

Software for isometric gene tree reconciliation

DOMINIKA MIHÁLOVÁ

SUPERVISOR: DOC. MGR. BRONISLAVA BREJOVÁ, PHD.

Introduction

- Species tree:
 - Leaves - present-day species
 - Internal nodes – speciations in the evolutionary history
- Gene tree:
 - Leaves – present-day copies of the gene
 - Internal nodes – duplications and gene losses in the evolutionary history
- Isometric reconciliation – branch lengths are taken into account while mapping a gene tree to a species tree
- Different approaches to gene tree reconciliation:
 - Scoring gene tree reconciliation (TreeBeST, TreeFix, Treerecs, Notung)
 - Probabilistic gene tree reconciliation (SPIMAP, PhylDog)
 - Isometric gene tree reconciliation

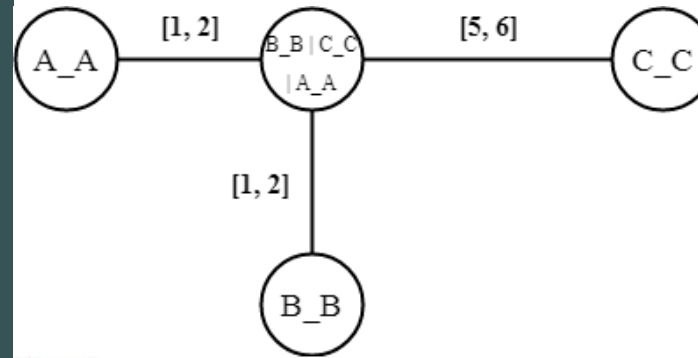
Main goal

- Implement software for isometric gene tree reconciliation
- Experimentally evaluate its accuracy on:
 - simulated biological data
 - real biological data

Software

- Source code based on master's thesis of Radoslav Chládek (Algorithms for isometric gene tree reconciliation. Univerzita Komenského v Bratislave, 2019)
- Programming language: Java
- Implemented algorithms:
 - Rooting the gene tree - get all possible rooted gene trees by shifting the root on every edge by given step
 - Counting algorithm:
 - Gets the number of duplications and gene losses
 - Counts in the direction from leaves to the root
 - Consider duplications and gene losses in node u and edge (u, v) , where node v is parent of node u
- Testing data: 12 Drosophila and 16 fungi (TreeFix, SPIMAP)

Testing – Rooting the gene tree



Tree 1

Edge: A_A, B_B | C_C | A_A

Interval 1: 0.0 0.0; Interval 2: 1.0 2.0

Interval 1: 1.0 2.0; Interval 2: 0.0 0.0

Interval 1: 0.0 0.5; Interval 2: 1.0 1.5

Interval 1: 1.0 1.5; Interval 2: 0.0 0.5

Edge: B_B, B_B | C_C | A_A

Interval 1: 0.0 0.0; Interval 2: 1.0 2.0

Interval 1: 1.0 2.0; Interval 2: 0.0 0.0

Interval 1: 0.0 0.5; Interval 2: 1.0 1.5

Interval 1: 1.0 1.5; Interval 2: 0.0 0.5

Edge: C_C, B_B | C_C | A_A

Interval 1: 0.0 0.0; Interval 2: 5.0 6.0

Interval 1: 5.0 6.0; Interval 2: 0.0 0.0

Interval 1: 0.0 0.5; Interval 2: 5.0 5.5

Interval 1: 1.0 1.5; Interval 2: 4.0 4.5

Interval 1: 2.0 2.5; Interval 2: 3.0 3.5

Interval 1: 3.0 3.5; Interval 2: 2.0 2.5

Interval 1: 4.0 4.5; Interval 2: 1.0 1.5

Interval 1: 5.0 5.5; Interval 2: 0.0 0.5

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

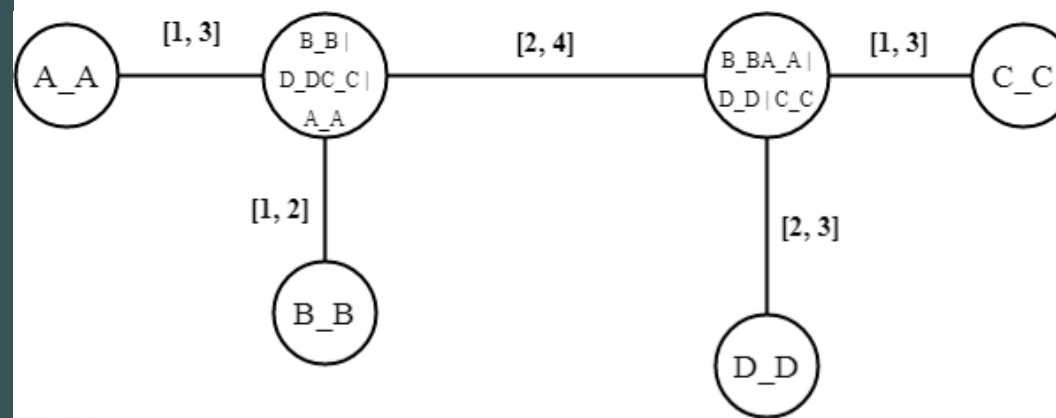
← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

Testing – Rooting the gene tree



Tree 2

Edge: A_A, B_B | D_DC_C | A_A

Interval 1: 0.0 0.0; Interval 2: 1.0 3.0

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

Interval 1: 1.0 3.0; Interval 2: 0.0 0.0

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

Interval 1: 0.0 1.0; Interval 2: 1.0 2.0

Interval 1: 1.0 2.0; Interval 2: 0.0 1.0

Edge: B_B, B_B | D_DC_C | A_A

Interval 1: 0.0 0.0; Interval 2: 1.0 2.0

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

Interval 1: 1.0 2.0; Interval 2: 0.0 0.0

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

Interval 1: 0.0 0.5; Interval 2: 1.0 1.5

Interval 1: 1.0 1.5; Interval 2: 0.0 0.5

Edge: B_BA_A | D_D | C_C, B_B | D_DC_C | A_A

Interval 1: 0.0 0.0; Interval 2: 2.0 4.0

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

Interval 1: 2.0 4.0; Interval 2: 0.0 0.0

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

Interval 1: 0.0 1.0; Interval 2: 2.0 3.0

Interval 1: 1.0 2.0; Interval 2: 1.0 2.0

Interval 1: 2.0 3.0; Interval 2: 0.0 1.0

Edge: C_C, B_BA_A | D_D | C_C

Interval 1: 0.0 0.0; Interval 2: 1.0 3.0

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

Interval 1: 1.0 3.0; Interval 2: 0.0 0.0

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

Interval 1: 0.0 1.0; Interval 2: 1.0 2.0

Interval 1: 1.0 2.0; Interval 2: 0.0 1.0

Edge: D_D, B_BA_A | D_D | C_C

Interval 1: 0.0 0.0; Interval 2: 2.0 3.0

← zakorenenie tesne nad vrcholmi = jedna hrana má dĺžku 0, druhá má pôvodnú dĺžku

Interval 1: 2.0 3.0; Interval 2: 0.0 0.0

← začne od 0 a ide s krokom 1; volnosť intervalu rozdelí na polovicu

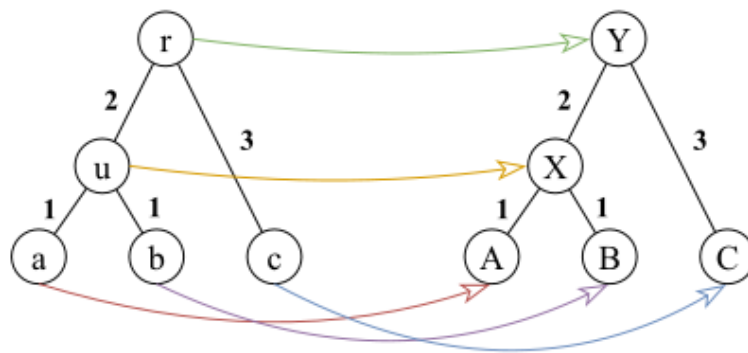
Interval 1: 0.0 0.5; Interval 2: 2.0 2.5

Interval 1: 1.0 1.5; Interval 2: 1.0 1.5

Interval 1: 2.0 2.5; Interval 2: 0.0 0.5

Testing – Counting algorithm

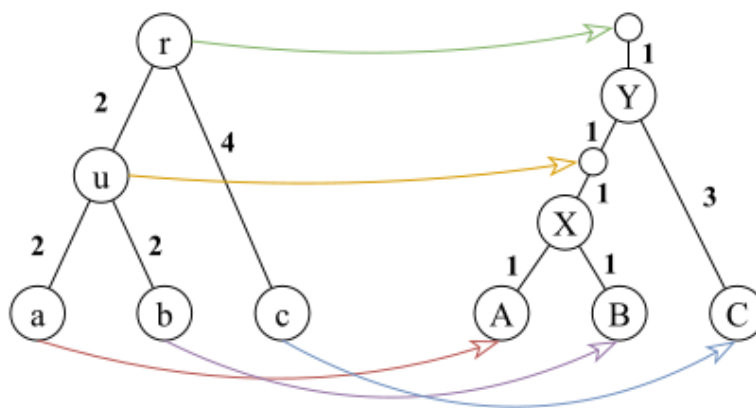
Rooted gene tree G



Gene tree 1

Node: c
Loss: 0
Node: b
Loss: 0
Node: a
Loss: 0
Node: u
Loss: 0
Node: r
Losses: 0 Duplications: 0

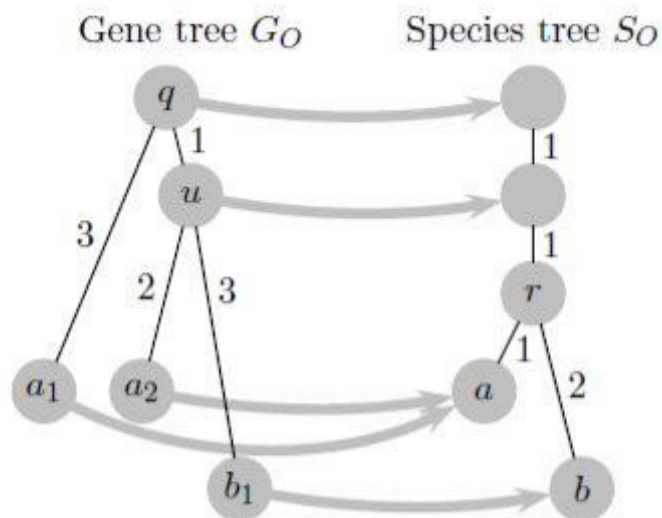
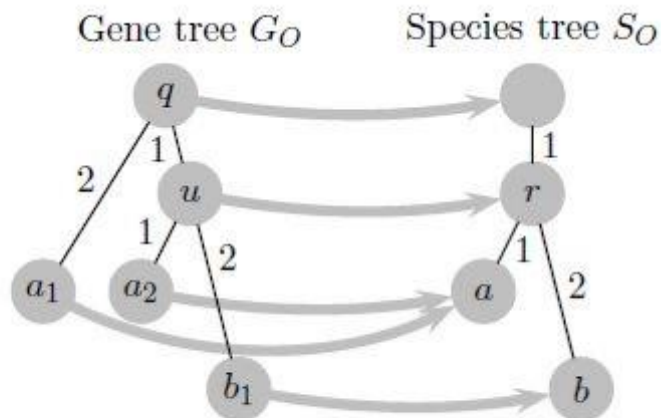
Rooted gene tree G



Gene tree 2

Node: c
Loss: 1 ← na ceste medzi "c" a "r" je S-node "Y"
Node: b
Loss: 1 ← na ceste medzi "b" a "u" je S-node "X"
Node: a
Loss: 1 ← na ceste medzi "a" a "u" je S-node "X"
Node: u
Loss: 1 ← na ceste medzi "u" a "r" je S-node "Y"
Duplication: 1 ← "u" sa namapuje nad "X" - jeho lca
Node: r
Duplication: 1 ← "r" sa namapuje nad "Y" - jeho lca
Losses: 4 Duplications: 2

Testing – Counting algorithm



Gene tree 1

Node: b1

Loss: 0

Node: a2

Loss: 0

Node: u

Loss: 0

Node: a1

Loss: 1 ← na ceste medzi "a1" a "q" je S-node "r"

Node: q

Duplication: 1 ← "q" sa namapuje nad "r"- jeho lca

Losses: 1 Duplications: 1

Gene tree 2

Node: b1

Loss: 1 ← na ceste medzi "b1" a "u" je S-node "r"

Node: a2

Loss: 1 ← na ceste medzi "a2" a "u" je S-node "r"

Node: u

Loss: 0

Duplication: 1 ← "u" sa namapuje nad "r"- jeho lca

Node: a1

Loss: 1 ← na ceste medzi "a1" a "q" je S-node "r"

Node: q

Duplication: 1 ← "q" sa namapuje nad "r"- jeho lca

Losses: 3 Duplications: 2

Thank you for your
attention.
