Reordering a tree according to an order on its leaves

L. Bulteau[1], P. Gambette[1], O. Seminck[2]

[1] LIGM, CNRS, Université Gustave Eiffel, France
[2] Lattice, CNRS & ENS/PSL & Université Sorbonne nouvelle, France

LIGM - 2022-06-28

Introduction

Initial Motivation

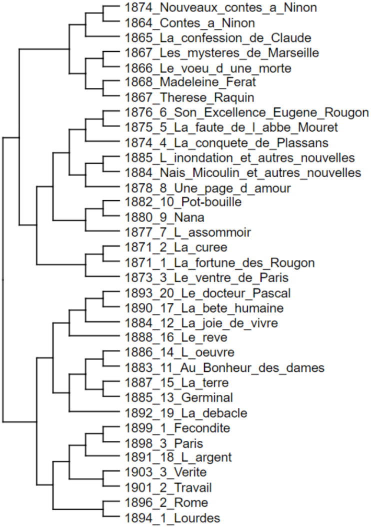
Linguistic Question:
How does an author style evolve through time?

Initial Motivation

Linguistic Question:
How does an author style evolve through time?

- novels are clustered by **linguistic criteria** (word and phrase frequencies, etc.)

→ *dendrogram*



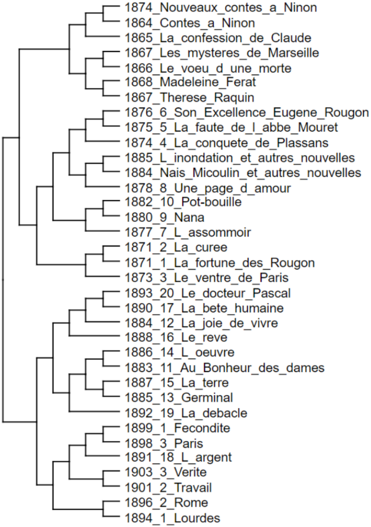
Initial Motivation

Linguistic Question:
How does an author style evolve through time?

- novels are clustered by **linguistic criteria** (word and phrase frequencies, etc.)

$\rightarrow$ *dendrogram*

- does the clustering group together novels published in **consecutive years**?



Modelization

Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements

D

A E

B F

G

C

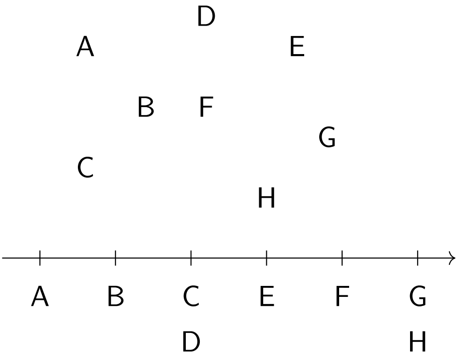
H

Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements
- Ordering (time-line, ...)

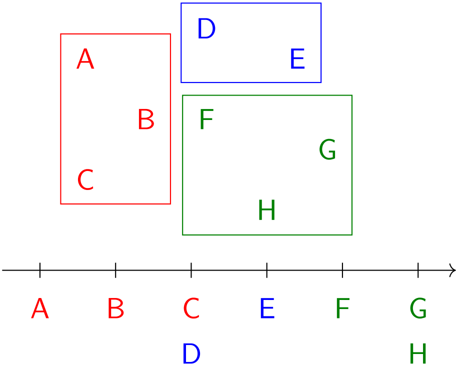


Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements
- Ordering (time-line, ...)
- Clustering



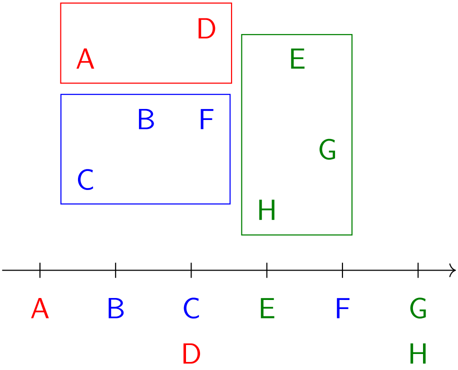
Is the clustering consistent with the ordering?

Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements
- Ordering (time-line, ...)
- Clustering



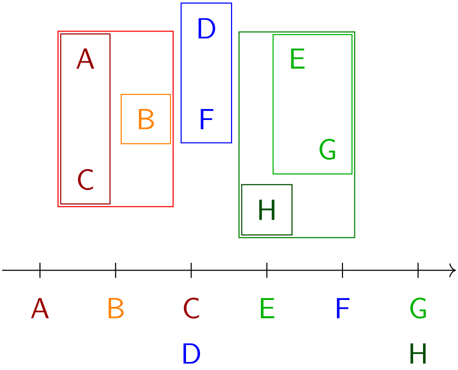
Is the clustering consistent with the ordering?

Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements
- Ordering (time-line, ...)
- Hierarchical Clustering



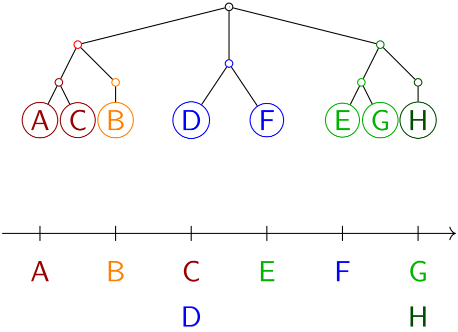
Is the clustering consistent with the ordering?

Motivation

Is a clustering consistent with an external ordering?

Input:

- Elements
- Ordering (time-line, ...)
- Hierarchical Clustering (seen as a tree / dendrogram)

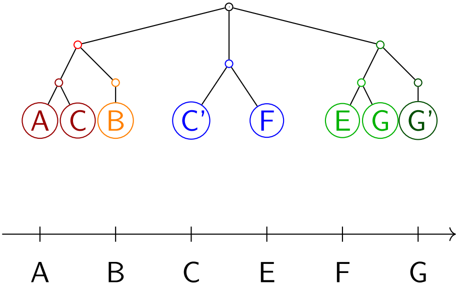


Motivation

Is a clustering consistent with an external ordering?

Input:

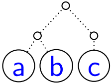
- Elements
- Non-strict Ordering (time-line, ...)
- Hierarchical Clustering (seen as a tree / dendrogram)



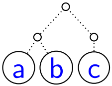
Definitions

- Tree T with leaf set X, ordering $\sigma \colon X \to \mathbb{N}$ (weak order $\leq_\sigma$)

Definitions

- Tree T with leaf set X, ordering σ: $X \to \mathbb{N}$ (weak order $\leq_\sigma$)
- Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and 

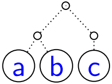
Definitions

- Tree T with leaf set X, ordering σ: $X \to \mathbb{N}$ (weak order $\leq_\sigma$)
- Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and 

OTDE One-Tree Drawing by Deleting Edges

Given T, σ, k,
Find $X' \subseteq X$, $|X'| \geq |X| - k$
Such that $T[X']$ has no conflict with σ

Definitions

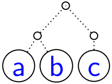
- Tree T with leaf set X, ordering σ: $X \to \mathbb{N}$ (weak order $\leq_\sigma$)
- Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and 

- Ordering of T: strict order σ' without conflict with T
  $\Leftrightarrow$ permute the children of each node, read leaves from left to right

TTDE Two-Tree Drawing by Deleting Edges

Given T_1, T_2, k,
Find $X' \subseteq X$, $|X'| \geq |X| - k$,
and an ordering σ of both $T_1[X']$ and $T_2[X']$

Definitions

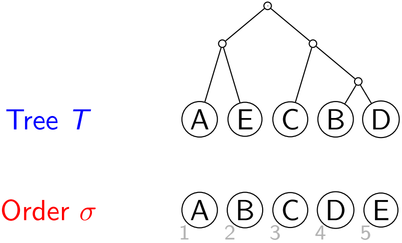
- Tree T with leaf set X, ordering σ: $X \to \mathbb{N}$ (weak order $\leq_\sigma$)
- Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and 

- Ordering of T: strict order σ' without conflict with T
  $\Leftrightarrow$ permute the children of each node, read leaves from left to right
- Crossing between σ and σ': pair $\{a, b\}$ with $a <_\sigma b$ and $b <_{\sigma'} a$

OTCM One-Tree Crossing Minimization

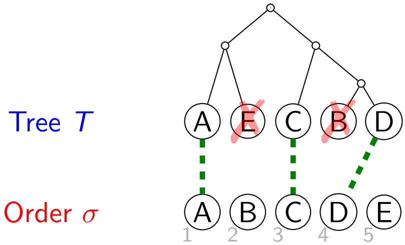
Given T, σ, k,
Find σ' ordering of T
Such that σ' has at most k crossings with σ

Example



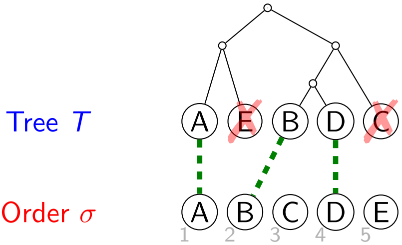
Input instance

Example



Score for OTDE: $k = 2$ deletions

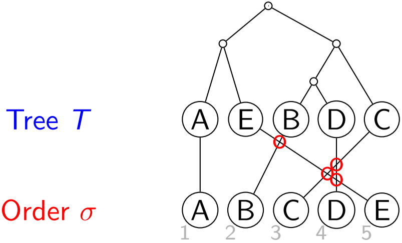
Example



Another solution with the same score

fun fact: all possible permutations of each node's children need 2 deletions

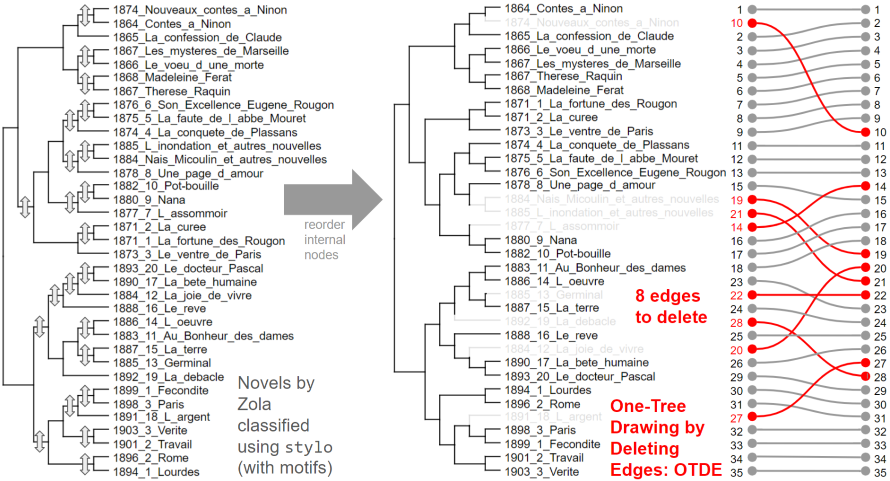
Example



Score for OTCM: 4 crossings

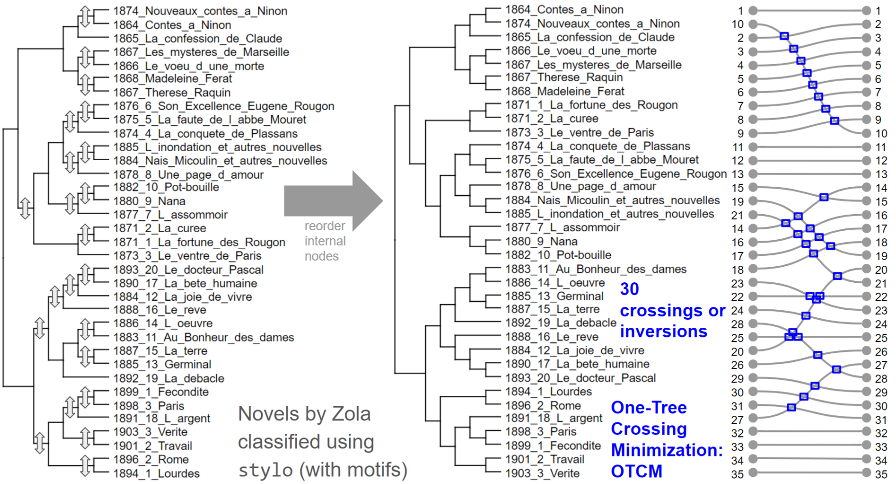
Initial Question

How much is the dendrogram consistent with time?



Initial Question

How much is the dendrogram consistent with time?



Previous Results

OTCM on binary trees

Most studied variant, from phylogenetics

- Dwyer, Schreiber '04: $O(n^2)$
- Fernau, Kaufmann, Poths '05: $O(n \log^2 n)$
- Bansal et al. '09: $O(n \log^2 n / \log \log n)$
- Fernau, Kaufmann, Poths. '10 and Venkatachalam, et al. '10: $O(n \log n)$

Previous Results

OTCM on binary trees

Most studied variant, from phylogenetics

- Dwyer, Schreiber '04: $O(n^2)$
- Fernau, Kaufmann, Poths '05: $O(n \log^2 n)$
- Bansal et al. '09: $O(n \log^2 n / \log \log n)$
- Fernau, Kaufmann, Poths. '10 and Venkatachalam, et al. '10: $O(n \log n)$

OTDE, TTDE

Introduced by Fernau et al.:

- Reduction **from** OTDE **to** 3-Hitting Set
- NP-hardness still open

Our Results

With arbitrary-degree trees

OTCM

- NP-hardness (from Feedback Arc Set)

Our Results

With arbitrary-degree trees

OTCM

- NP-hardness (from Feedback Arc Set)

OTDE

- NP-hardness (from Independent Set)
- Parameterized algorithms
  - (simple) XP for the degree d [1]
  - (advanced) FPT for the *deletion-degree* ∂ [2]

[1] $O(d!n^{d+2})$
[2] $O(d^2 2^{\partial} n^4)$ with $\partial =$ degree of $T[X \setminus X']$, $\partial \leq \min\{d, k\}$

Our Results

With arbitrary-degree trees

OTCM

- NP-hardness (from Feedback Arc Set)

OTDE

- NP-hardness (from Independent Set)
- Parameterized algorithms
  - (simple) XP for the degree d [1]
  - (advanced) FPT for the *deletion-degree* ∂ [2]

TTDE

- NP-hardness (from OTDE)

---

[1] $O(d! n^{d+2})$

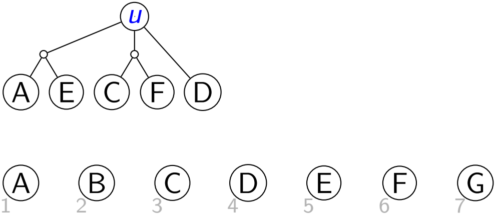
[2] $O(d^2 2^{\partial} n^4)$ with $\partial =$ degree of $T[X \setminus X']$, $\partial \leq \min\{d, k\}$

Algorithms

OTDE is XP for the degree

Bottom-up Dynamic Programming

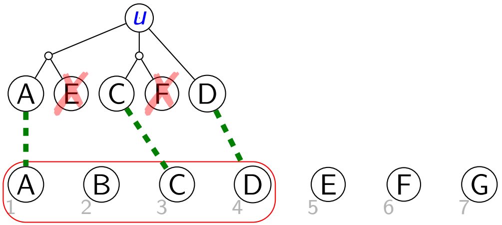
For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$

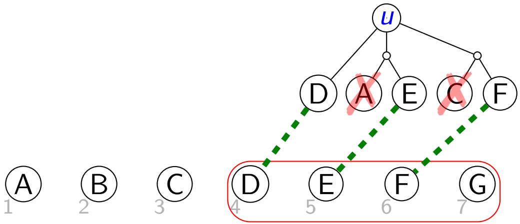


$X(u, 1, 4) = 2$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$

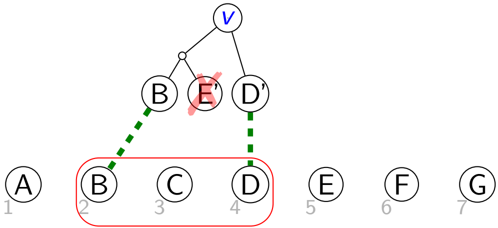


$X(u, 1, 4) = 2$
$X(u, 4, 7) = 2$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



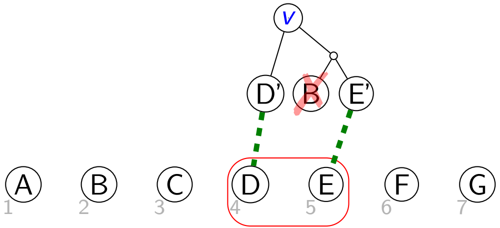
$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$

$X(u, 4, 7) = 2$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



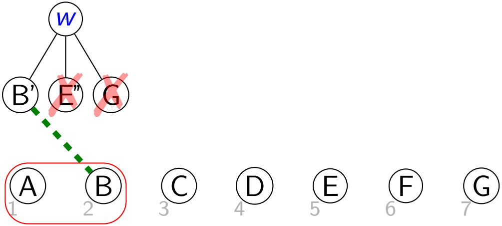
$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$

$X(u, 4, 7) = 2$ $\quad$ $X(v, 4, 5) = 1$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$

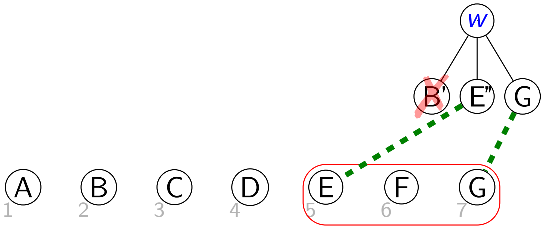


$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$ $\quad$ $X(w, 1, 2) = 2$
$X(u, 4, 7) = 2$ $\quad$ $X(v, 4, 5) = 1$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



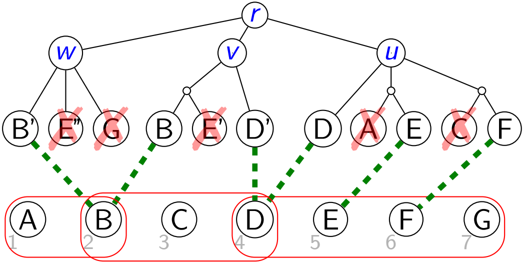
$X(u, 1, 4) = 2$ $\quad X(v, 2, 4) = 1$ $\quad X(w, 1, 2) = 2$

$X(u, 4, 7) = 2$ $\quad X(v, 4, 5) = 1$ $\quad X(w, 5, 7) = 1$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



$X(u, 1, 4) = 2$ $\qquad$ $X(v, 2, 4) = 1$ $\qquad$ $X(w, 1, 2) = 2$

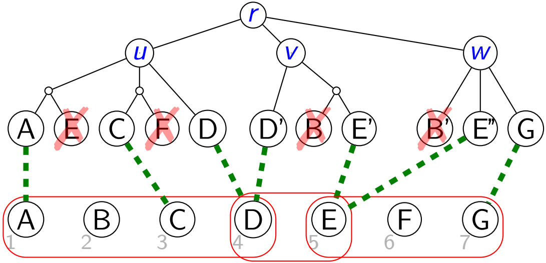
$X(u, 4, 7) = 2$ $\qquad$ $X(v, 4, 5) = 1$ $\qquad$ $X(w, 5, 7) = 1$

$X(r, 1, 7) = \min(5, \ldots)$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$ $\quad$ $X(w, 1, 2) = 2$

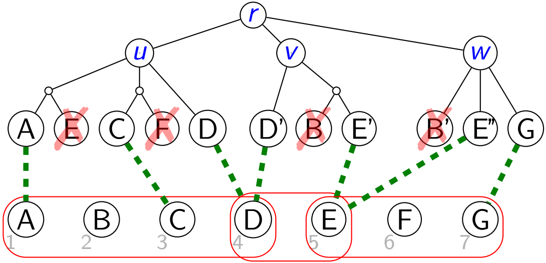
$X(u, 4, 7) = 2$ $\quad$ $X(v, 4, 5) = 1$ $\quad$ $X(w, 5, 7) = 1$

$X(r, 1, 7) = \min(5, 4, \ldots)$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$\boxed{X(u, 1, 4) = 2}$ $X(v, 2, 4) = 1$ $X(w, 1, 2) = 2$

$X(u, 4, 7) = 2$ $\boxed{X(v, 4, 5) = 1}$ $\boxed{X(w, 5, 7) = 1}$

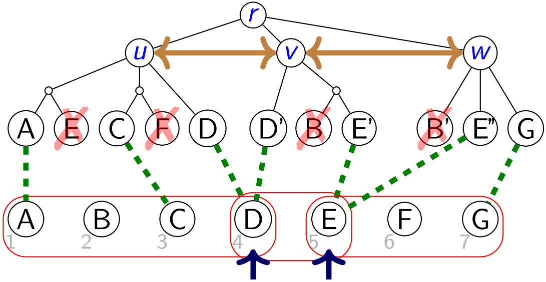
$X(r, 1, 7) = \min(5, 4, \ldots)$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r

$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$d!$ permutations of the children

n^{d-1} pivots

$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$ $\quad$ $X(w, 1, 2) = 2$

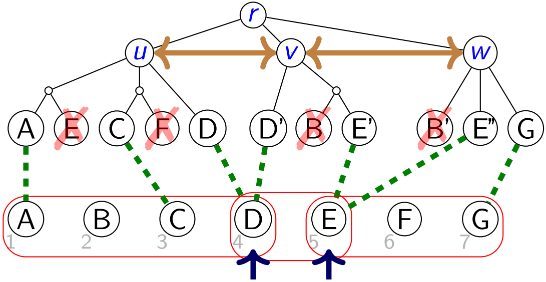
$X(u, 4, 7) = 2$ $\quad$ $X(v, 4, 5) = 1$ $\quad$ $X(w, 5, 7) = 1$

$X(r, 1, 7) = \min(5, 4, \ldots)$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v, interval l, r
$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$d!$ permutations of the children

n^{d-1} pivots

Overall $O(d!n^{d+2})$

$X(u, 1, 4) = 2$ $\quad$ $X(v, 2, 4) = 1$ $\quad$ $X(w, 1, 2) = 2$

$X(u, 4, 7) = 2$ $\quad$ $X(v, 4, 5) = 1$ $\quad$ $X(w, 5, 7) = 1$

$X(r, 1, 7) = \min(5, 4, \ldots)$

OTDE is FPT for the deletion degree

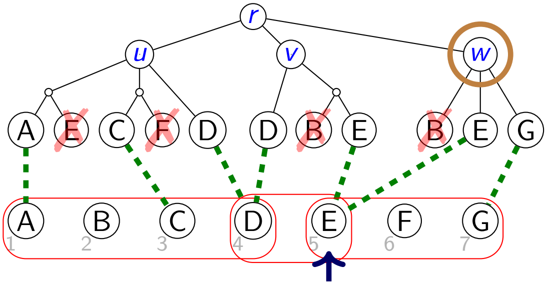
From XP to FPT ($n^{f(d)} \rightarrow f(d)n^c$)

- augment the DP table with sets of children,
- progress one pivot at a time

OTDE is FPT for the deletion degree

From XP to FPT ($n^{f(d)} \to f(d)n^c$)

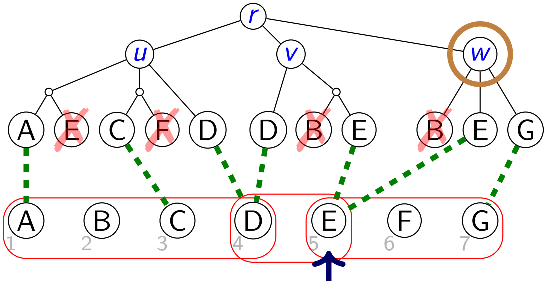
- augment the DP table with sets of children,
- progress one pivot at a time



OTDE is FPT for the deletion degree

From XP to FPT ($n^{f(d)} \to f(d)n^c$)

- augment the DP table with sets of children,
- progress one pivot at a time

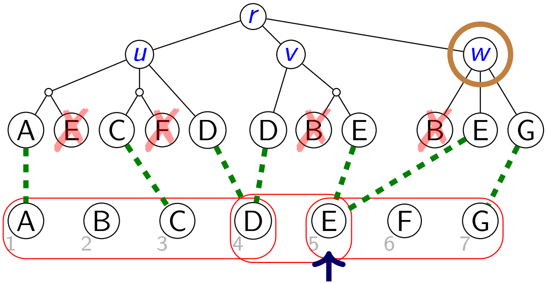


$$
\begin{aligned}
X(\{r\},1,7) &= X(\{u,v,w\},1,7) \\
&= X(\{u,v\},1,5) + X(\{w\},5,7)
\end{aligned}
$$

OTDE is FPT for the deletion degree

From XP to FPT ($n^{f(d)} \to f(d)n^c$)

- augment the DP table with sets of children,
- progress one pivot at a time



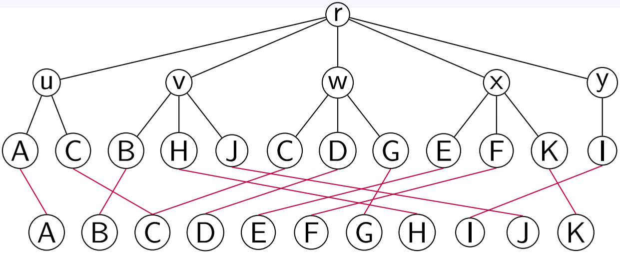
$$X(\{r\},1,7) = X(\{u,v,w\},1,7)$$
$$= X(\{u,v\},1,5) + X(\{w\},5,7)$$

Table size: $2^d n^3$, marginalization in $O(dn)$, overall: $O(d2^d n^4)$

OTDE is FPT for the deletion degree

From degree to deletion-degree

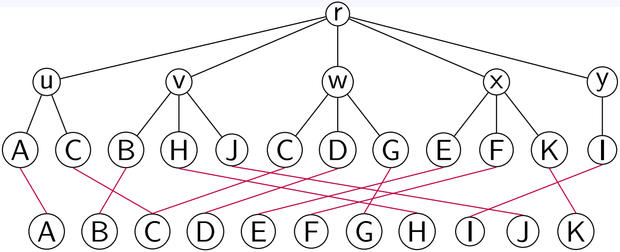
- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict

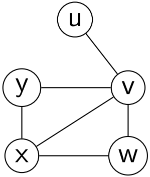


OTDE is FPT for the deletion degree

From degree to deletion-degree

- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict
- compute some backbone using Vertex Cover

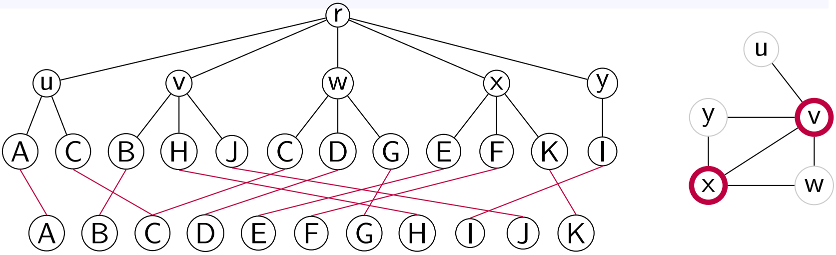




OTDE is FPT for the deletion degree

From degree to deletion-degree

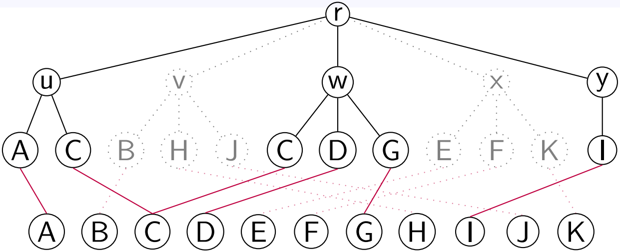
- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict
- compute some backbone using Vertex Cover
  VC$=\{x, v\} \to$ backbone $=(u, w, y)$

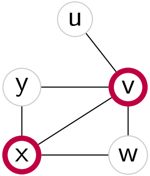


OTDE is FPT for the deletion degree

From degree to deletion-degree

- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict
- compute some backbone using Vertex Cover
  VC$=\{x, v\} \rightarrow$ backbone $=(u, w, y)$

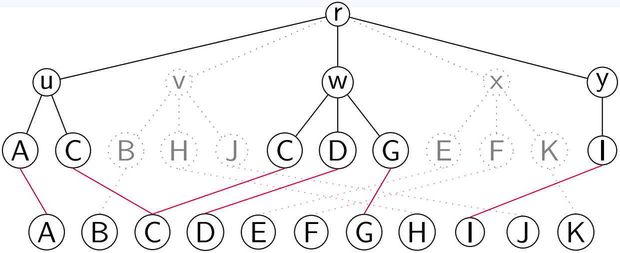


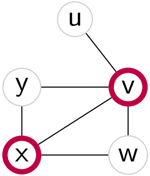


OTDE is FPT for the deletion degree

From degree to deletion-degree

- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict
- compute some backbone using Vertex Cover
  VC$=\{x, v\} \to$ backbone $=(u, w, y)$
- compute DP entries for each
  (prefix of the backbone) $\cup$ (any vertices out of the backbone)

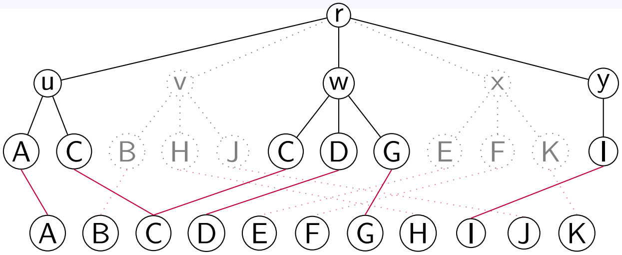


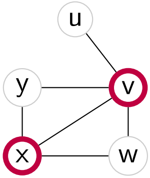


OTDE is FPT for the deletion degree

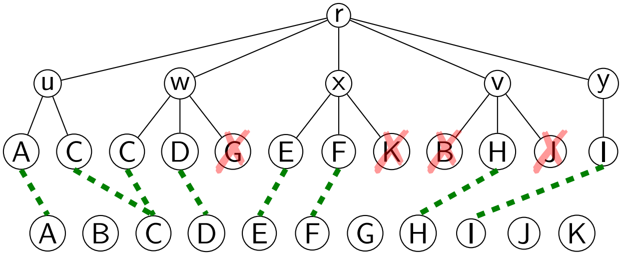
From degree to deletion-degree

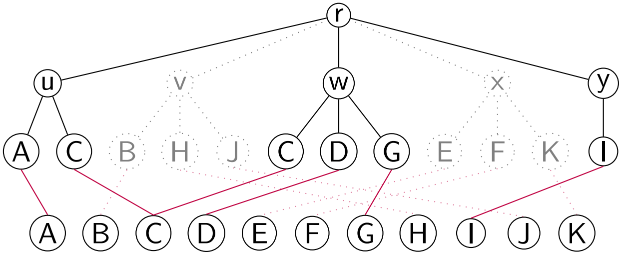
- only $\partial \ll d$ children with a deletion
- there exists a large backbone without self-conflict
- compute some backbone using Vertex Cover
  VC$=\{x, v\} \to$ backbone $=(u, w, y)$
- compute DP entries for each
  (prefix of the backbone) $\cup$ (any vertices out of the backbone)
- $2^d \to d2^\partial$ (+ VC preprocessing in $O(1.3^\partial d + \partial d^2)$)

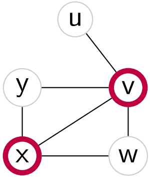




OTDE is FPT for the deletion degree



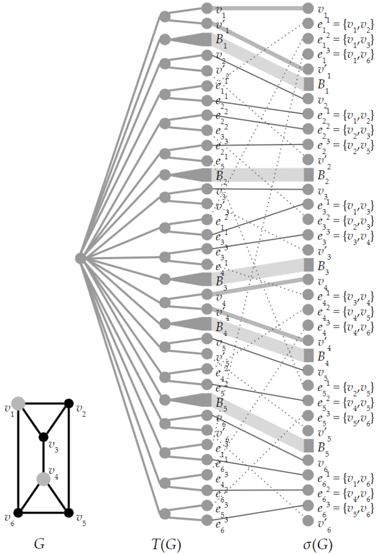




Hardness Results

OTDE is NP-hard

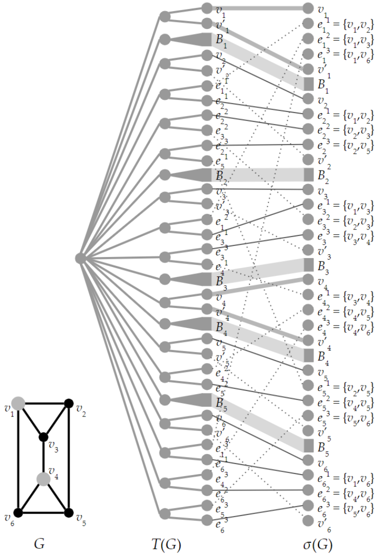
Reduction from Independent Set



Given a graph G,

OTDE is NP-hard

Reduction from Independent Set

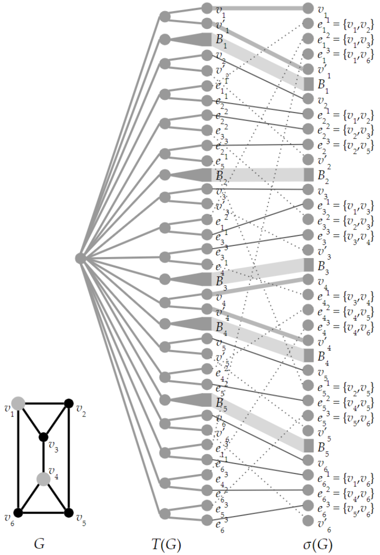


Given a graph G,
Build tree $T(G)$:

- One cherry per vertex (u, u')
- One cherry per edge (e, e')
- Separators

OTDE is NP-hard

Reduction from Independent Set



Given a graph G,
Build tree $T(G)$:

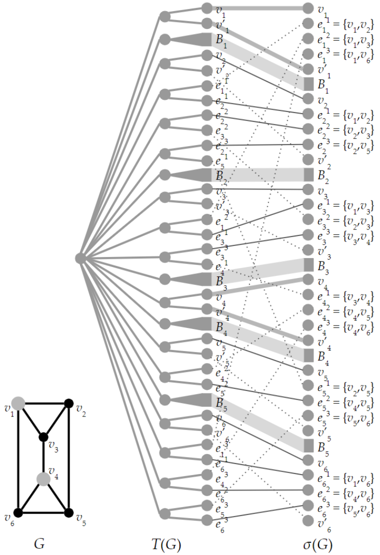
- One cherry per vertex (u, u')
- One cherry per edge (e, e')
- Separators

Build order $\sigma(G)$ (seen as a string):

- Factor $ue_1e_2e_3u'$ for each vertex and incident edges
- Separators between factors

OTDE is NP-hard

Reduction from Independent Set



Given a graph G,
Build tree $T(G)$:

- One cherry per vertex (u, u')
- One cherry per edge (e, e')
- Separators

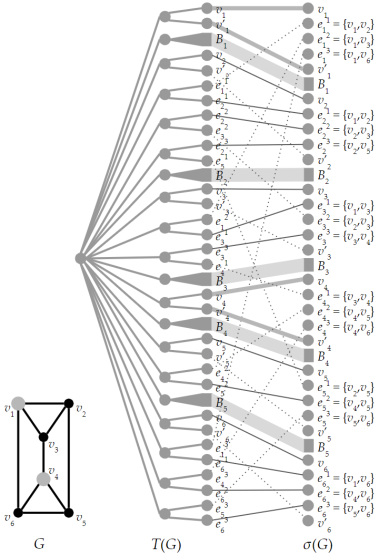
Build order $\sigma(G)$ (seen as a string):

- Factor $ue_1e_2e_3u'$ for each vertex and incident edges
- Separators between factors

Wlog, delete ≤ 1 leaf per cherry,

OTDE is NP-hard

Reduction from Independent Set



Given a graph G,
Build tree $T(G)$:

- One cherry per vertex (u, u')
- One cherry per edge (e, e')
- Separators

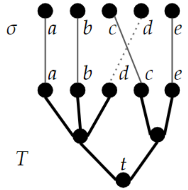
Build order $\sigma(G)$ (seen as a string):

- Factor $u e_1 e_2 e_3 u'$ for each vertex and incident edges
- Separators between factors

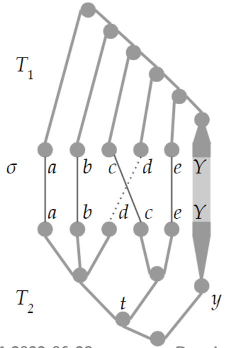
Wlog, delete ≤ 1 leaf per cherry, keep both leaves for vertices in an independent set.

TTDE is NP-hard

Reduction from OTDE

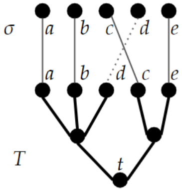


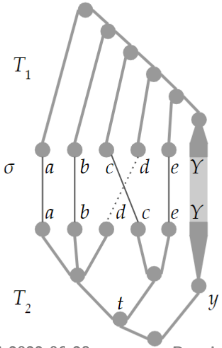
Given T, σ



TTDE is NP-hard

Reduction from OTDE



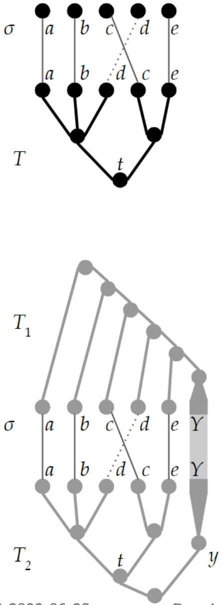


Given T, σ
Build T_1:

- Caterpillar following σ
- Large subtree ("anchor") at the bottom

TTDE is NP-hard

Reduction from OTDE



Given T, σ

Build T_1:

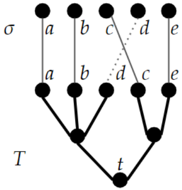
- Caterpillar following σ
- Large subtree ("anchor") at the bottom

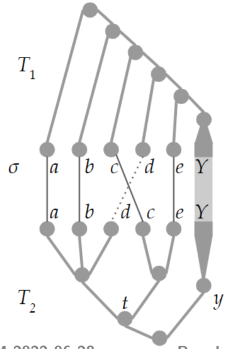
Build T_2:

- Start with T
- Connect anchor to the root

TTDE is NP-hard

Reduction from OTDE





Given T, σ

Build T_1:

- Caterpillar following σ
- Large subtree ("anchor") at the bottom

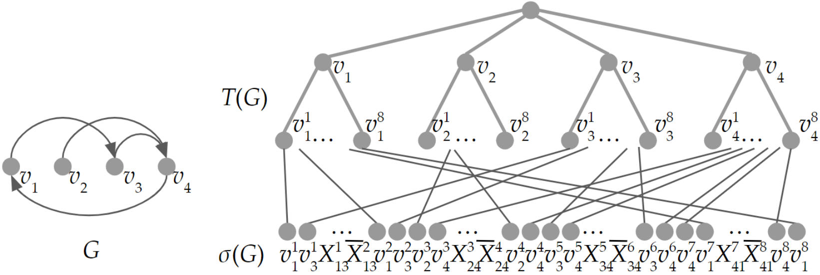
Build T_2:

- Start with T
- Connect anchor to the root

The anchor must be at one end of $T_1 \Rightarrow$ leaf order is the same as σ.

OTCM is NP-hard

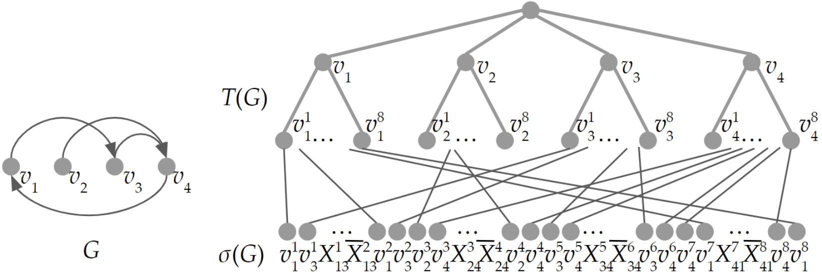
Reduction from Feedback Arc Set



Given G, build $T(G)$ with one large subtree per vertex.

OTCM is NP-hard

Reduction from Feedback Arc Set

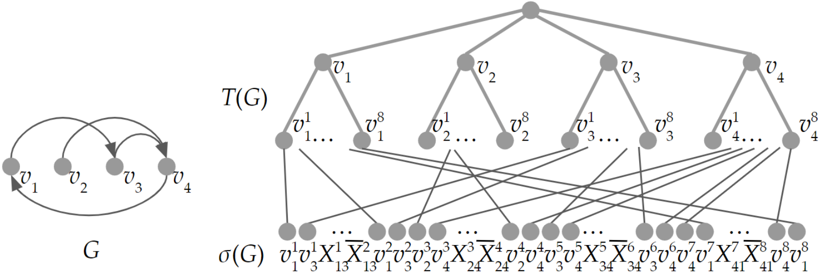


Given G, build $T(G)$ with one large subtree per vertex.
Build $\sigma(G)$ with one substring per arc :

$$v_1 \rightarrow v_3 \Longrightarrow v_1\, v_3\, v_2\, v_4\, v_4\, v_2\, v_1\, v_3$$

OTCM is NP-hard

Reduction from Feedback Arc Set



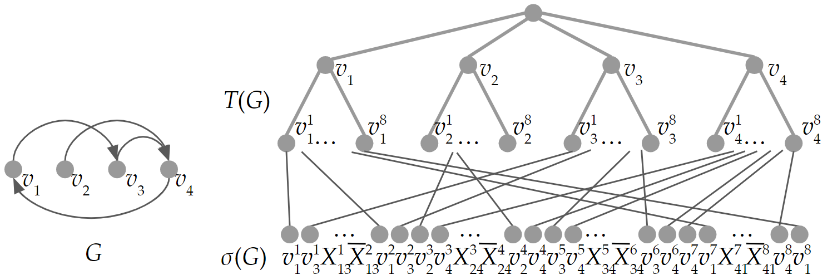
Given G, build $T(G)$ with one large subtree per vertex.
Build $\sigma(G)$ with one substring per arc :

$$v_1 \rightarrow v_3 \implies v_1\, v_3\, v_2\, v_4\, v_4\, v_2\, v_1\, v_3$$

Solution: pick a permutation of the vertices

OTCM is NP-hard

Reduction from Feedback Arc Set



Given G, build $T(G)$ with one large subtree per vertex.
Build $\sigma(G)$ with one substring per arc :

$$v_1 \to v_3 \Longrightarrow v_1 v_3 v_2 v_4 v_4 v_2 v_1 v_3$$

Solution: pick a permutation of the vertices
In the arc gadget:

- v_1, v_3 have 1 crossing if v_1 is before v_3, 3 otherwise
- Each other v_i, v_j have 2 crossings.

Experiments

Experiments: data & methods

Data

- Dated novels of 11 French 19$^{\text{th}}$ century writers
- Distance tables of novels using the relative frequencies of the 500 most frequent tokens
- Hierarchical clustering based on the distance tables, producing binary trees

Experiments: speed

tree	# leaves	OTCM time (ms)	# inversions	OTDE time (ms)	# deleted leaves
Ségur	22	1	40	200	9
Féval	23	2	47	268	8
Aimard	24	1	35	401	8
Zévaco	29	1	42	727	11
Lesueur	31	1	48	676	13
Zola	35	2	60	1203	9
Gréville	36	2	105	2211	18
Ponson	42	3	167	3447	18
Verne	58	3	183	13446	27
Balzac	59	4	248	8292	34
Sand	62	4	283	17557	39

Future work

$\Rightarrow$ Improve the dynamic programming algorithm/implementation solving OTDE

Experiments: presence of chronological signal

tree	# leaves	# inversions	p_{OTCM}	# deleted leaves	p_{OTDE}
Ségur	22	40	0.24	9	1
Féval	23	47	0.38	8	0
Aimard	24	35	0	8	0
Zévaco	29	42	0	11	0
Lesueur	31	48	0	13	0
Zola	35	60	0	9	0
Gréville	36	105	0	18	1
Ponson	42	167	2.23	18	0
Verne	58	183	0	27	0
Balzac	59	248	0	34	0
Sand	62	283	0	39	1

p_{OTCM}, p_{OTDE} = probablility (%) that a random order gives a better score than the chronological order

(over 10000 tries for OTCM, 100 for OTDE).

Experiments: identification of noise

Simulation experiment by adding errors in the leaf order

Repeat 100 times:

1. randomly choose "dates" from the interval [0,999]
2. build a distance matrix of the absolute differences between "dates" and the corresponding dendrogram
3. insert e artificial errors: pick a new random "date" for e randomly chosen leaves.

- Does OTDE output the set L_e of leaves with artificial errors?

Experiments: identification of noise

$n =$ # leaves	$e =$ # errors	proportion of cases when $L = L_e$	when $\lvert L - L_e \rvert = 1$
20	1	0.79	1
20	2	0.62	0.96
20	3	0.39	0.88
20	4	0.33	0.77
20	5	0.27	0.67
50	1	0.93	1
50	2	0.83	0.99
50	3	0.70	0.98
50	4	0.59	0.91
50	5	0.56	0.90

Observations

- if at most 2 errors, identified in more than 60% of the experiments, at least 1 identified in more than 96%.

Conclusion

Main results

- NP-hardness proofs for problems useful in bioinformatics and digital humanities
- FPT-algorithm in the deletion degree
- implementation in Python of an algorithm solving OTCM and OTDE, to evaluate the chronological signal in a tree
- a direct method to study the presence of the chronological signal in the data

Conclusion

Future works

- optimize the dynamic programming algorithm for OTDE
- evaluate the expected number of inversions or deleted leaves for a random order
- extend experiments:
  - run OTCM / OTDE on other datasets from different fields (see `https://github.com/oseminck/tree_order_evaluation`)
  - in-depth studies of cases where some leaves are expected to be wrongly ordered for OTDE
  - discuss the obtained results about the evolution of author styles with specialists of the authors