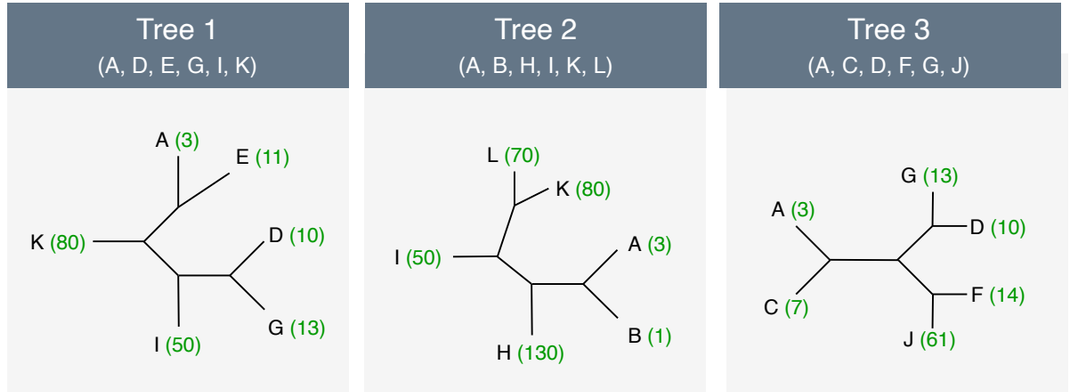


Step 1
Compute the score of all edges

Score per edge		
Tree 1	DG - AEKI	0.59
Tree 1	DGI - AEK	0.36
Tree 1	DGIK - AE	1.78
Tree 2	AB - HIKL	4.39
Tree 2	ABH - IKL	2.68
Tree 2	ABHI - KL	2.13
Tree 3	AC - DGFJ	1.60
Tree 3	ACFJ - DG	0.017
Tree 3	ACDG - FJ	1.61

Step 2
Partition the edges

I (50), K (80), L (70), H (130) | A (3), B (1)

C (7), D (10), E (11), F (14), G (13), J (61) ?

Step 3
Recruit remaining edges

Example: Recruitment of D

Score per edge		
Tree 1	DG - AEKI	0.59
Tree 1	DGI - AEK	0.36
Tree 1	DGIK - AE	1.78
Tree 2	AB - HIKL	4.39
Tree 2	ABH - IKL	2.68
Tree 2	ABHI - KL	2.13
Tree 3	AC - DGFJ	1.60
Tree 3	ACFJ - DG	0.017
Tree 3	ACDG - FJ	1.61

$D + G \rightarrow /$
$D - A \rightarrow$ Left
$D - E \rightarrow /$
$D - K \rightarrow$ Right
$D - I \rightarrow$ Right

Recruitment of D	
Left side	**Right side**
1×0.59	2×0.59

$D + G \rightarrow /$
$D + I \rightarrow$ Left
$D - A \rightarrow$ Left
$D - E \rightarrow /$
$D - K \rightarrow$ Right

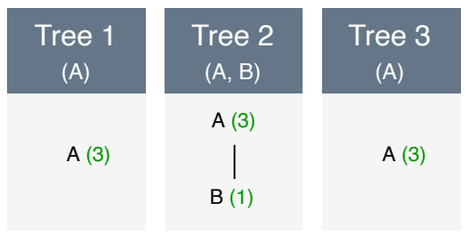
Recruitment of D	
Left side	**Right side**
2×0.36	1×0.36

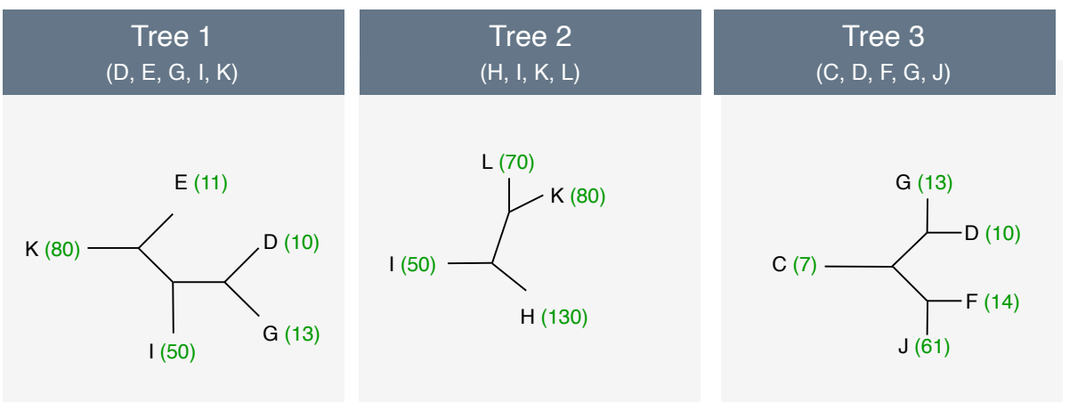
Recruitment of D	
Left side	**Right side**
2.21	16.5
12 %	88 %

Step 4
Recursion

C (7), D (10), E (11), F (14), G (13), J (61), K (80), L (70), H (130) | A (3), B (1)

- Apply the same partitioning procedure to each subset
- Trim each tree to only include the subset



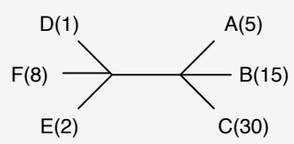


⇒ random draw or argmax
⇒ D goes right
⇒ Next

Step 2
Recruitment of missing edges

Score per edge (adjusted for correlation)		
Tree 1	AB - CDE	0.435
Tree 2	AE - CFG	1.9
Tree 2	CG - AEF	2.0

Recruitment of F	
Edge	**Partition**
AF	{A,B,C}
EF	{D,E}
CF	{D,E}
GF	/
Result	**{D,E}**



Recruitment of G