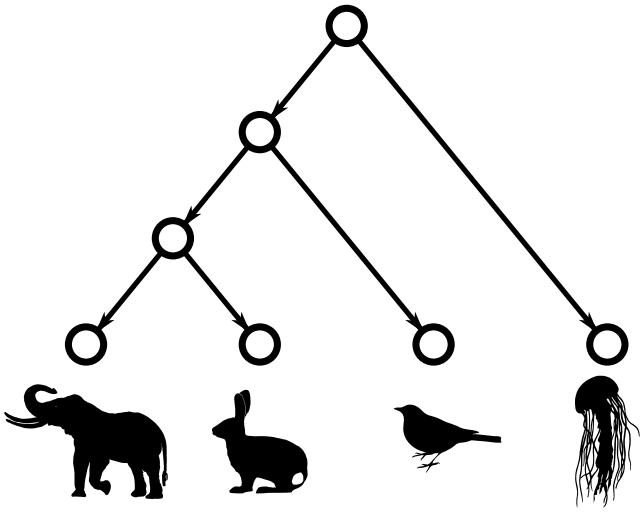
Stockholm University

LCA or no LCA:
A short story about simplifying networks

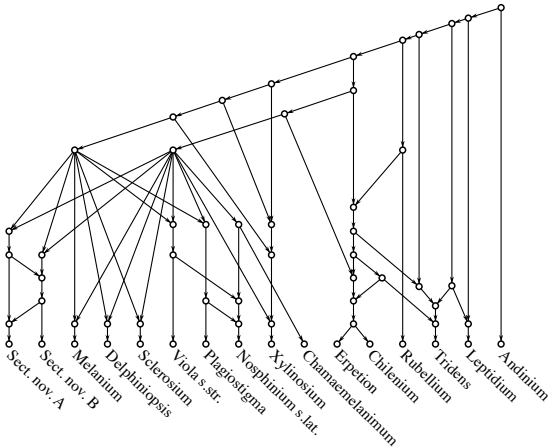
Anna Lindeberg
& Marc Hellmuth

Department of Mathematics
Stockholm University

Evolution: trees and networks



Evolution: trees and networks



Network of groups of *Viola* flowers*

However. . . Some species are not too keen on evolving in a tree-like fashion.

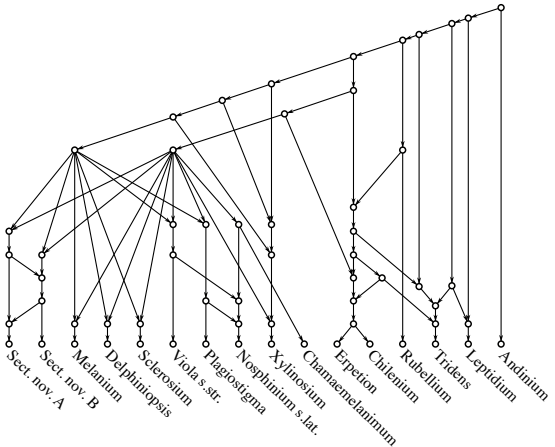
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

However. . . Some species are not too keen on evolving in a tree-like fashion.

Hybridization: species-monogamy is not always so important

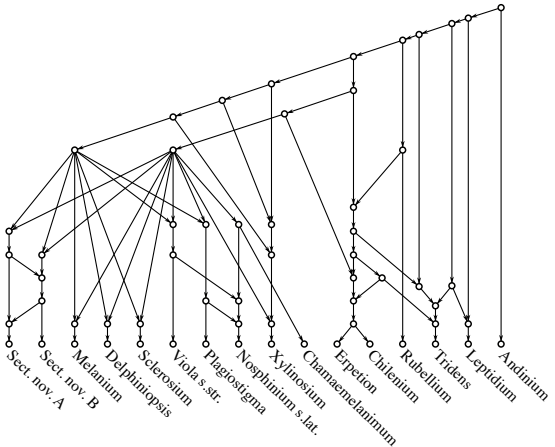
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

However. . . Some species are not too keen on evolving in a tree-like fashion.

Hybridization: species-monogamy is not always so important

Horizontal gene transfer: why not share some DNA?

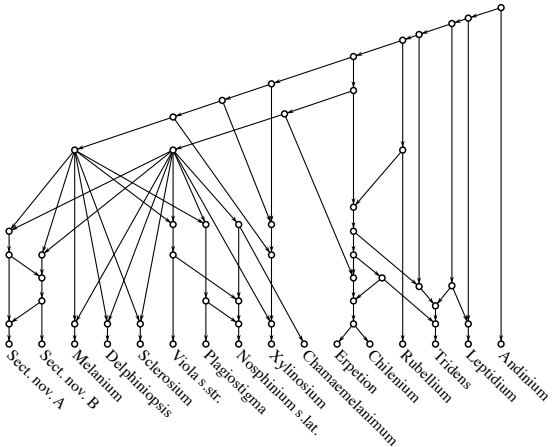
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

However... Some species are not too keen on evolving in a tree-like fashion.

Hybridization: species-monogamy is not always so important

Horizontal gene transfer: why not share some DNA?

Etc. ...

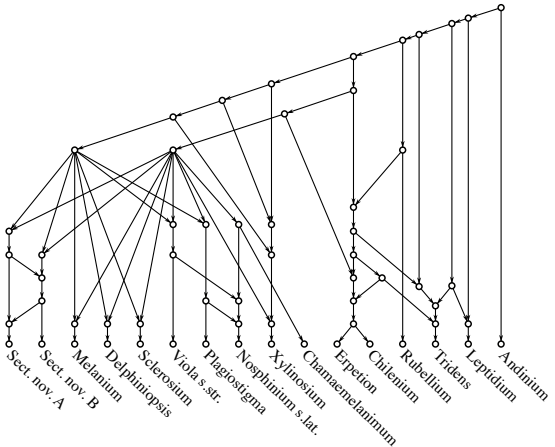
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

The evolutionary history is network-like!

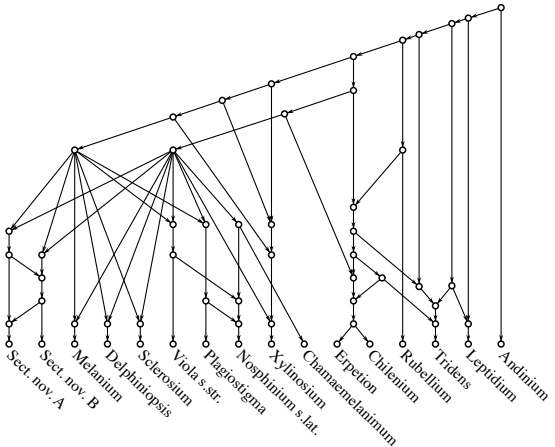
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

The evolutionary history is network-like!

How to obtain such networks?

Start with observable data

= genomic sequences of extant taxa

= set $L(N)$ of leaves in network N

Based on observable data, infer† the network N

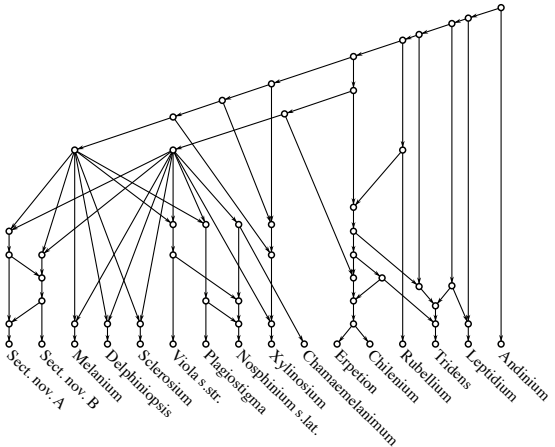
---

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Network of groups of *Viola* flowers*

Networks inferred from genomic data can . . .

. . . be highly complex and tangled

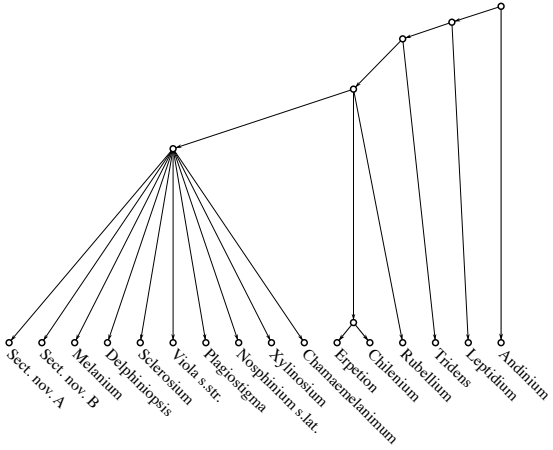
. . . contain information that is not supported by "observable data"

* From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

† A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

‡ Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Networks inferred from genomic data can ...

... be highly complex and tangled

... contain information that is not supported by "observable data"

Aim: methods to simplify networks while keeping main structural features

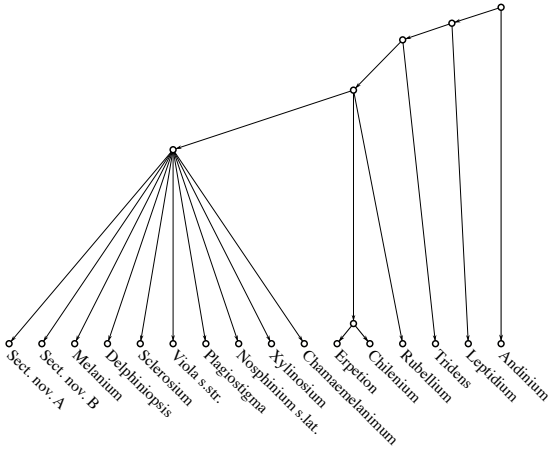
Lowest Stable Ancestor (LSA) tree to show "trend of evolution"[‡]

[*] From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

[†] A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

[‡] Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Evolution: trees and networks



Networks inferred from genomic data can . . .

. . . be highly complex and tangled

. . . contain information that is not supported by "observable data"

Aim: methods to simplify networks while keeping main structural features

The method we propose aims to simplify by eliminating vertices that are not supported by "observable data"

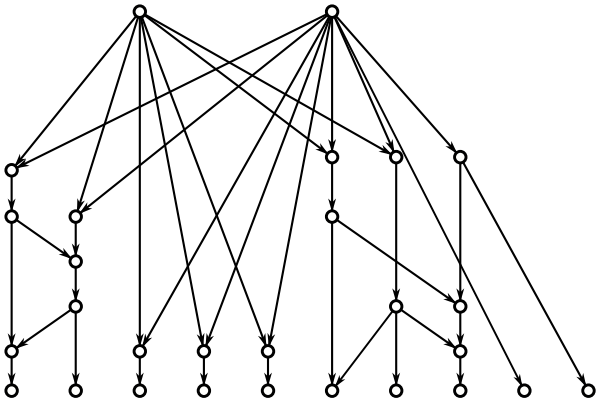
Lowest Stable Ancestor (LSA) tree to show "trend of evolution"‡

*From: Gene Trees to a Dated Allopolyploid Network: Insights from the Angiosperm Genus Viola (Violaceae), Marcussen et al., Syst. Biol., 2015

†A Survey of Combinatorial Methods for Phylogenetic Networks, Huson and Scornavacca, GBE, 2010

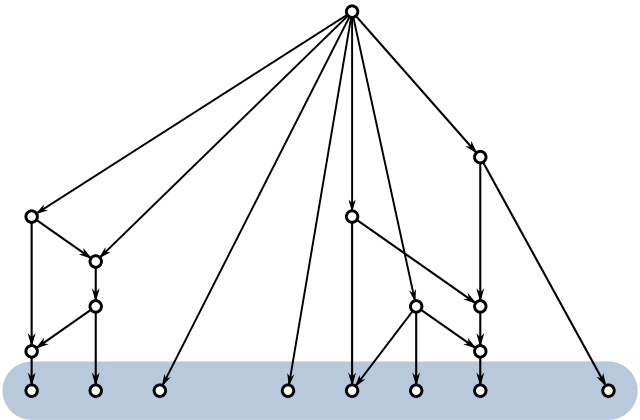
‡Transformations to Simplify Phylogenetic Networks, Heiss, Huson and Steel, Bulletin of Math. Bio., 2025

Absolutely necessary definitions



Directed Acyclic Graph, known as a DAG. The vertices are $V(G)$.

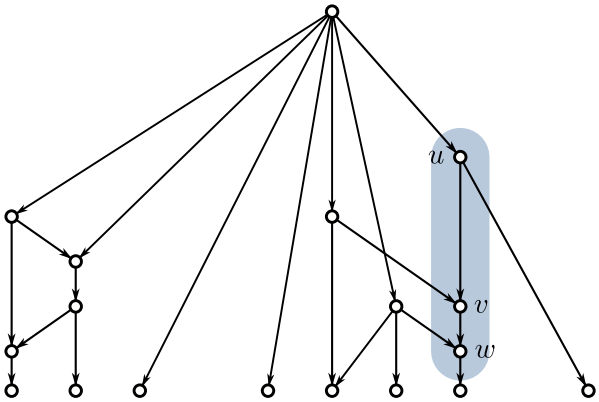
Absolutely necessary definitions



A network N is a DAG with a unique root (= source).

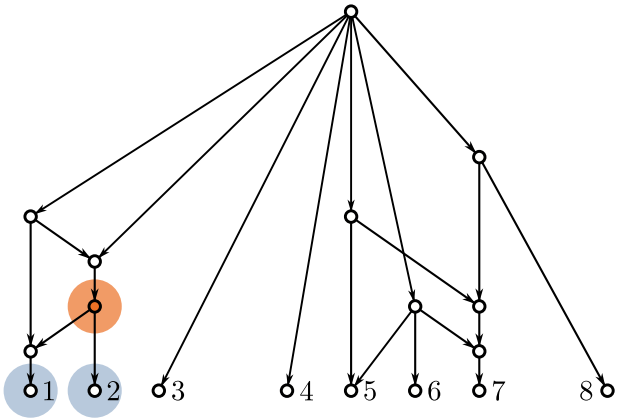
$L(N)$ denotes all leaves.

Absolutely necessary definitions



Ancestor and descendant: joined by directed path. Denoted $w \preceq u$.

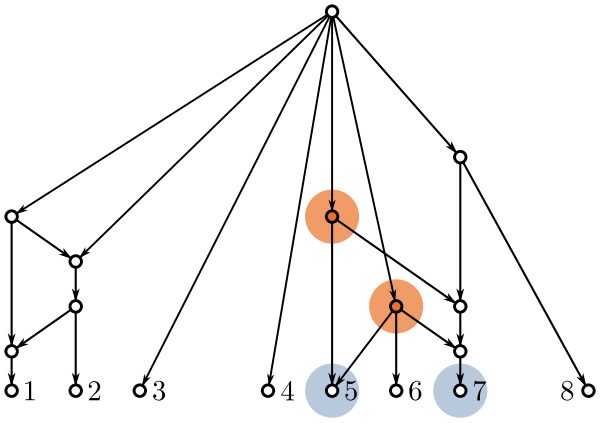
Absolutely necessary definitions



Least Common Ancestors: for $A \subseteq L(G)$, the set $\mathrm{LCA}(A)$ comprise all $\preceq$-minimal vertices that are ancestors of all leaves in A

$\mathrm{LCA}(\{1, 2\})$

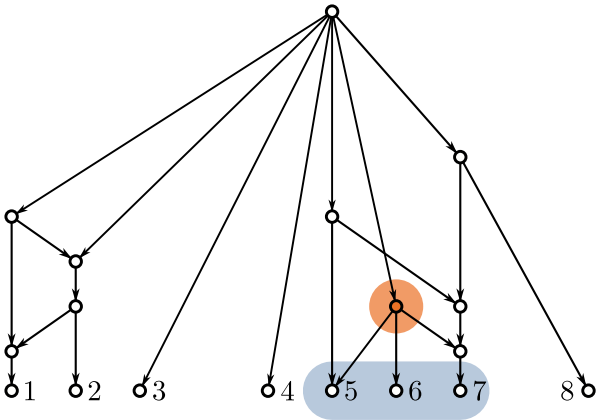
Absolutely necessary definitions



Least Common Ancestors: for $A \subseteq L(G)$, the set $\mathrm{LCA}(A)$ comprise all $\preceq$-minimal vertices that are ancestors of all leaves in A

$\mathrm{LCA}(\{5, 7\})$

Absolutely necessary definitions



Least Common Ancestors: for $A \subseteq L(G)$, the set $\mathrm{LCA}(A)$ comprise all $\preceq$-minimal vertices that are ancestors of all leaves in A

$\mathrm{LCA}(\{5, 6, 7\})$

LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

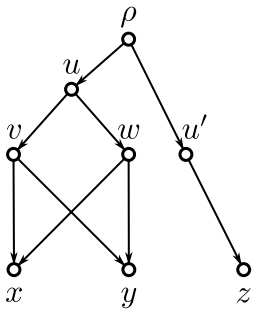
A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

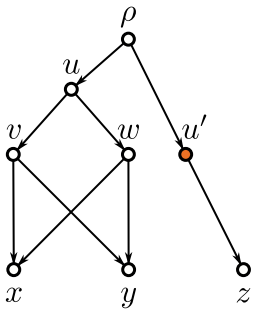


LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

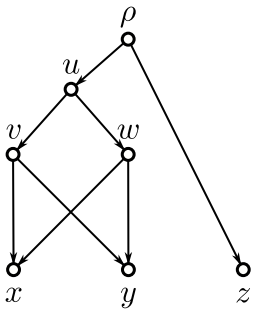


LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

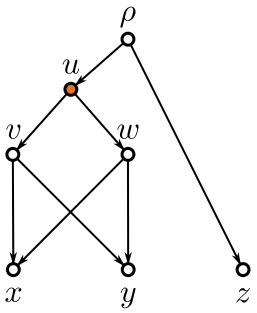


LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

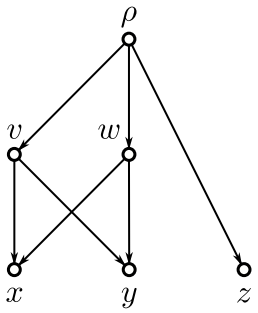


LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.



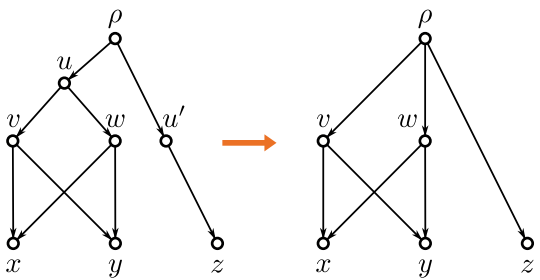
LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

- How to determine unsupported vertices?



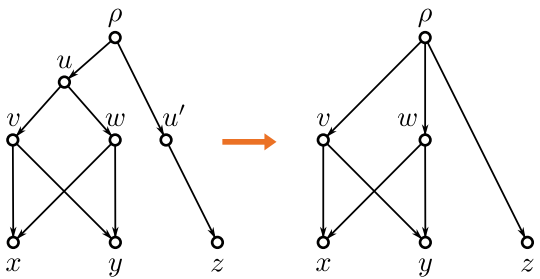
LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

- How to determine unsupported vertices?
- How to get rid of them?



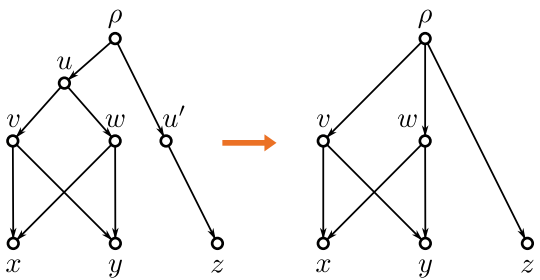
LCA-Relevant DAGs

We aim to **simplify** by removing vertices not supported by data.

A vertex v is an LCA-vertex (**= supported**), if v is an LCA of some subset of leaves.

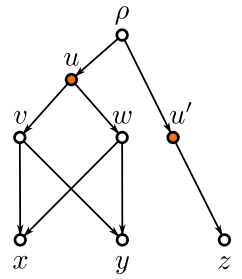
A DAG G is LCA-Relevant if every vertex is an LCA-vertex.

- How to determine unsupported vertices?
- How to get rid of them?
- Can we characterize LCA-Relevant DAGs?

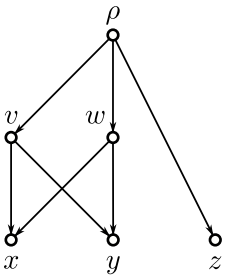


Characterization

- How to determine unsupported vertices?
- How to get rid of them?
- Can we characterize LCA-Relevant DAGs?



not LCA-Relevant



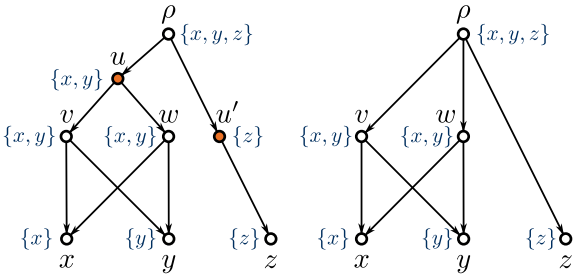
LCA-Relevant

Characterization

- How to determine unsupported vertices?
- How to get rid of them?
- Can we characterize LCA-Relevant DAGs?

Lemma

A vertex v is not an LCA-vertex $\iff$ v has a child u s.t. $\mathtt{C}(v) = \mathtt{C}(u)$



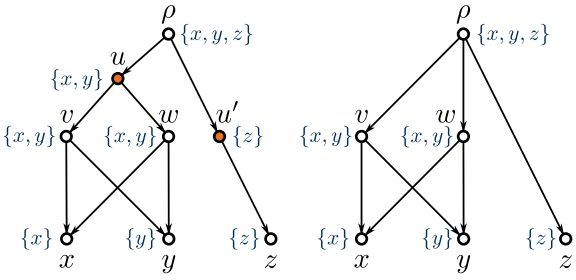
Characterization

- How to determine unsupported vertices?
- How to get rid of them?
- Can we characterize LCA-Relevant DAGs?

Lemma

A vertex v is not an LCA-vertex $\iff$ v has a child u s.t. $\mathtt{C}(v) = \mathtt{C}(u)$

All non-LCA-vertices can be determined in polynomial time.



Characterization

- How to determine unsupported vertices?
- How to get rid of them?
- Can we characterize LCA-RELEVANT DAGs?

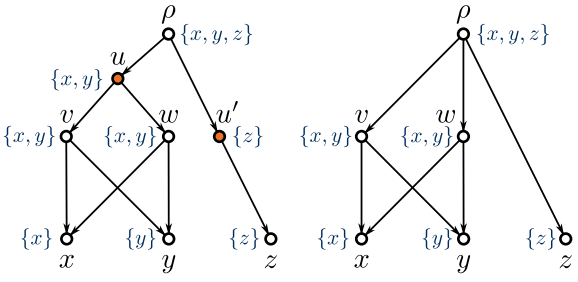
Lemma

A vertex v is not an LCA-vertex $\iff$ v has a child u s.t. $\mathtt{C}(v) = \mathtt{C}(u)$

All non-LCA-vertices can be determined in polynomial time.

Theorem

A DAG G is LCA$-$RELEVANT $\iff$ no two adjacent vertices of G have the same cluster



Characterization

- How to determine unsupported vertices? ✓
- How to get rid of them?
- Can we characterize LCA-RELEVANT DAGs? ✓

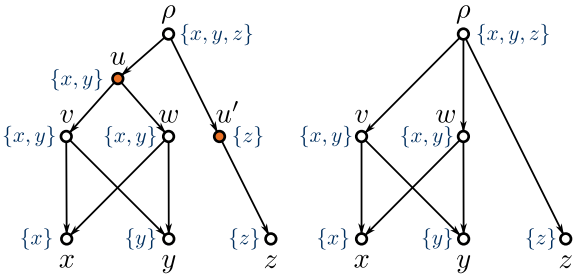
Lemma

A vertex v is not an LCA-vertex $\iff$ v has a child u s.t. $\mathtt{C}(v) = \mathtt{C}(u)$

All non-LCA-vertices can be determined in polynomial time.

Theorem

A DAG G is LCA$-$RELEVANT $\iff$ no two adjacent vertices of G have the same cluster



Characterization

- How to determine unsupported vertices? ✓
- **How to get rid of them?**
- Can we characterize LCA-RELEVANT DAGs? ✓

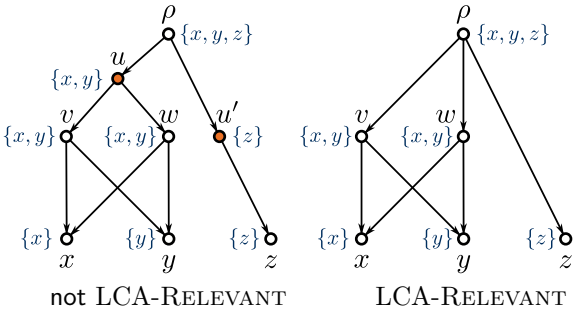
Lemma

A vertex v is not an LCA-vertex $\iff$ v has a child u s.t. $\mathtt{C}(v) = \mathtt{C}(u)$

All non-LCA-vertices can be determined in polynomial time.

Theorem

A DAG G is LCA$-$RELEVANT $\iff$ no two adjacent vertices of G have the same cluster

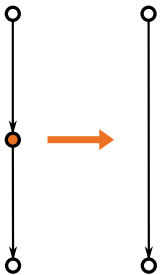


Simplification to LCA-Relevant

If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.

Simplification to LCA-Relevant

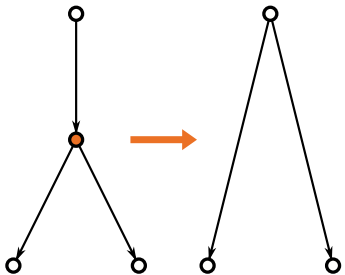
If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



$G \ominus v$ generalizes "vertex supression"

Simplification to LCA-Relevant

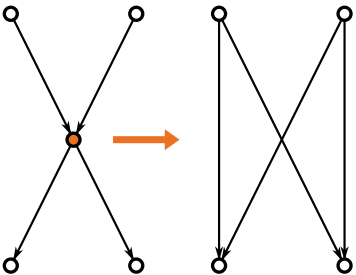
If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



$G \ominus v$ generalizes "vertex supression"

Simplification to LCA-Relevant

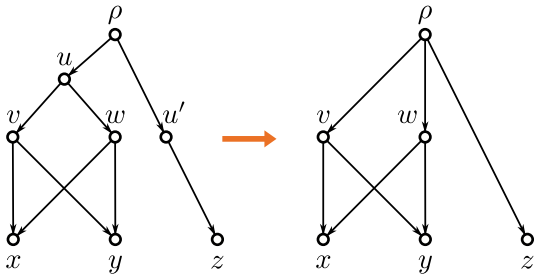
If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



$G \ominus v$ generalizes "vertex supression"

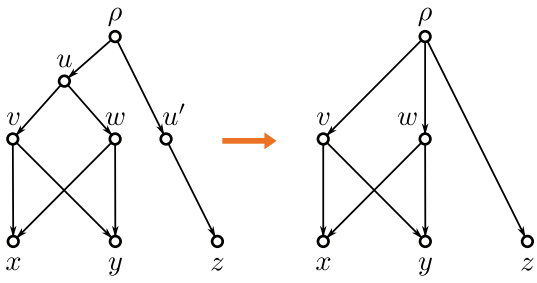
Simplification to LCA-Relevant

If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



Simplification to LCA-Relevant

If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



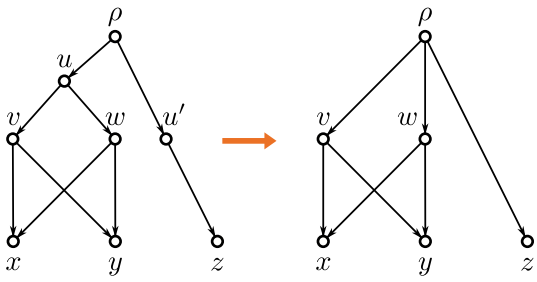
N $\qquad$ $N \ominus \{u, u'\}$

Theorem

If W is the set of all non-LCA vertices of G, then $G \ominus W$ is LCA-RELEVANT and satisfy

Simplification to LCA-Relevant

If v is a vertex of G, then $G \ominus v$ is obtained by adding an edge from each parent of v to each child of v, and then removing v.



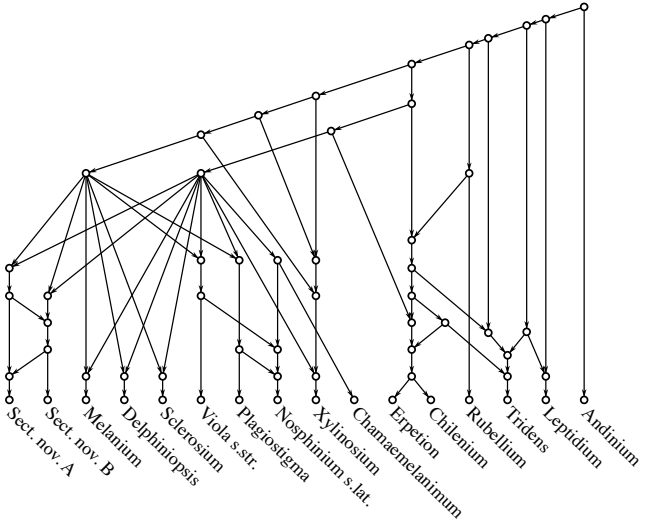
N $\qquad$ $N \ominus \{u, u'\}$

Theorem

If W is the set of all non-LCA vertices of G, then $G \ominus W$ is LCA-RELEVANT and satisfy

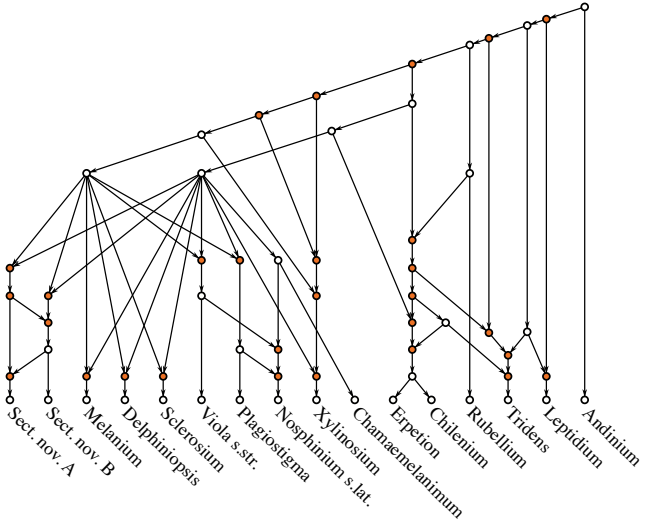
Well... a long list of appealing properties of what structure of G is kept

Bigger example



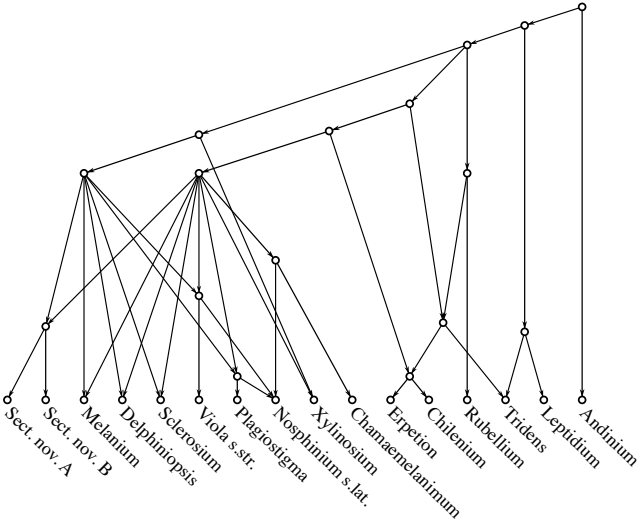
Network N

Bigger example



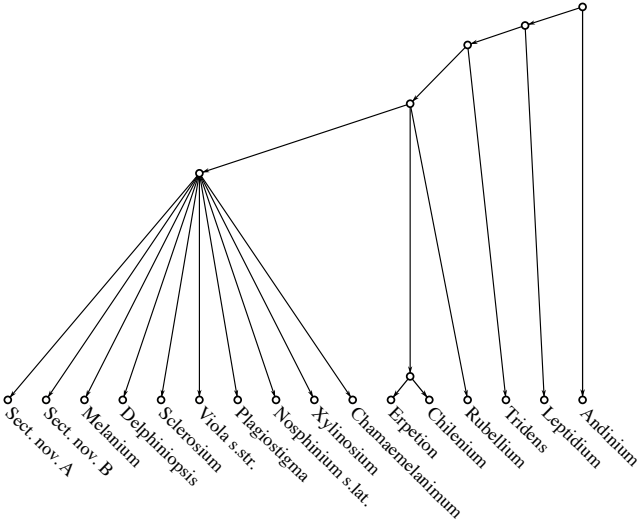
Network N with non-LCA-vertices highlighted

Bigger example



LCA-RELEVANT version $N \ominus W$ of N

Bigger example



"Trend of evolution" as captured by LSA-tree of N

Strengthen LCA-Relevant DAGs a bit

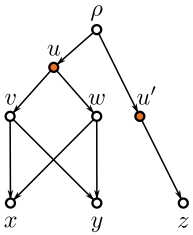
A DAG G is $\mathrm{LCA\text{-}Relevant}$ if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $v \in \mathrm{LCA}(A)$.

Strengthen LCA-Relevant DAGs a bit

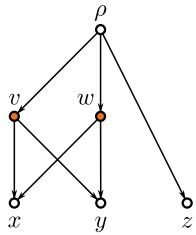
A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

Strengthen LCA-Relevant DAGs a bit

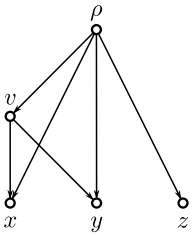
A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.



Network N



LCA-Relevant



LCA^u-Relevant

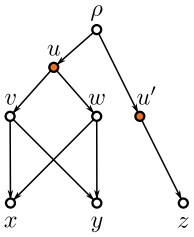
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

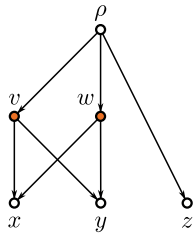
Characterization

Let G be a DAG. The following are equivalent:

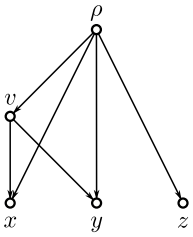
- G is LCA^u-Relevant



Network N



LCA-Relevant



LCA^u-Relevant

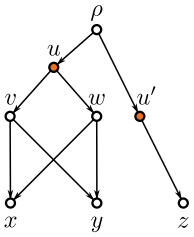
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

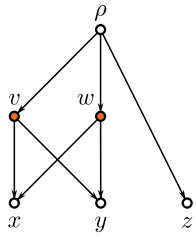
Characterization

Let G be a DAG. The following are equivalent:

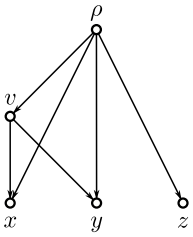
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)



Network N



LCA-Relevant



LCA^u-Relevant

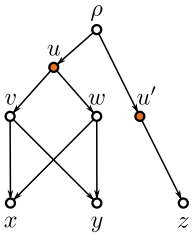
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

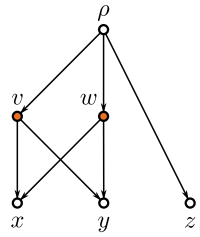
Characterization

Let G be a DAG. The following are equivalent:

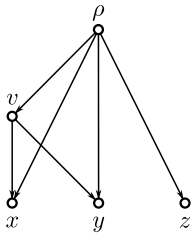
- G is LCA^u-RELEVANT
- G is LCA-RELEVANT and satisfy (PCC)



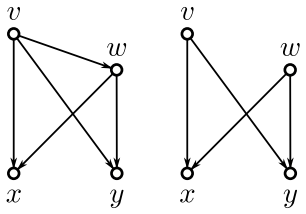
Network N



LCA-RELEVANT



LCA^u-RELEVANT



(PCC): there is a vw-path $\iff$ $\mathtt{C}(v)$ and $\mathtt{C}(w)$ are $\subseteq$–comparable.

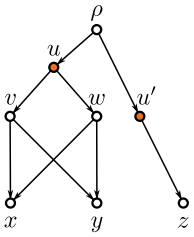
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

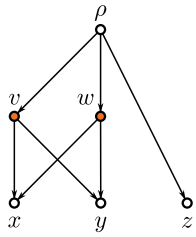
Characterization

Let G be a DAG. The following are equivalent:

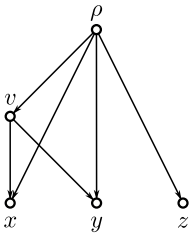
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



Network N



LCA-Relevant



LCA^u-Relevant

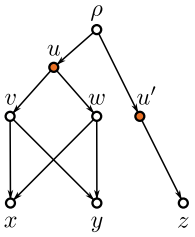
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

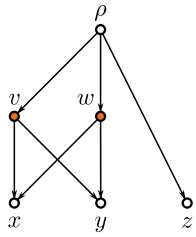
Characterization

Let G be a DAG. The following are equivalent:

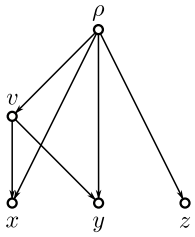
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



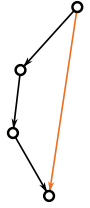
Network N



LCA-Relevant



LCA^u-Relevant



Shortcut: edge (u, v) for which there is some other uv-path

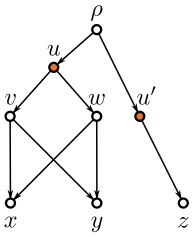
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

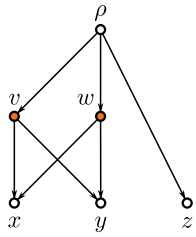
Characterization

Let G be a DAG. The following are equivalent:

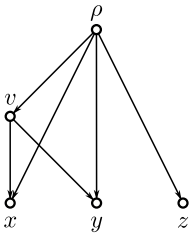
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



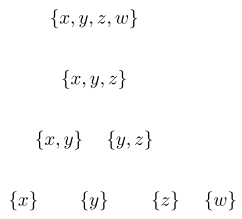
Network N



LCA-Relevant



LCA^u-Relevant



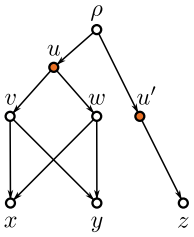
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

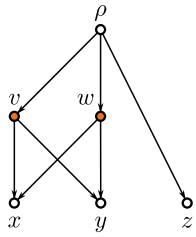
Characterization

Let G be a DAG. The following are equivalent:

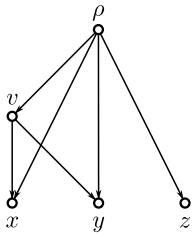
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



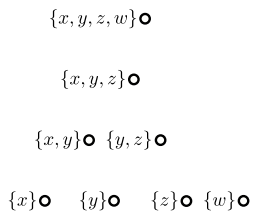
Network N



LCA-Relevant



LCA^u-Relevant



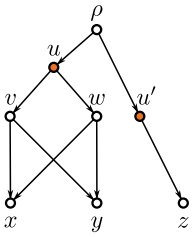
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

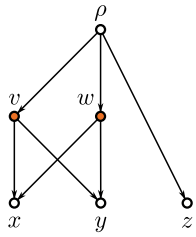
Characterization

Let G be a DAG. The following are equivalent:

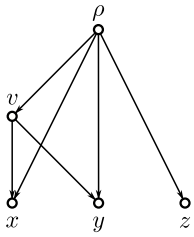
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



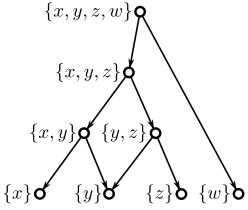
Network N



LCA-Relevant



LCA^u-Relevant



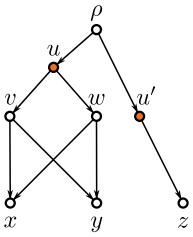
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

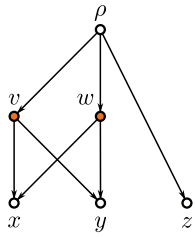
Characterization

Let G be a DAG. The following are equivalent:

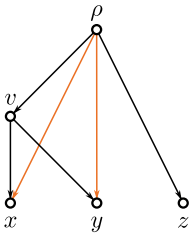
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



Network N



LCA-Relevant



LCA^u-Relevant

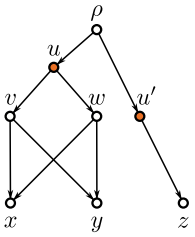
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

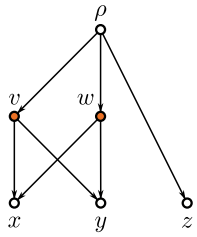
Characterization

Let G be a DAG. The following are equivalent:

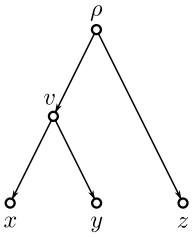
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



Network N



LCA-Relevant



LCA^u-Relevant

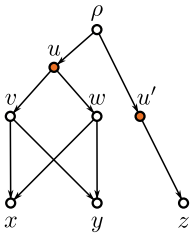
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

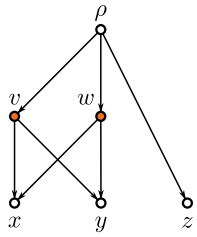
Characterization

Let G be a DAG. The following are equivalent:

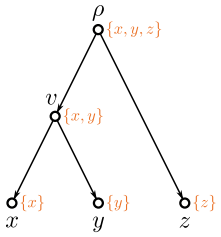
- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$



Network N



LCA-Relevant



LCA^u-Relevant

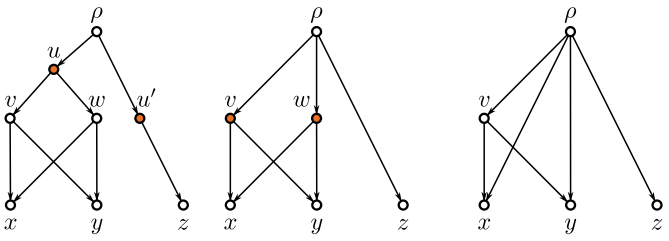
Strengthen LCA-Relevant DAGs a bit

A DAG G is LCA^u-Relevant if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $\{v\} = \mathrm{LCA}(A)$.

Characterization

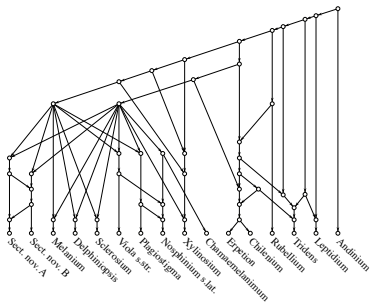
Let G be a DAG. The following are equivalent:

- G is LCA^u-Relevant
- G is LCA-Relevant and satisfy (PCC)
- If you remove all shortcuts from G, what you obtain is isomorphic to the Hasse diagram of $\mathfrak{C}_G$

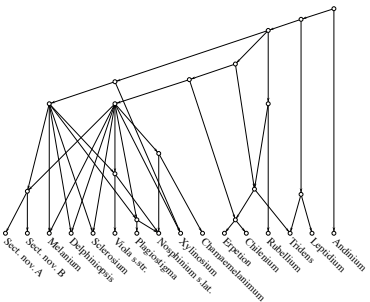


Computing an LCA^u-Relevant version of any DAG G can be done in polynomial-time

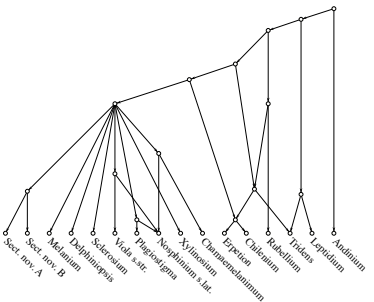
Bigger example (again)



Viola-flower network N



LCA-RELEVANT version



LCA^u-RELEVANT version

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.
- Our results also fit rather nicely together with an axiomatic framework recently introduced in the context of another type of simplification (resulting in so-called LSA-trees).

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.
- Our results also fit rather nicely together with an axiomatic framework recently introduced in the context of another type of simplification (resulting in so-called LSA-trees).

Still lots of interesting open questions:

- For $G \ominus W$ being $\mathrm{LCA}-$RELEVANT the set W is uniquely determined and $\mathfrak{C}_{G \ominus W} = \mathfrak{C}_G$. For $G \ominus W$ being LCA^u–RELEVANT the set W is *not* uniquely determined and $\mathfrak{C}_{G \ominus W} \subsetneq \mathfrak{C}_G$ is possible. How to minimize $|W|$? Or minimize $|\mathfrak{C}_G \setminus \mathfrak{C}_{G \ominus W}|$?

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.
- Our results also fit rather nicely together with an axiomatic framework recently introduced in the context of another type of simplification (resulting in so-called LSA-trees).

Still lots of interesting open questions:

- For $G \ominus W$ being $\mathrm{LCA}-$RELEVANT the set W is uniquely determined and $\mathfrak{C}_{G \ominus W} = \mathfrak{C}_G$. For $G \ominus W$ being LCA^u–RELEVANT the set W is *not* uniquely determined and $\mathfrak{C}_{G \ominus W} \subsetneq \mathfrak{C}_G$ is possible. How to minimize $|W|$? Or minimize $|\mathfrak{C}_G \setminus \mathfrak{C}_{G \ominus W}|$?
- Is it too much to require vertices to be LCA's of leaves?

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.
- Our results also fit rather nicely together with an axiomatic framework recently introduced in the context of another type of simplification (resulting in so-called LSA-trees).

Still lots of interesting open questions:

- For $G \ominus W$ being $\mathrm{LCA}-$RELEVANT the set W is uniquely determined and $\mathfrak{C}_{G \ominus W} = \mathfrak{C}_G$. For $G \ominus W$ being LCA^u–RELEVANT the set W is *not* uniquely determined and $\mathfrak{C}_{G \ominus W} \subsetneq \mathfrak{C}_G$ is possible. How to minimize $|W|$? Or minimize $|\mathfrak{C}_G \setminus \mathfrak{C}_{G \ominus W}|$?
- Is it too much to require vertices to be LCA's of leaves?

Other results and open problems

- Generally worked with a generalization: a DAG is $\mathcal{I}$-LCA-RELEVANT if for every vertex $v \in V(G)$ there is some $A \subseteq L(G)$ s.t. $|A| \in \mathcal{I}$ and $v \in \mathrm{LCA}(A)$.
- Our results also fit rather nicely together with an axiomatic framework recently introduced in the context of another type of simplification (resulting in so-called LSA-trees).

Still lots of interesting open questions:

- For $G \ominus W$ being $\mathrm{LCA}-$RELEVANT the set W is uniquely determined and $\mathfrak{C}_{G \ominus W} = \mathfrak{C}_G$. For $G \ominus W$ being LCA^u–RELEVANT the set W is *not* uniquely determined and $\mathfrak{C}_{G \ominus W} \subsetneq \mathfrak{C}_G$ is possible. How to minimize $|W|$? Or minimize $|\mathfrak{C}_G \setminus \mathfrak{C}_{G \ominus W}|$?
- Is it too much to require vertices to be LCA's of leaves? Recursive definition:
  Every leaf in G is pertinent.
  A vertex is pertinent if it is the LCA of pertinent vertices.
  What is the structure of DAGs in which all vertices are pertinent and how to determine such vertices?

Joint work with

Marc Hellmuth

Joint work with

Marc Hellmuth

Thank you!

Joint work with

Marc Hellmuth

Thank you!

Want to know more? See: Lindeberg & Hellmuth, Simplifying and Characterizing DAGs and Phylogenetic Networks via Least Common Ancestor Constraints, Bulletin of Math. Bio. (2025)