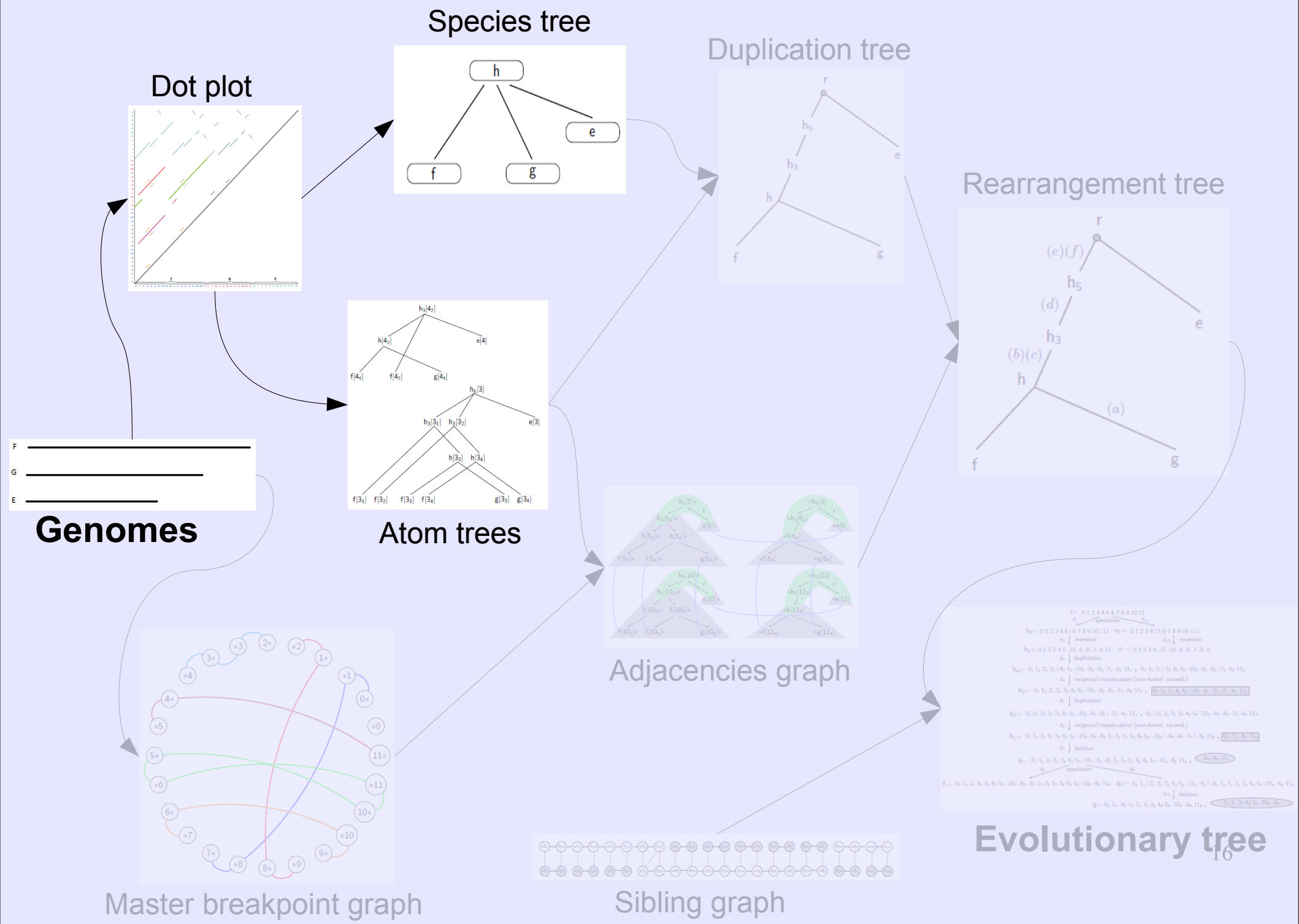
Genome G

0 1 -8 1 2 3 3 4 5 -10 -6 11

Genome E

0 1 2 3 4 -11 -10 -9 -8 -7 -6 -5


Dot plot
Species tree
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree


Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

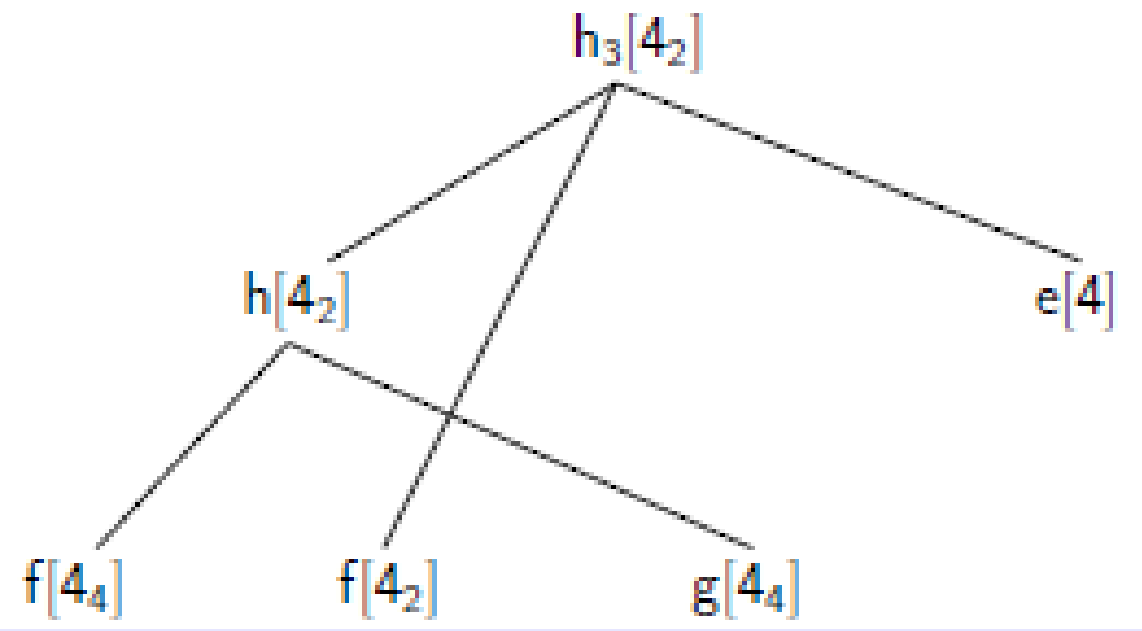
Creation of atom trees

Aim: Represent information specific of each atom
Technique: Neighbor Joining [Saitou and Nai - 1987], distance D of local alignments.

Genome F
0 1 2 3 3 **4** 5 -10 -9 -8 1 2 3 3 **4** 5 -10 -6 11

Genome G
0 1 -8 1 2 3 3 **4** 5 -10 -6 11

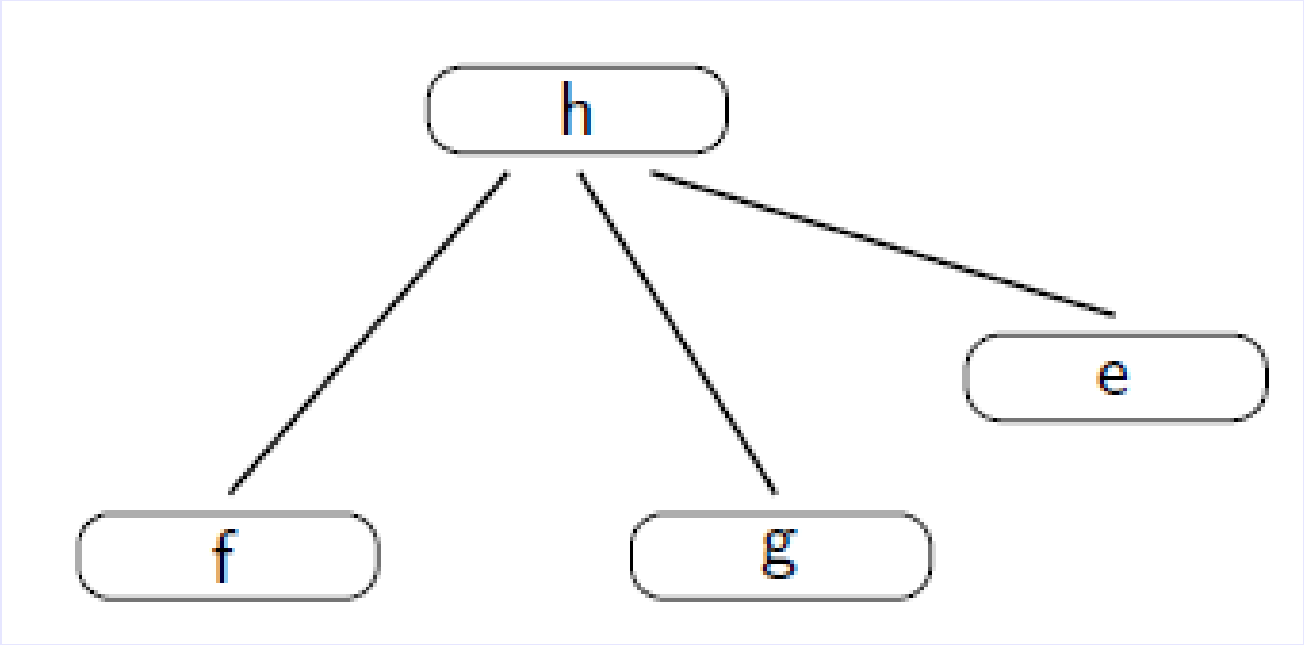
Genome E
0 1 2 3 **4** -11 -10 -9 -8 -7 -6 -5

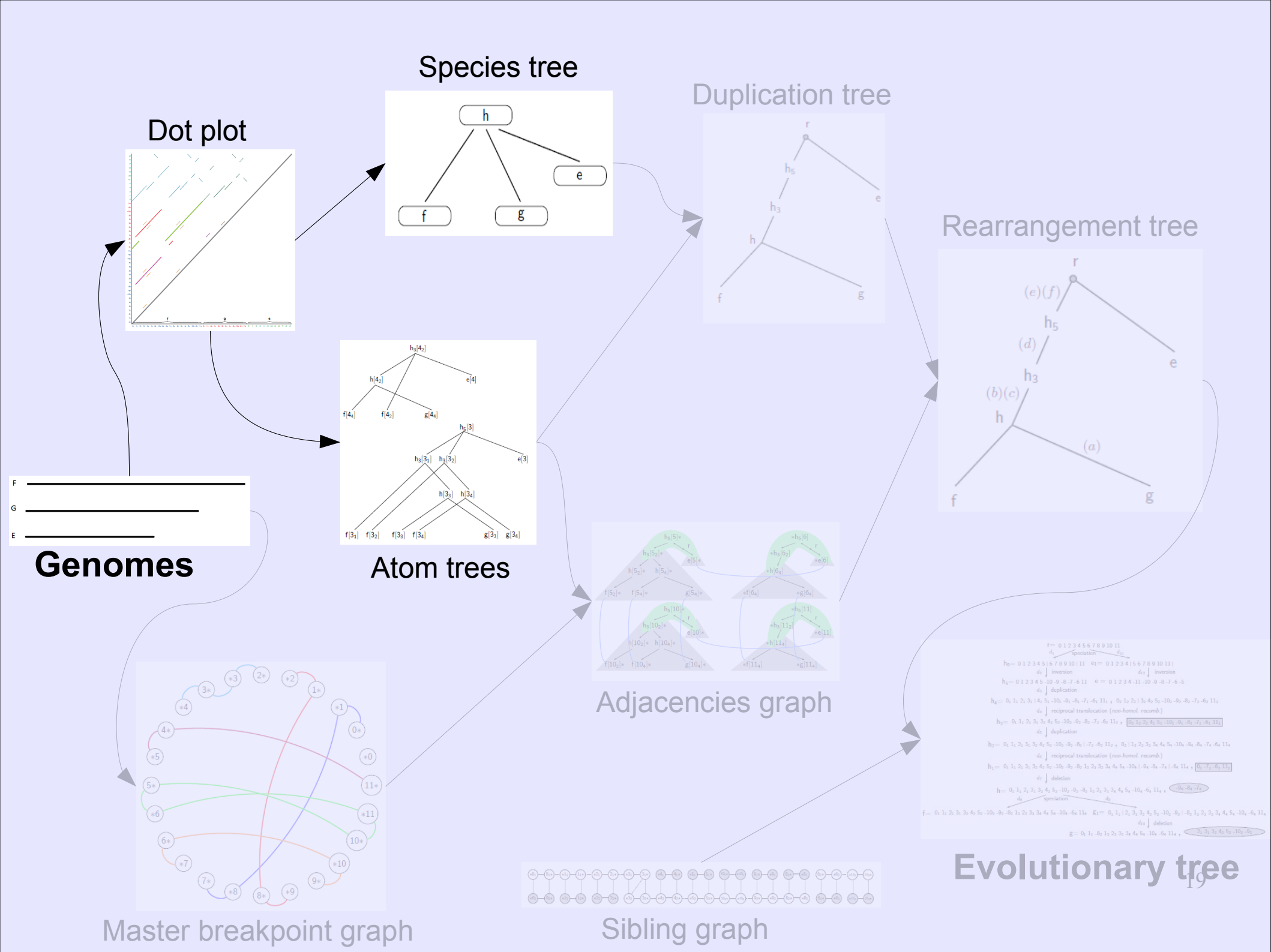


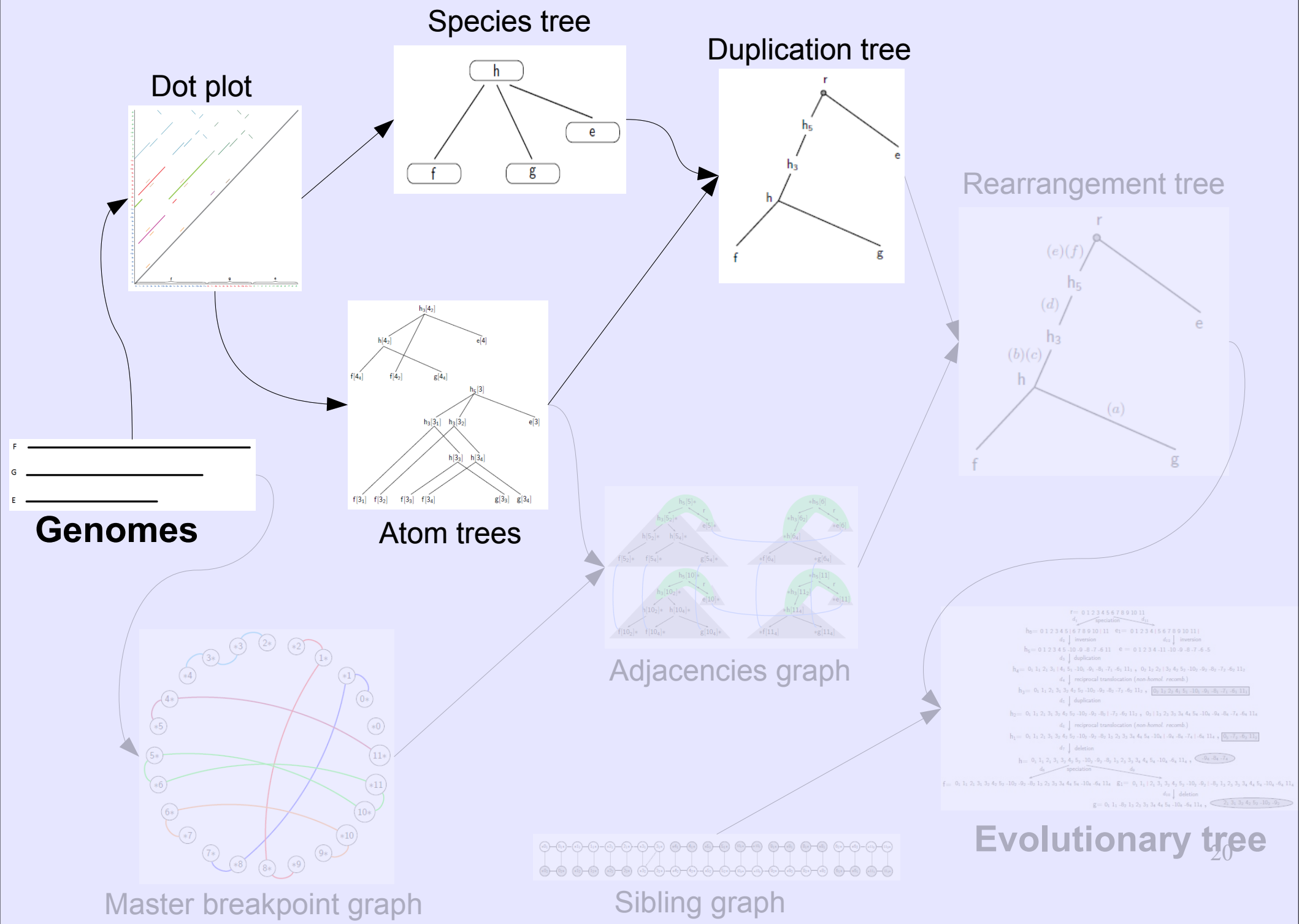
Atom tree T4

Creation of species tree

Aim: Represent the relation between species
Technique: Neighbor Joining, distance = min(D(x,y))
Hypothesis: The substitution rate is the same for all sites in a species.




Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

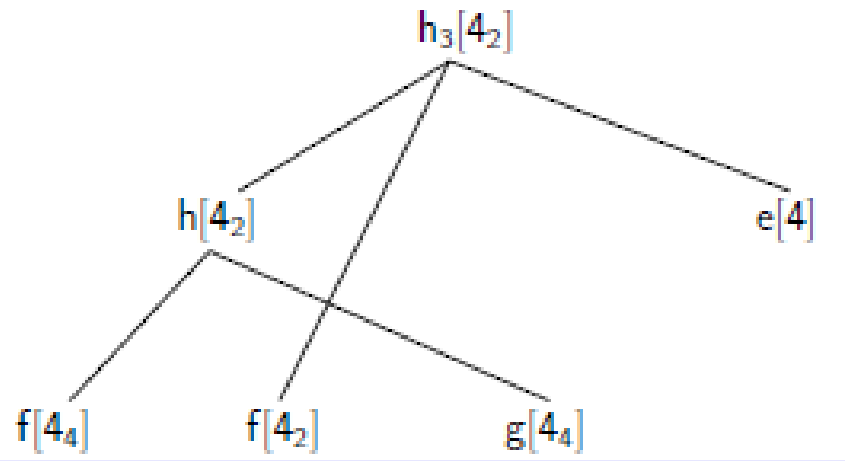

Dot plot
Species tree
h
f
g
e
Duplication tree
r
h5
h3
h
e
f
g
Rearrangement tree
r
(e)(f)
h5
(d)
h3
e
(b)(c)
h
(a)
f
g
F
G
E
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

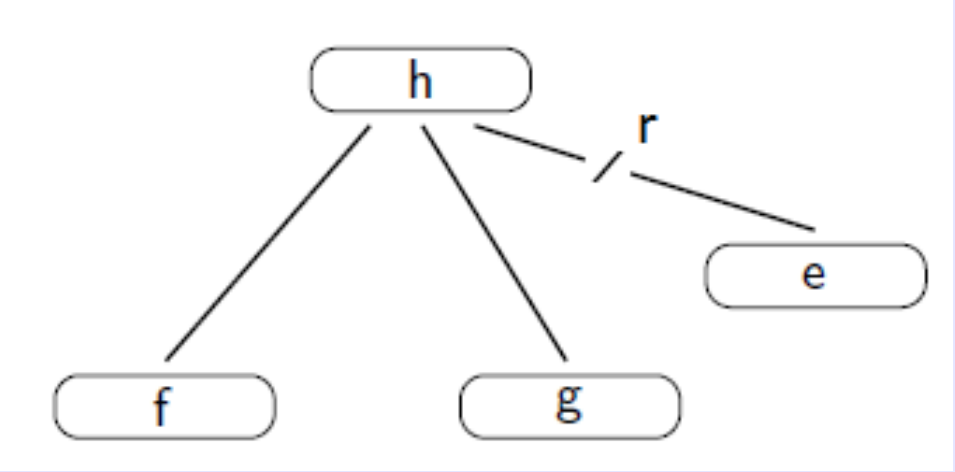
Creation of the duplication tree

Aim: Add duplications in the species tree.

Technique: Reconcile the atom trees with the species tree with a personal algorithm.

Hypothesis: The substitution rate is the same for all sites in a species.



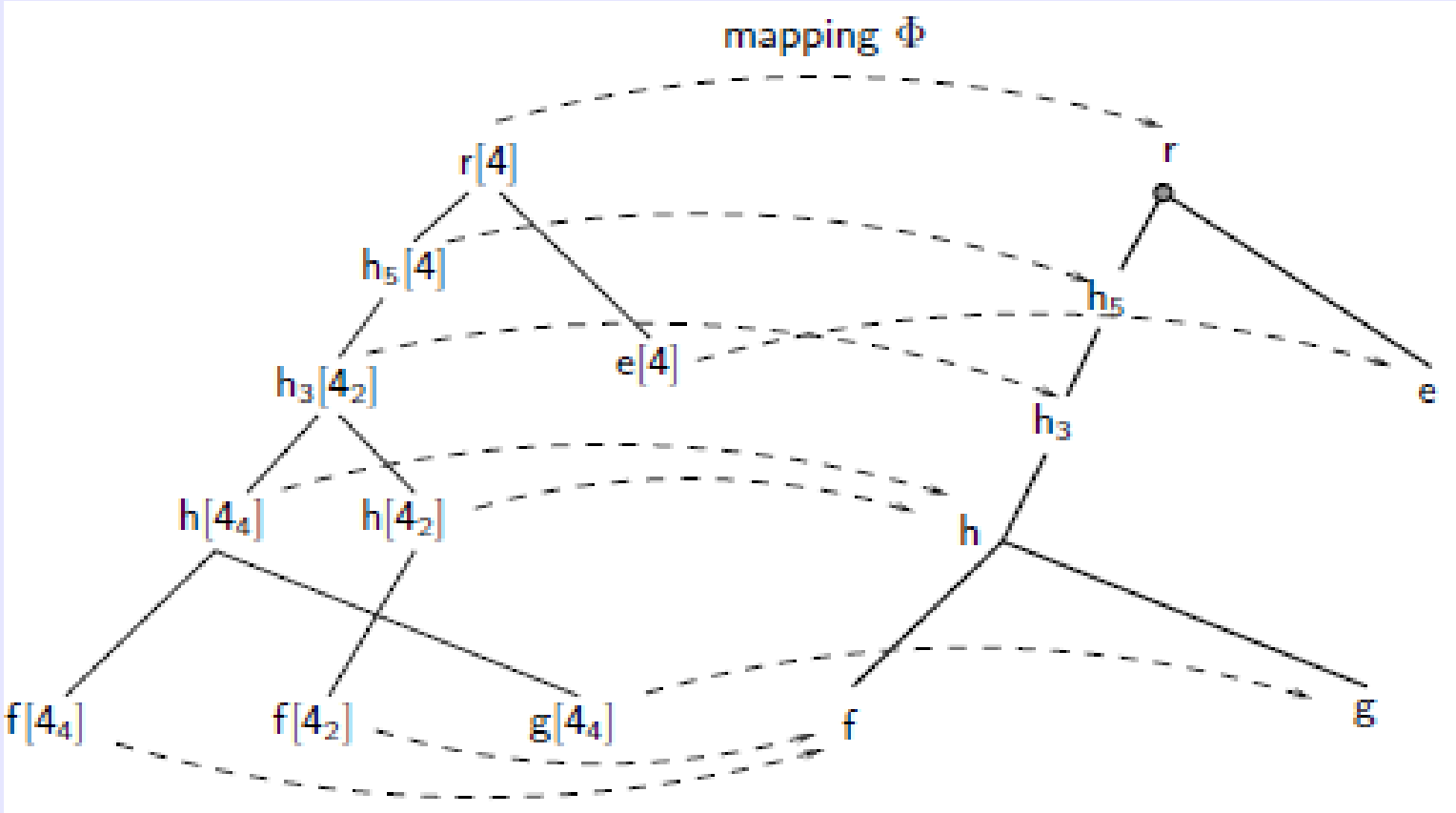


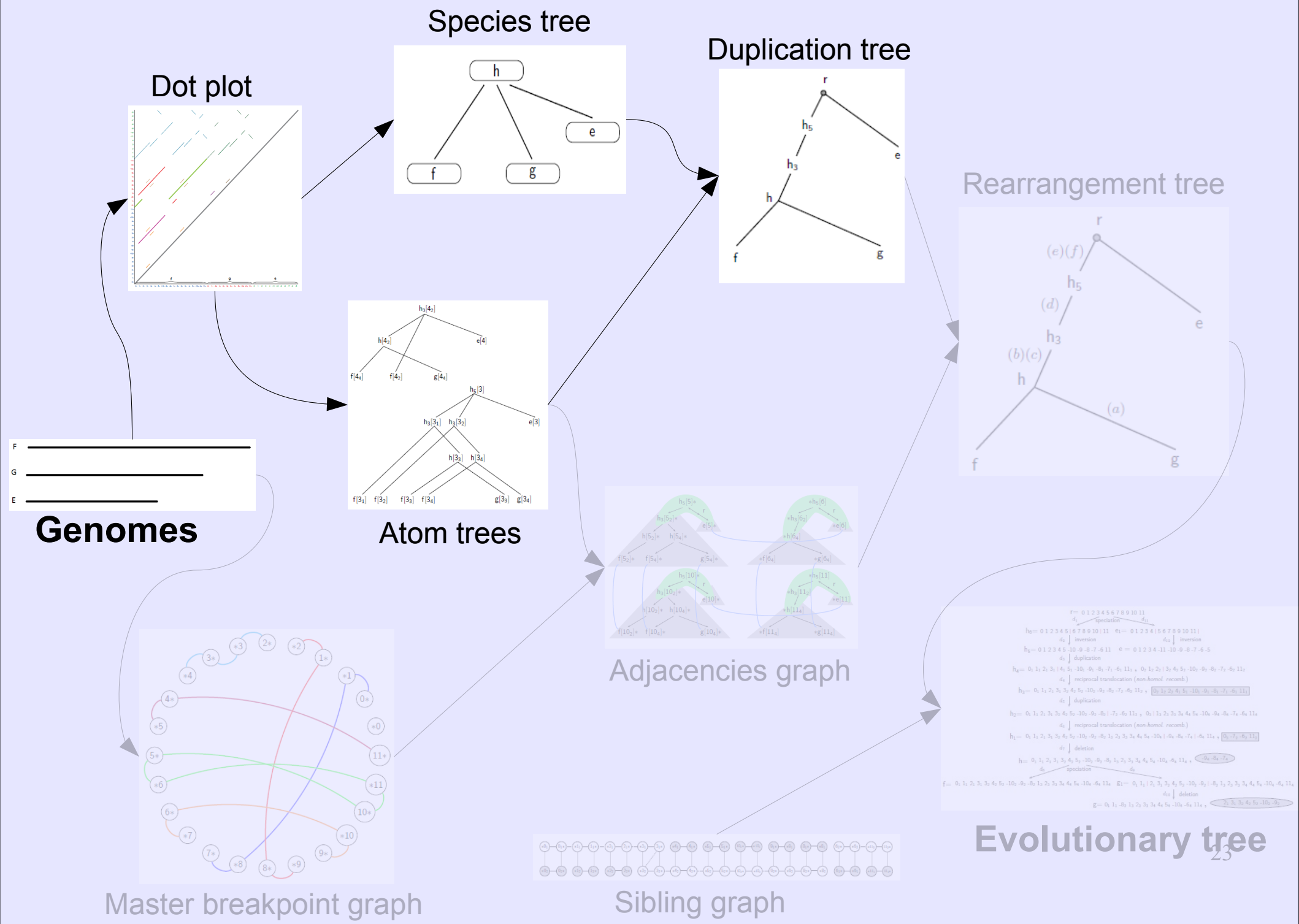
Creation of the duplication tree

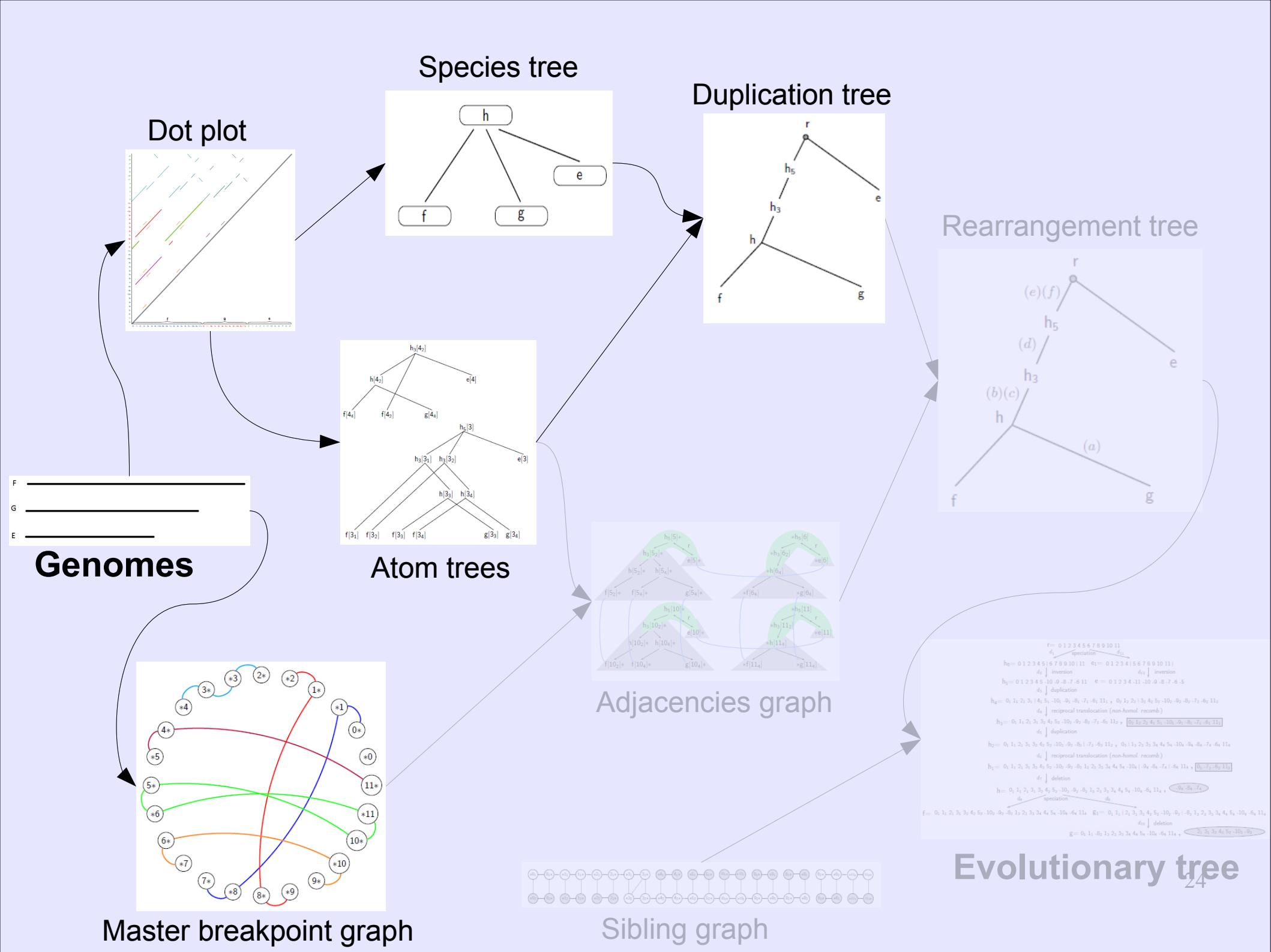
Aim: Add duplications in the species tree.

Technique: Reconcile the atom trees with the species tree with a personal algorithm.

Hypothesis: The substitution rate is the same for all sites in a species.

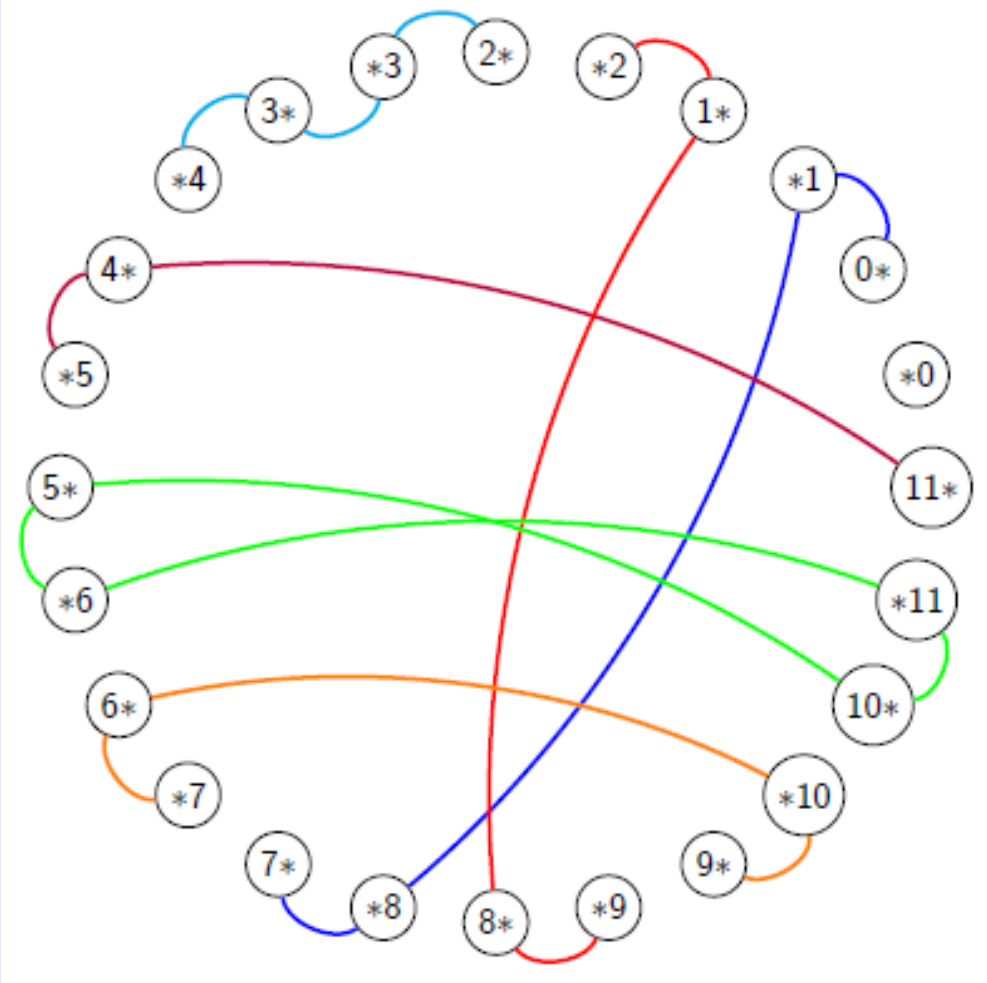



Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree


Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

Creation of the master breakpoint graph

Aim: Define rearrangement
Hypothesis: No breakpoint is used twice.



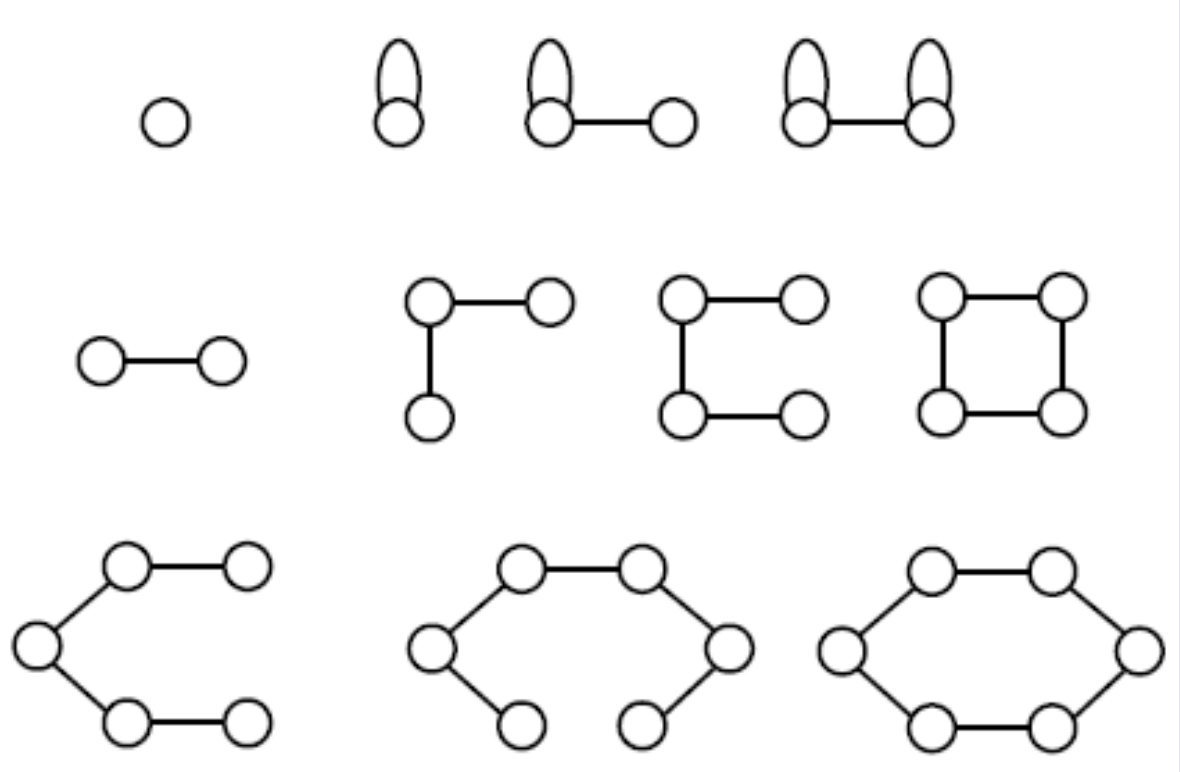
Genome F
0 1 2 3 3 4 5 -10 -9 -8 1 2 3 3 4 5 -10 -6 11

Genome G
0 1 -8 1 2 3 3 4 5 -10 -6 11

Genome E
0 1 2 3 4 -11 -10 -9 -8 -7 -6 -5

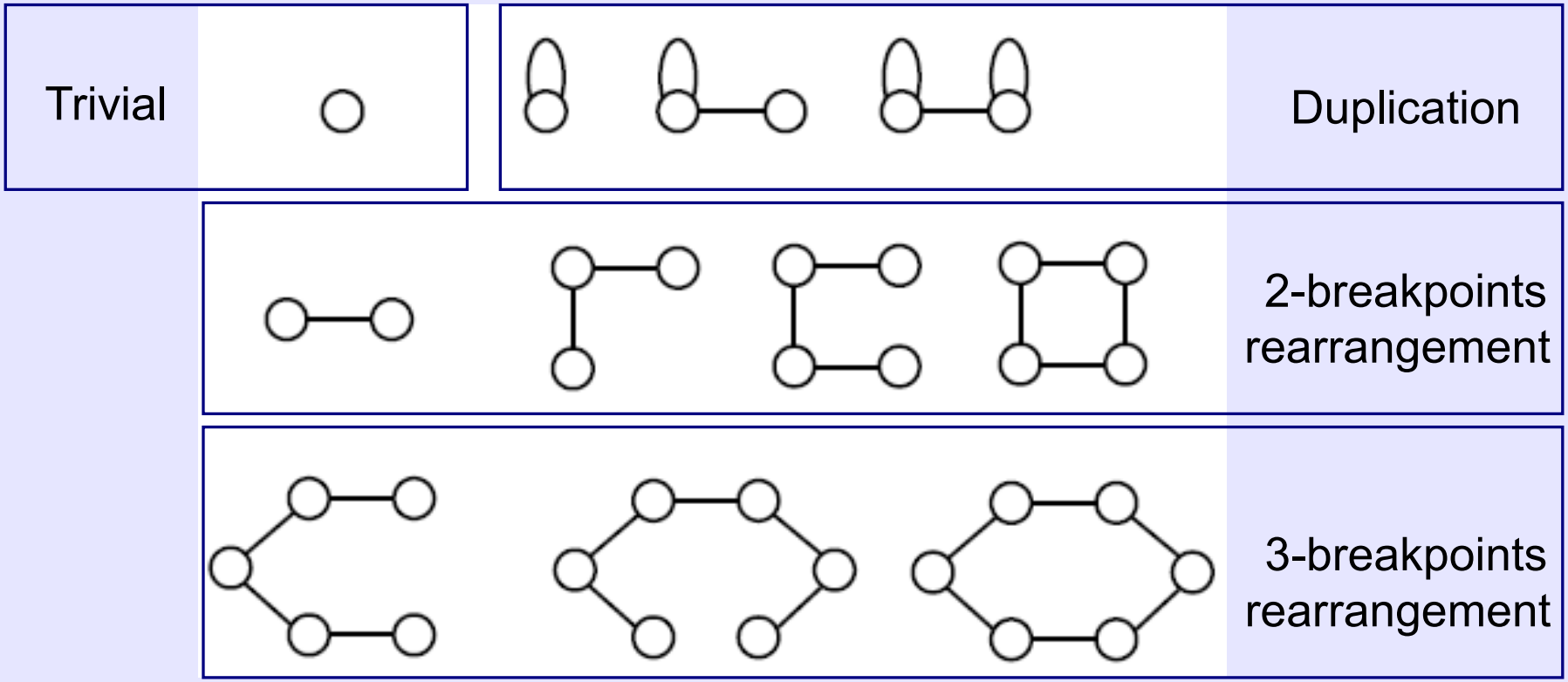
Decomposition

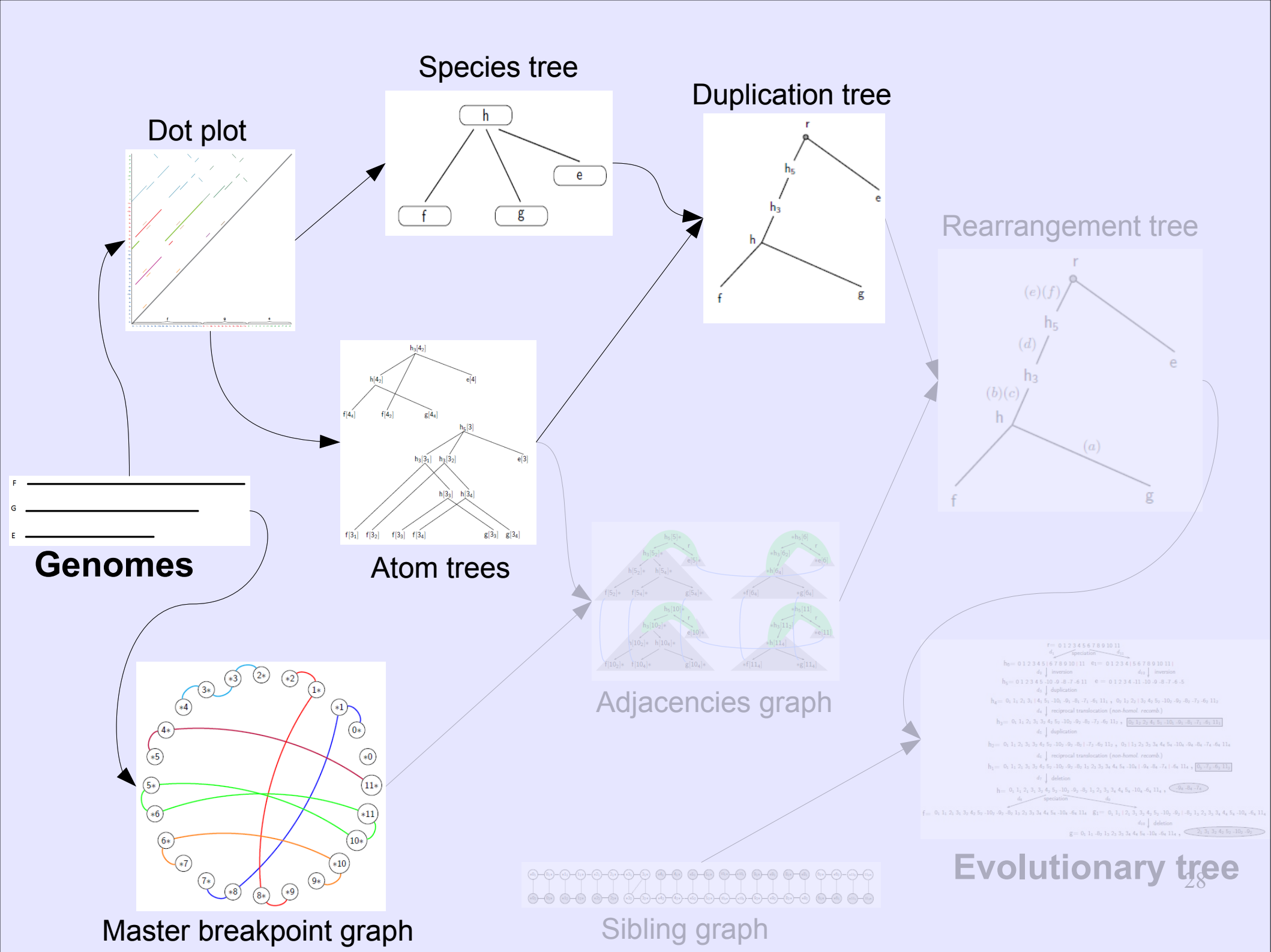
Each component correspond to a rearrangement.

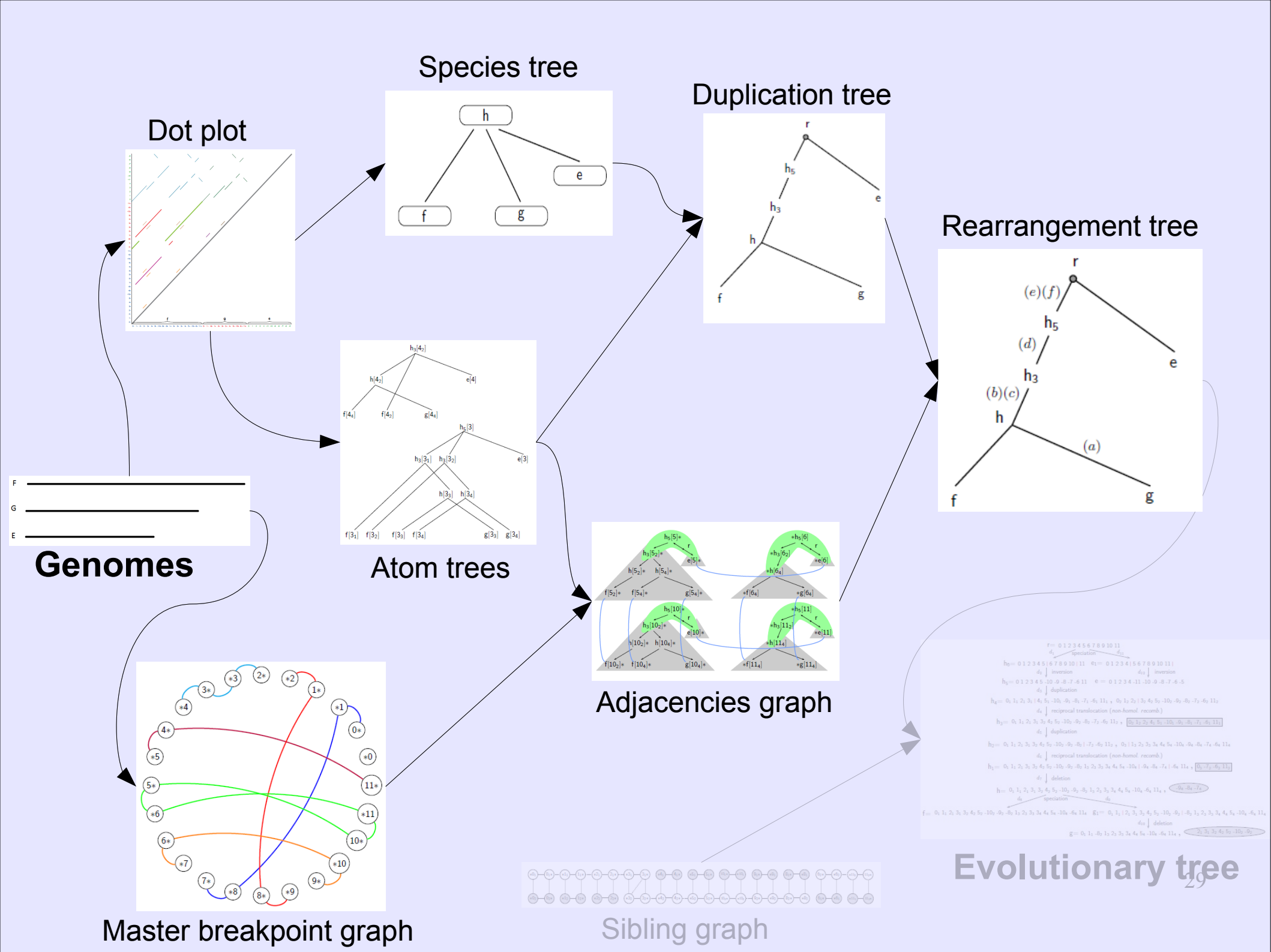


Decomposition

Each component correspond to a rearrangement.

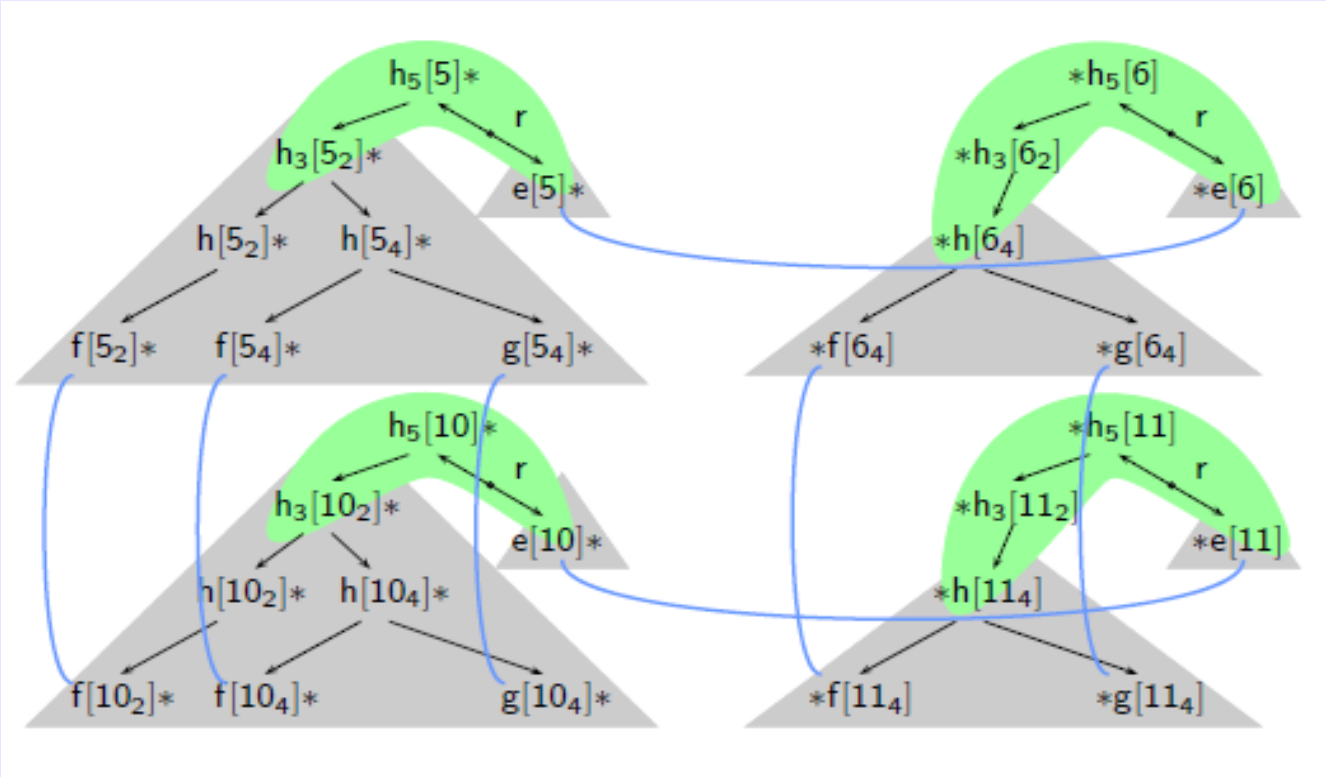



Dot plot
Species tree
h
e
f
g
Duplication tree
r
h5
h3
h
e
f
g
Rearrangement tree
r
(e)(f)
h5
(d)
h3
(b)(c)
h
(a)
e
f
g
F
G
E
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree


Species tree
Duplication tree
Dot plot
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

Creation of the adjacencies graph

Aim: Find the possible places for each rearrangement
Technique: Use master breakpoint graph and atom trees



Genome F
0 1 2 3 3 4 **5 -10** -9 -8 1 2 3 3 4 **5 -10 -6 11**

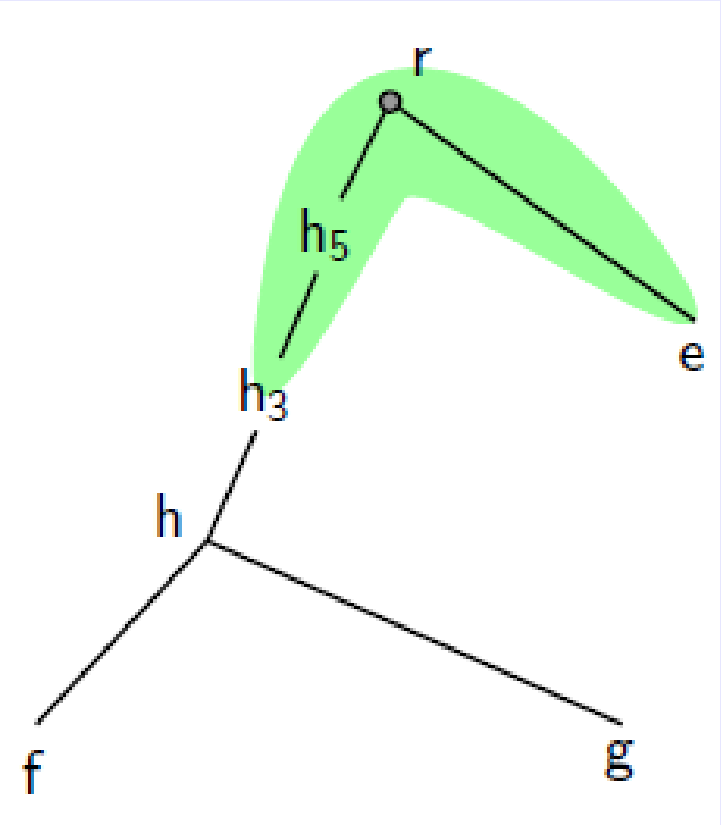
Genome G
0 1 -8 1 2 3 3 4 **5 -10 -6 11**

Genome E
0 1 2 3 4 **-11 -10** -9 -8 -7 **-6 -5**

Adjacencies graph for 1 rearrangement with iso-adjacencies subtrees and tethers.

Creation of the switchpoint zone

Aim: Find the possible places for each rearrangement
Technique: The switchpoint zone is the intersection of tethers.

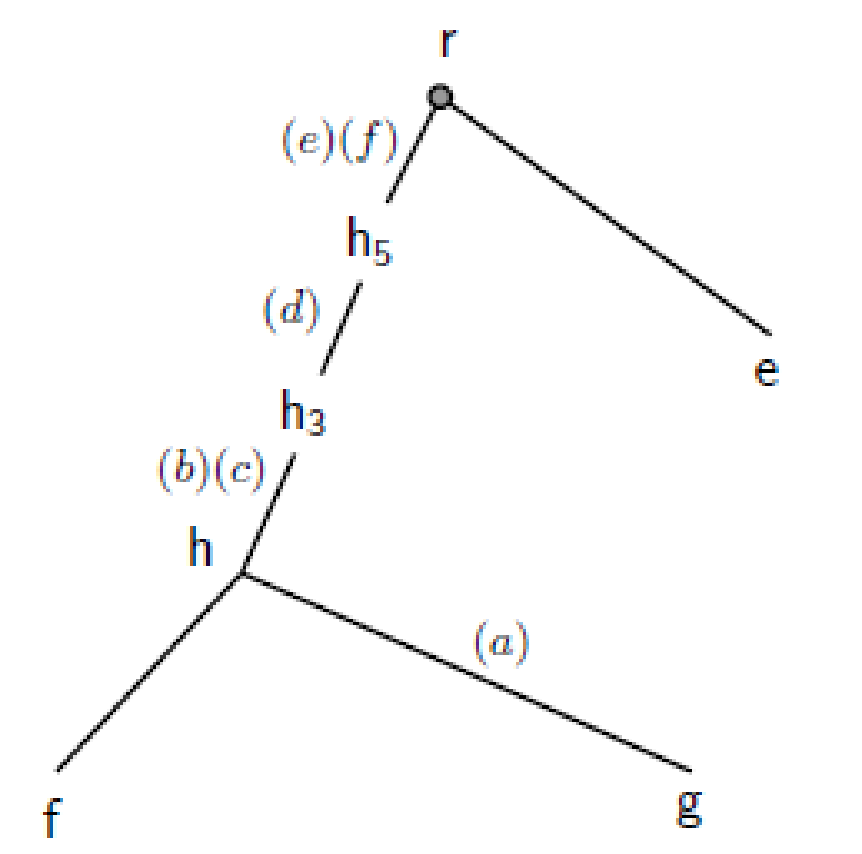


Switchpoint zone for the 2-breakpoints rearrangement

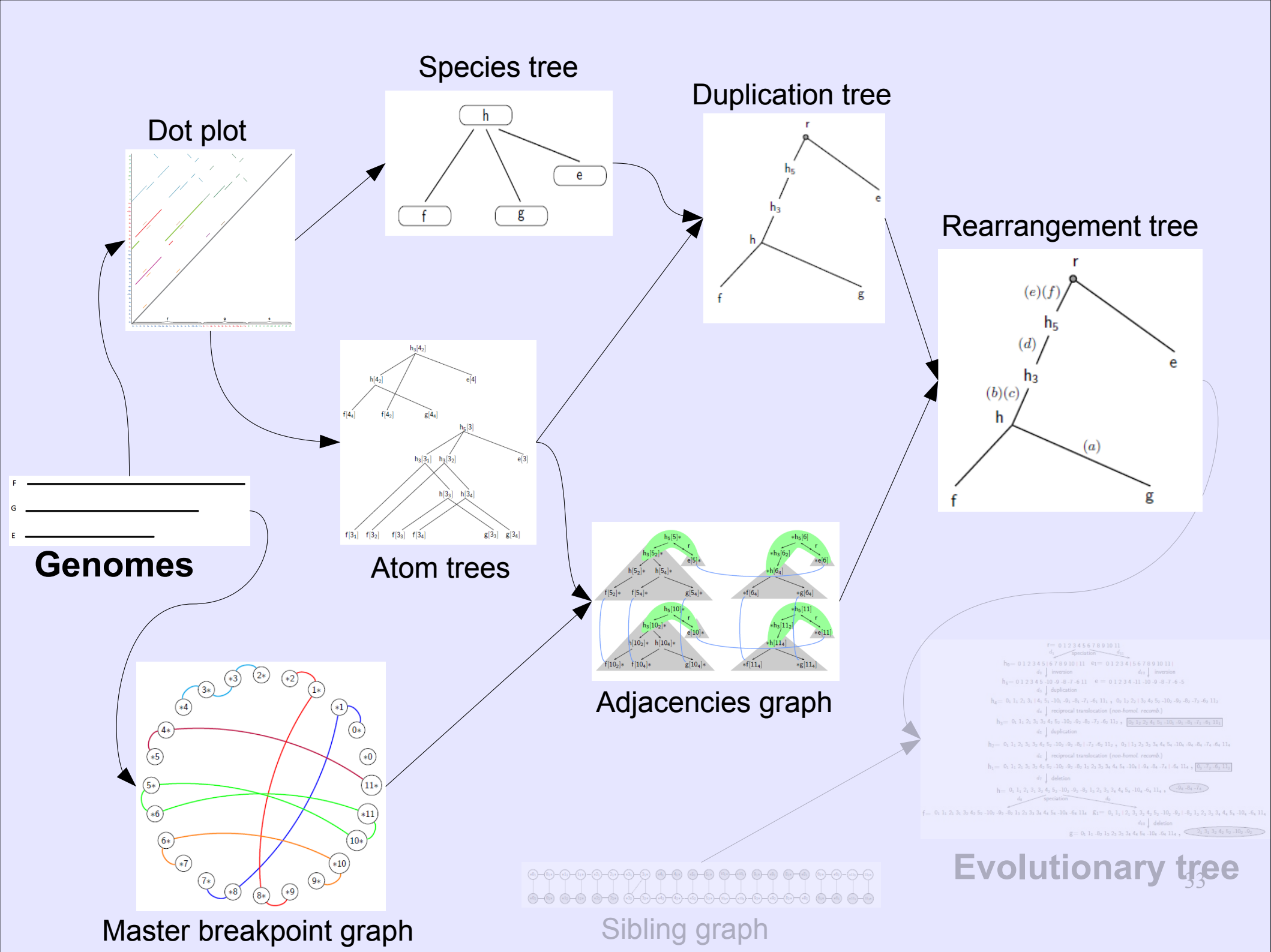
Creation of the adjacencies tree

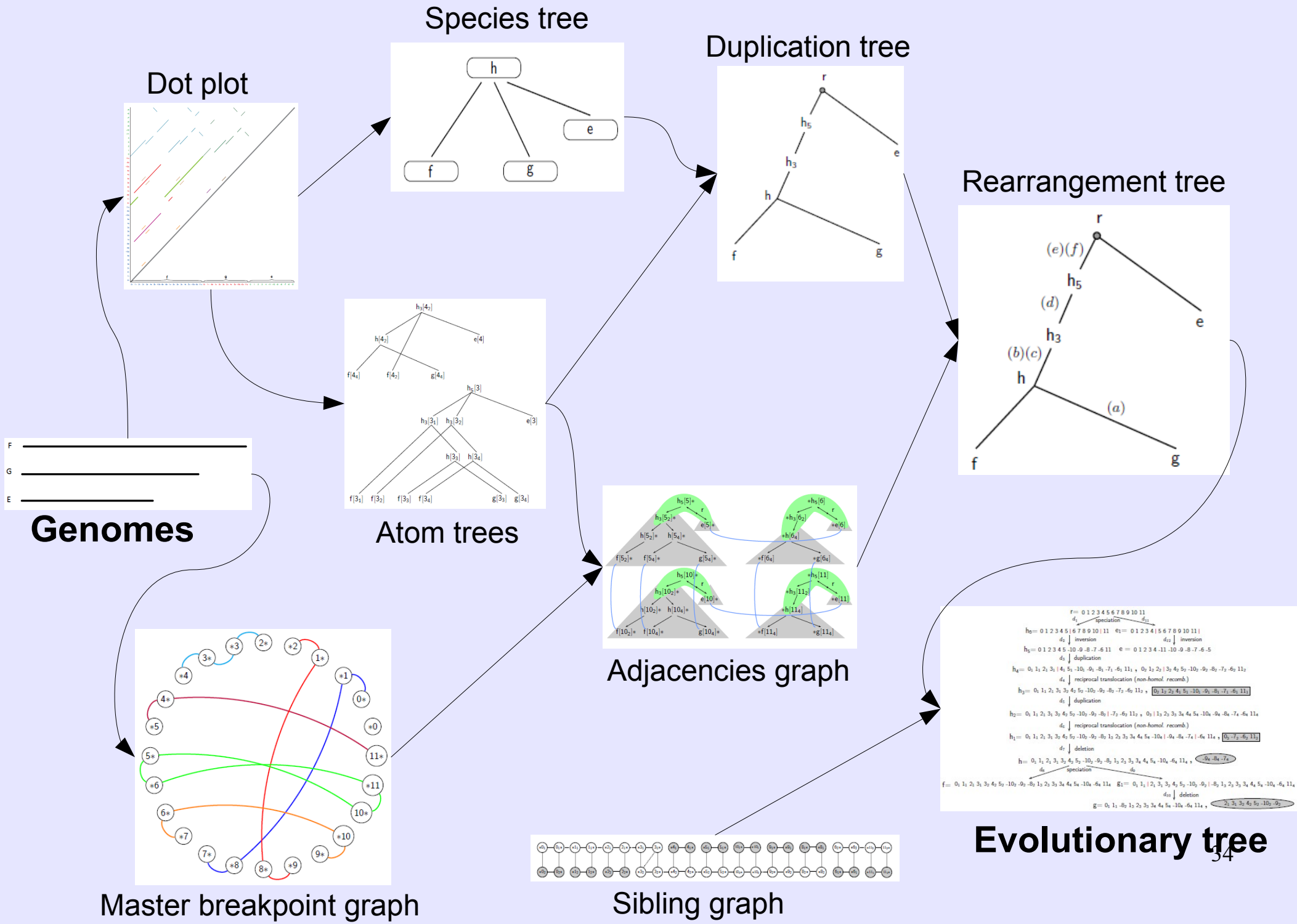
Aim: Place each rearrangement

Technique: Random choice with the respect of each switchpoint zone.



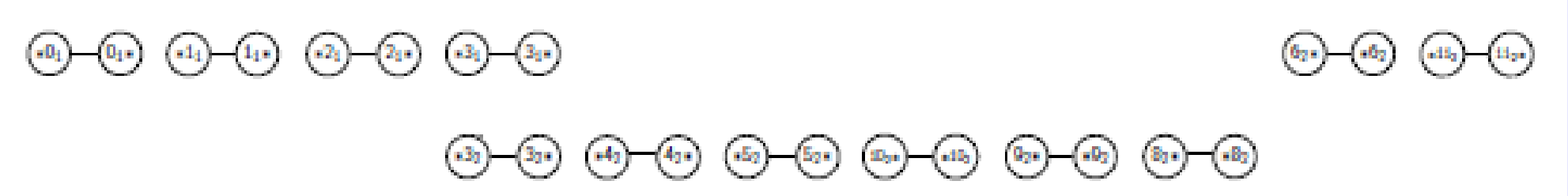
Adjacencies tree


Dot plot
Species tree
h
e
f
g
Duplication tree
r
h5
h3
h
e
f
g
Rearrangement tree
r
(e)(f)
h5
(d)
h3
(b)(c)
h
(a)
e
f
g
F
G
E
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree


Species tree
Dot plot
Duplication tree
Rearrangement tree
Genomes
Atom trees
Adjacencies graph
Master breakpoint graph
Sibling graph
Evolutionary tree

Creation of the sibling graph

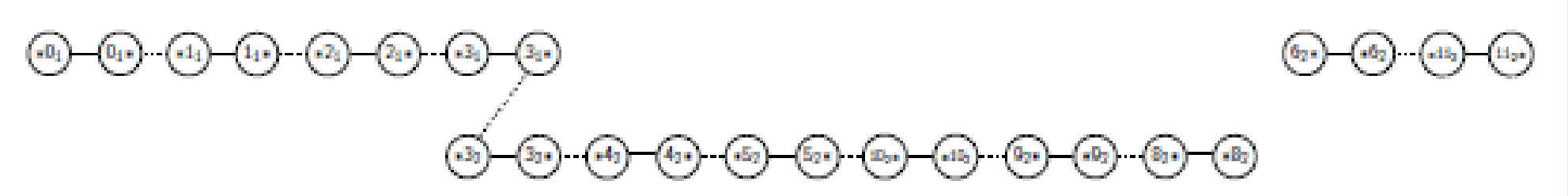
Aim: Deduce the sequence of ancestral genomes
Technique: Use a sibling graph



A sibling graph
Atom ends

Creation of the sibling graph

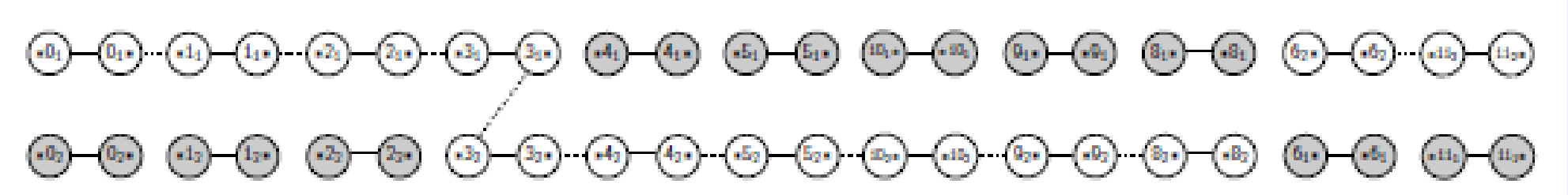
Aim: Deduce the sequence of ancestral genomes
Technique: Use a sibling graph



A sibling graph
Atom ends
Child adjacency edges

Creation of the sibling graph

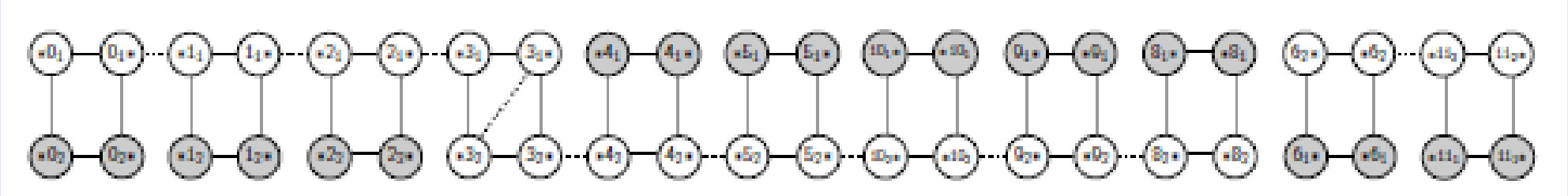
Aim: Deduce the sequence of ancestral genomes
Technique: Use a sibling graph



A sibling graph
Atom ends
Child adjacency edges
Artificial sibling nodes

Creation of the sibling graph

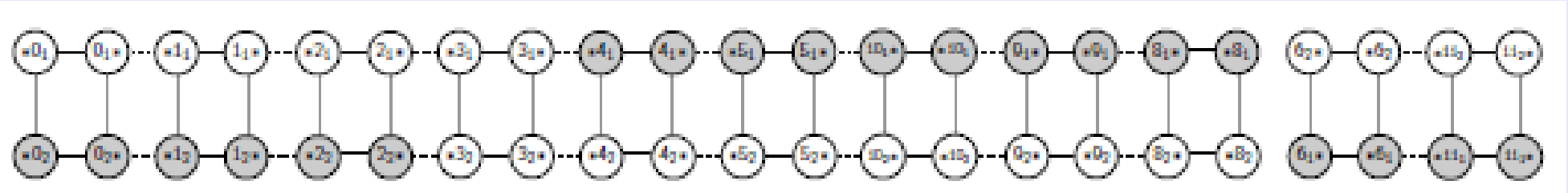
Aim: Deduce the sequence of ancestral genomes
Technique: Use a sibling graph



A sibling graph
Atom ends
Child adjacency edges
Artificial sibling nodes
Sibling edges

Creation of the sibling graph

Aim: Deduce the sequence of ancestral genomes
Technique: Use a sibling graph



A sibling graph
Atom ends
Child adjacency edges
Artificial sibling nodes
Sibling edges
Parent adjacency edges

The evolutionary tree

$r = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11$

d_1 speciation d_{11}

$h_6 = 0\ 1\ 2\ 3\ 4\ 5 \mid 6\ 7\ 8\ 9\ 10 \mid 11$ $\quad e_1 = 0\ 1\ 2\ 3\ 4 \mid 5\ 6\ 7\ 8\ 9\ 10\ 11 \mid$

d_2 inversion $\quad d_{12}$ inversion

$h_5 = 0\ 1\ 2\ 3\ 4\ 5\ -10\ -9\ -8\ -7\ -6\ 11$ $\quad e = 0\ 1\ 2\ 3\ 4\ -11\ -10\ -9\ -8\ -7\ -6\ -5$

d_3 duplication

$h_4 = 0_1\ 1_1\ 2_1\ 3_1 \mid 4_1\ 5_1\ -10_1\ -9_1\ -8_1\ -7_1\ -6_1\ 11_1\ ,\ 0_2\ 1_2\ 2_2 \mid 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2\ -7_2\ -6_2\ 11_2$

d_4 reciprocal translocation (*non-homol. recomb.*)

$h_3 = 0_1\ 1_1\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2\ -7_2\ -6_2\ 11_2\ ,\ \boxed{0_2\ 1_2\ 2_2\ 4_1\ 5_1\ -10_1\ -9_1\ -8_1\ -7_1\ -6_1\ 11_1}$

d_5 duplication

$h_2 = 0_1\ 1_1\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2 \mid -7_2\ -6_2\ 11_2\ ,\ 0_3 \mid 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4\ -9_4\ -8_4\ -7_4\ -6_4\ 11_4$

d_6 reciprocal translocation (*non-homol. recomb.*)

$h_1 = 0_1\ 1_1\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2\ 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4 \mid -9_4\ -8_4\ -7_4 \mid -6_4\ 11_4\ ,\ \boxed{0_3\ -7_2\ -6_2\ 11_2}$

d_7 deletion

$h = 0_1\ 1_1\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2\ 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4\ -6_4\ 11_4\ ,\ -9_4\ -8_4\ -7_4$

d_8 speciation d_9

$f = 0_1\ 1_1\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2\ -8_2\ 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4\ -6_4\ 11_4$ $\quad g_1 = 0_1\ 1_1 \mid 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2 \mid -8_2\ 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4\ -6_4\ 11_4$

d_{10} deletion

$g = 0_1\ 1_1\ -8_2\ 1_3\ 2_3\ 3_3\ 3_4\ 4_4\ 5_4\ -10_4\ -6_4\ 11_4\ ,\ 2_1\ 3_1\ 3_2\ 4_2\ 5_2\ -10_2\ -9_2$

Experimentation

- **Real genomes:**
  - 5 chromosomes X (human, chimp, rhesus, mouse, rat).
  - Difficulties to assess.

- **Simulated genomes** (100 experiments):

	simulated data	real data
reuse ratio r	1.38	1.38
atoms	2008.10	1917
engineered atoms	33.97	15
reused adj [%]	19.55	15.02
reused adj (once) [%]	13.74	11.53
reused adj (twice) [%]	4.18	3.00
reused adj (3 times) [%]	1.18	0.39
reused adj (4 times) [%]	0.31	0.13
operations	1667.55	1660
2-bp operations	1470.94	1462
3-bp operations	196.61	198
deletions	695.58	747
insertions	232.39	289
atoms in human	1483.47	1343
atoms in chimp	1442.12	1229
atoms in macaque	1457.33	1211
atoms in mouse	1050.76	896
atoms in rat	1046.58	784
atoms in dog	974.41	727

Results

- **Complexity**: Polynomial in the number of chromosomes and local alignments.

- Computation **time**: Unknown.

- **Unambiguous** atoms and adjacencies: Find correctly.

- **Ambiguous** and **new** atoms and adjacencies: Very bad results.

- **When reuse** breakpoint rate increase: Worst results.

- With a **outgroup**: Worst results

Results

- **Complexity**: Polynomial in the number of chromosomes and local alignments.

- Computation **time**: Unknown.

- **Unambiguous** atoms and adjacencies: Find correctly.

- **Ambiguous** and **new** atoms and adjacencies: Very bad results.

- When **reuse breakpoint** rate increase: Worst results.

- With a **outgroup**: Worst results

Conclusion

- Polynomial-time algorithm
- Strong constraints
- Weight
- Horizontal transfer