Fast tree aggregation for consensus hierarchical clustering

A. Hulot[1,2,3], J. Chiquet[2], F. Jaffrézic[1], G. Rigaill[4,5]

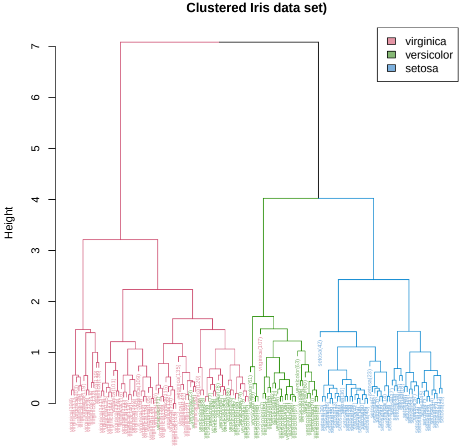
[1] GABI, INRA, Université Paris-Saclay, 78350 Jouy-en-Josas, France
[2] MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay, 75005 Paris, France
[3] INSERM UMR 1173, Université de Versailles-Saint-Quentin-en-Yvelines, 78180 France
[4] LaMME, UEVE, CNRS/ENSIIE/USC INRA, 91000 Evry, France
[5] Institute of Plant Sciences Paris-Saclay, UMR 9213/ULR 1403 CNRS INRA,
Université Paris-SUD, Université d'Evry, Université Paris-Diderot, Sorbonne Paris-Cité,

A tree and its interpretation

Definition (Graph Theory)

Undirected graph in which any two vertices are connected by exactly one path, or equivalently a connected acyclic undirected graph.

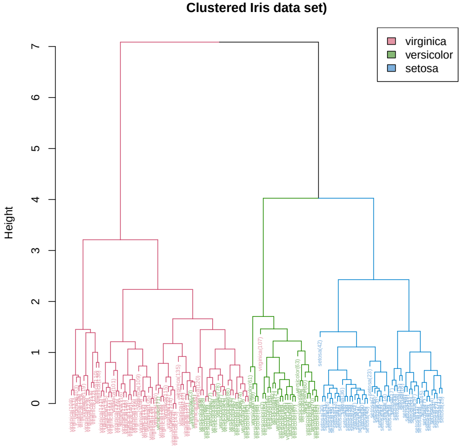
- Exploratory method, unsupervised
- Graphical representation of the dissimilarities between clusters/individuals (height of fusion)
- Efficiently visualize group structure in the data for various number of groups



How to build a tree?

Agglomerative hierarchical clustering

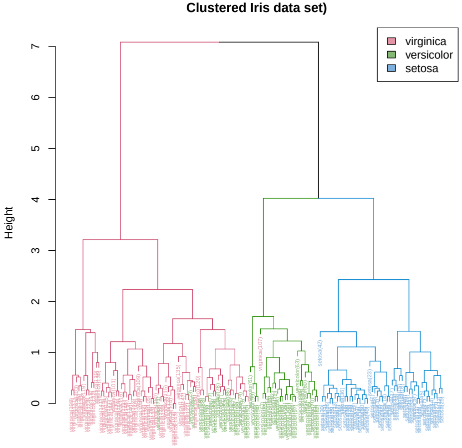
1. Compute distances/dissimilarities between individuals (bottom of the tree)
2. Aggregate the closest individuals or clusters *agglomerative criterion* and update the distance matrix
3. Repeat the (2) until all individuals are in one group



Pros / cons

+ Require no prior information
+ Require no/very little treatment of the data

– $\mathcal{O}(n^2)$+ (not a huge number of leaves)
– **Not directly adapted to the treatment of multiple datasets / heterogeneous data**



Why a consensus of trees?

- Multiple table providing multiple trees (multi-omics)
- Bootstrap (Phylogenetics)
- Hope for a more stable information
- Hope for less diluted group information (shared among the trees)

Field of interest: multi-omics analysis.

Multi-Omics

- Recent development in the last decade about clustering
- They do not return a tree
- Phylogenetics methods not applicable here

Context: single / multi-omics data analysis

Why?

+ Better understanding of biological processes
+ Better understanding of entities relationships
  $\hookrightarrow$ Better diagnosis / Earlier diagnosis
  $\hookrightarrow$ Better treatments

Difficulties

- Heterogeneous data (continuous, counts, percentage...)
- High-dimensional data ($n \ll p$)
- Noisy

Methods

Direct Clustering

1. Merge all datasets into one
2. *Scale the data*
3. Compute distance and apply aggregation criterion

+ Very easy to compute and highly interpretable
– Giant matrix $\rightarrow$ memory issues

Average Distance

Merge Trees

Methods

Direct Clustering

+ Very easy to compute and highly interpretable
– Giant matrix $\rightarrow$ memory issues

Average Distance

1. Distance on each dataset
2. Average all of the matrices
3. Apply aggregation criterion on this new matrix

+ Easy / highly interpretable
– Not very robust to noise

Merge Trees

Methods

Direct Clustering

- \+ Very easy to compute and highly interpretable
- – Giant matrix $\rightarrow$ memory issues

Average Distance

- \+ Easy / highly interpretable
- – Not very robust to noise

Merge Trees

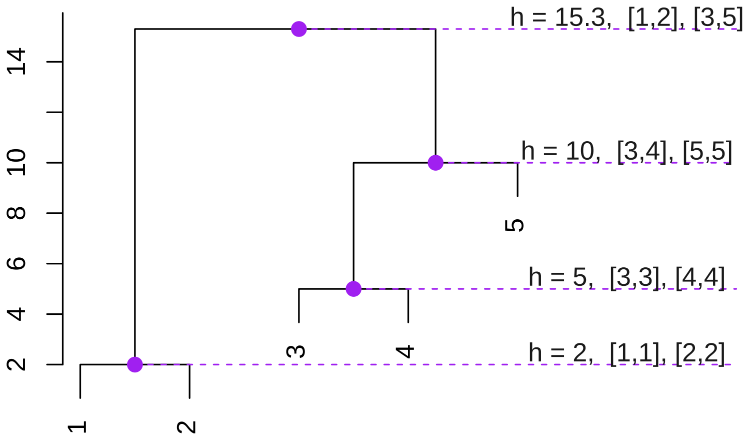
1. Distance on each dataset
2. Build hierarchical clustering
3. Merge the trees

Tree definition

Definition

Let T be a tree.
T is a succession of $(n-1)$ *splits*.
Characterized by:

- height of the division
- the 2 clusters created by the division



Merging method

Definition

Let $\mathcal{T} = \{T_1, \ldots, T_d\}$ be a set of d trees obtained by a hierarchical clustering method.

$\longrightarrow$ list of $(n-1) \times d$ possible splits

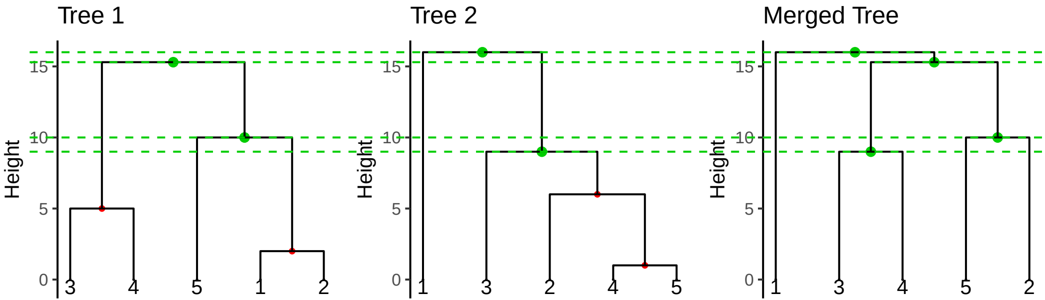
Merging the trees: (divisive clustering method)

- Order all of the possible splits by decreasing height
- For each split: check if it is active in the current situation i.e. if at least one element is impacted by the division
- If it is active, apply it, else, go to the next split
- Stop when every variable is in its own group

An example

Definition

Active split: split that impacts the current situation of the tree.
We call **consensus tree** the tree formed by the active splits

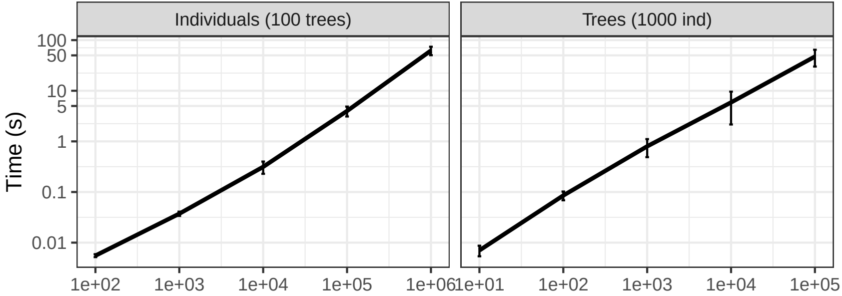


Result tree is not always a binary tree!

Timing Complexity

Theorem

The consensus tree can be obtained in $\mathcal{O}(dn\log(n))$



- Able to aggregate a large number of trees
- Able to aggregate trees with lot of individuals

Breast cancer data

Omics data

- 4 datasets
- Heterogeneous data
- Different dimensions and scales

Data		Features
methylation	percentage	21 123
miRNA	continuous	725
proteins	continuous	156
genes	counts (log2)	19 738

Individuals

- 104 patients
- 4 Subtypes
- ER/PR status (+/−)

Subtype	Individuals
Luminal A	44
Luminal B	20
HER2-enriched	18
Basal-like	22

Data downloaded from TCGA website

Treatment of data/trees (1)

Data treatment

- All datasets: centered, not scaled
- **Divided by the first singular value**

Clustering building

- Distance: Euclidean
- Aggregation criterion: Ward

Murtagh F. & Legendre P. (2014) Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? Journal of Classification, 31,274-295

Performance evaluation

NID *Normalized Information Distance*: distance between classifications

Performance evaluation

NID *(Normalized Information Distance)*

$$1 - \frac{I(U,V)}{\max\left(H(U), H(V)\right)}$$

$\rightarrow$ Distance between classifications, $\in [0,1]$

U/V	V_1	V_2	$\dots$	V_C	Sums
U_1	n_{11}	n_{12}	$\dots$	n_{1C}	$n_{\bullet 1}$
U_2	n_{21}	n_{22}	$\dots$	n_{2C}	$n_{\bullet 2}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_R	n_{R1}	n_{R2}	$\dots$	n_{RC}	$n_{\bullet R}$
Sums	$n_{1\bullet}$	$n_{2\bullet}$	$\dots$	$n_{C\bullet}$	$\sum_{ij} n_{ij} = N$

Entropy:

$$H(U) = -\sum_{i=1}^{R} \frac{n_{i\bullet}}{N} \log \frac{n_{i\bullet}}{N}$$

Mutual Information

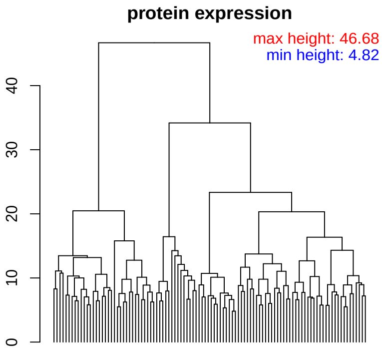
$$I(U,V) = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{n_{i\bullet} n_{\bullet j}/N^2}$$

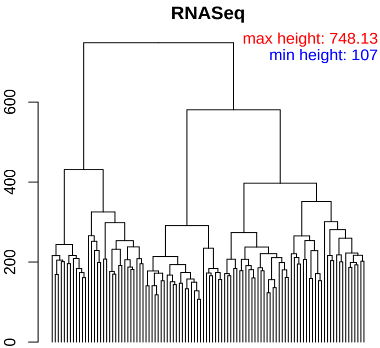
N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research , 11(Oct) :2837-2854, 2010.

Treatment of data/trees (2)

- Heterogeneous data → different range of values
- Different datasets → different number of variables

⇒ **Different range of distances and height splits in the trees**





↪ All of RNASeq's tree splits happen before any division of protein tree, consensus tree IS RNASeq tree

Treatment of data/trees (3)

- Heterogeneous data $\rightarrow$ different range of values
- Different datasets $\rightarrow$ different number of variables

$\Rightarrow$ **Different range of distances and height splits in the trees**

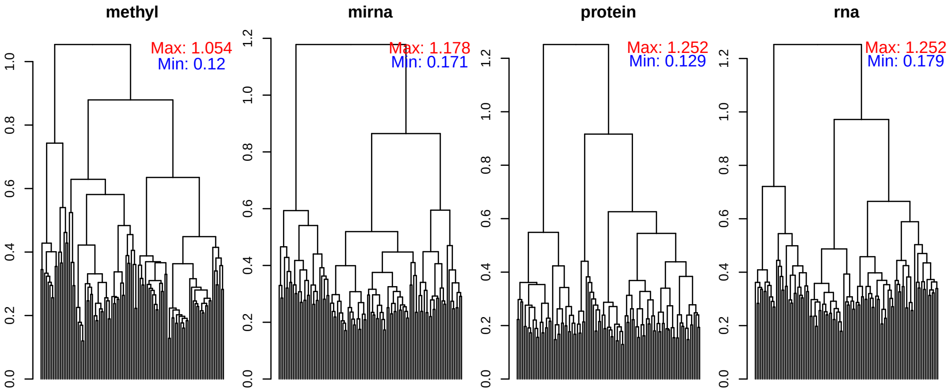
Some ideas

- Scale all the datasets
- Divide each distance matrix by its maximum
- Divide each tree by its maximum height (non binary tree result)
- Not taking the height into account but the number of fusions
- **Divide each dataset by its first singular value (root square of first eigenvalue)**

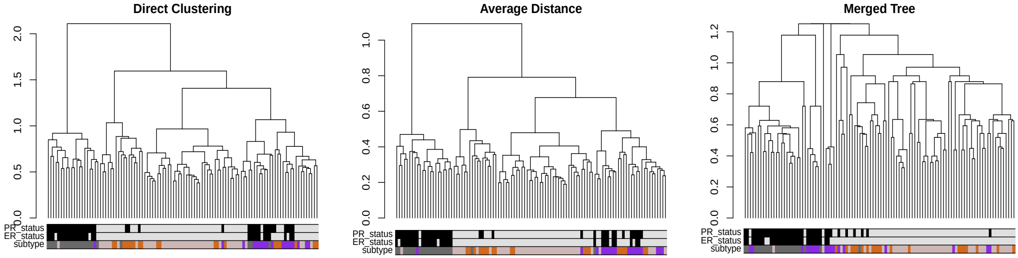
Treatment of data/trees (4)

- Heterogeneous data → different range of values
- Different datasets → different number of variables

⇒ **Dividing datasets by their first singular value**



Breast cancer data: results



	ER status		PR status		Subtype	
	N	NID	N	NID	N	NID
methyl	3	0.77	4	0.78	9	0.69
mirna	2	0.72	2	0.71	4	0.67
protein	2	**0.32**	2	**0.45**	5	0.53
rna	2	0.40	2	0.55	4	0.59
Average Distance	2	0.61	2	0.66	**4**	**0.54**
Direct Clustering	2	0.63	2	0.74	4	0.60
Merge Trees	**2**	**0.40**	**3**	**0.51**	8	0.56

Conclusion and Perspective

Summary:

- Fast algorithm: $\mathcal{O}(nd\log(n))$
- Consistant results on applications
- R package `mergeTrees` available on the CRAN devs: A. Hulot, J. Chiquet, G. Rigaill

Perspective

- Weighting applied on data/trees
- Spectral application
- Judging quality of a hierarchical clustering

Thank you for your attention!

Timing theorem and sketch of the proof

Theorem

The consensus tree can be obtained in $\mathcal{O}(dn\log(n))$

Proof: Based on a recurrence relation for $T(n)$, the worst time scenario to build an n-elements-tree with our method.

- Main idea to speed up the algorithm: at each split-activating step, consider only the smallest number of elements to split, $n/2$ variables at most
- Leads to the recurrence relation:

$$T(n) = \max_{i=1}^{n/2}\{i + T(i) + T(n-i)\}$$

- Result of function bounderies: $T(n) \leqslant \frac{n}{2}\log_2(n)$
- Having d trees to consider:

The merging algorithm is of complexity $\mathcal{O}(dn\log(n))$