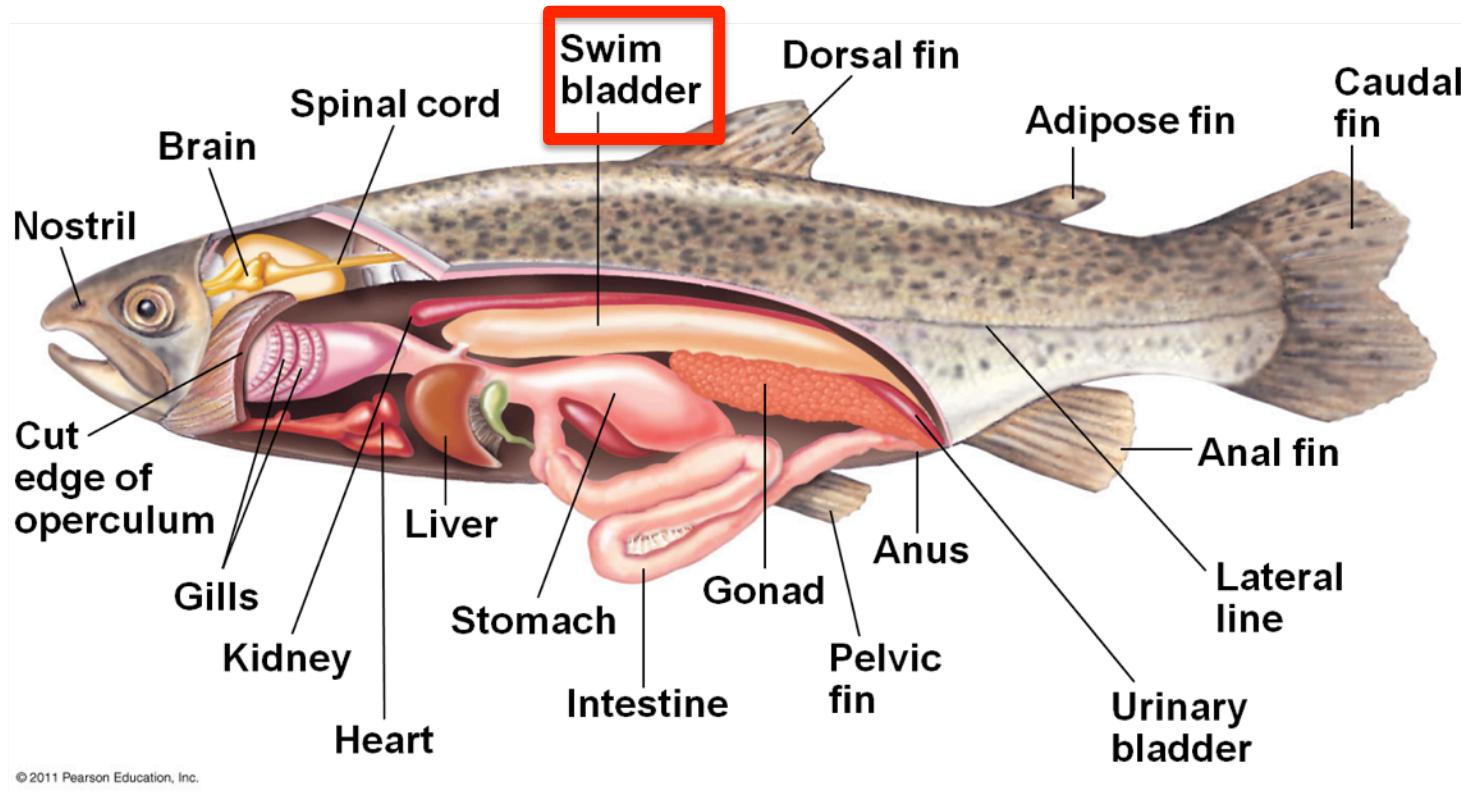


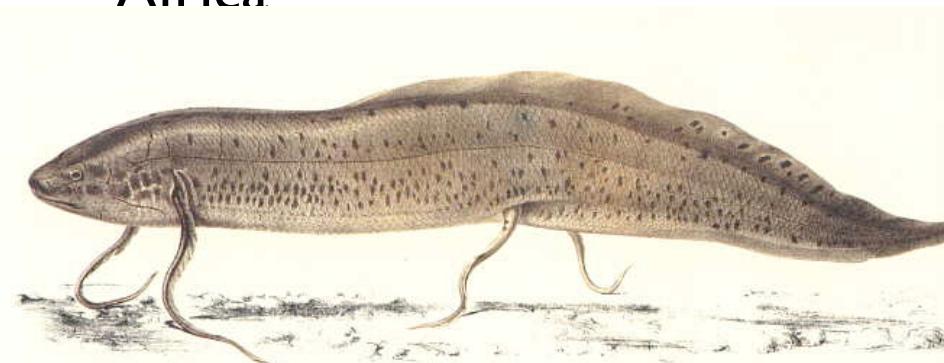
# I. Ancestral states and optimising data on trees

# Which evolved first: the swim bladder or the lung?

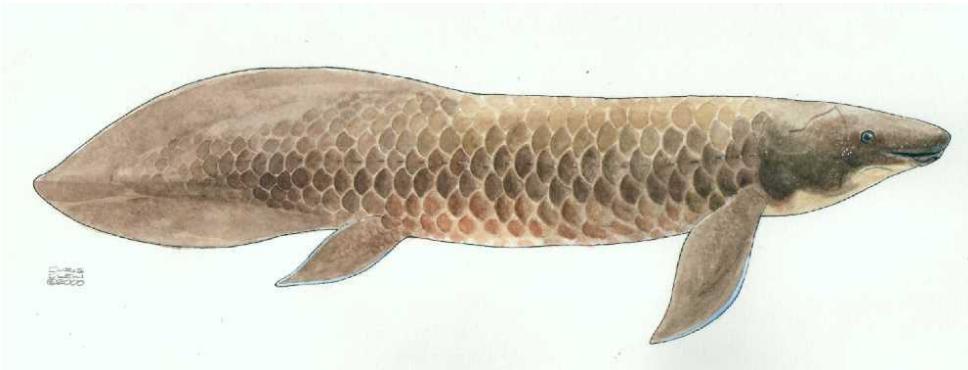


# Lungfishes

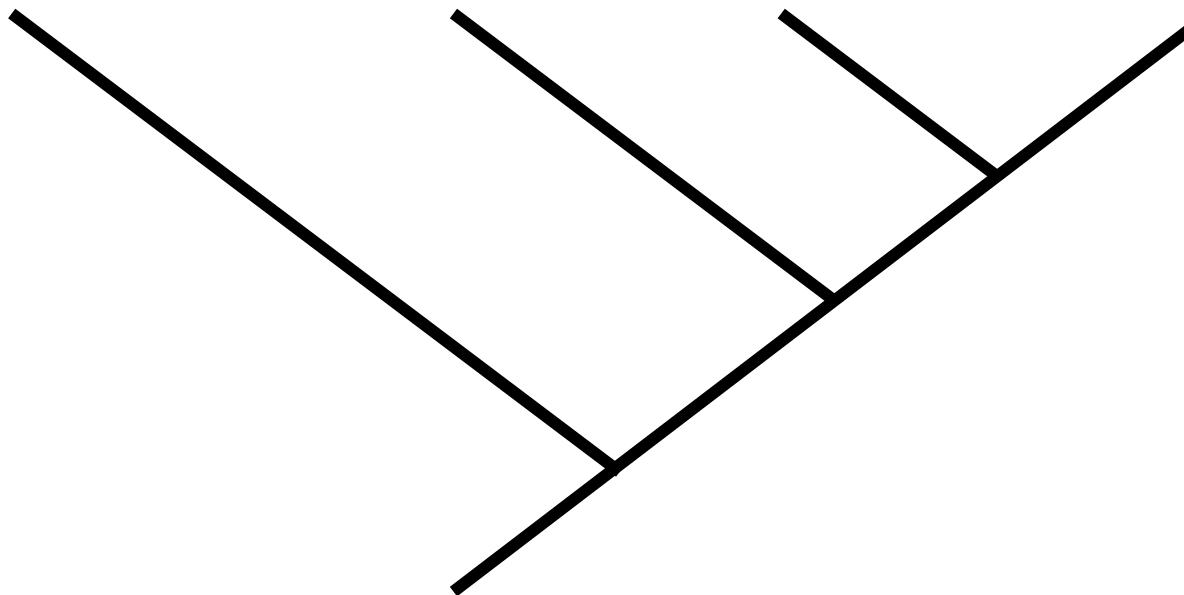
*Protopterus* – 4 spp.  
Africa



*Lepidosiren paradoxa* – from South America



*Neoceratodus forsteri*  
Australia  
Proper lobate fins

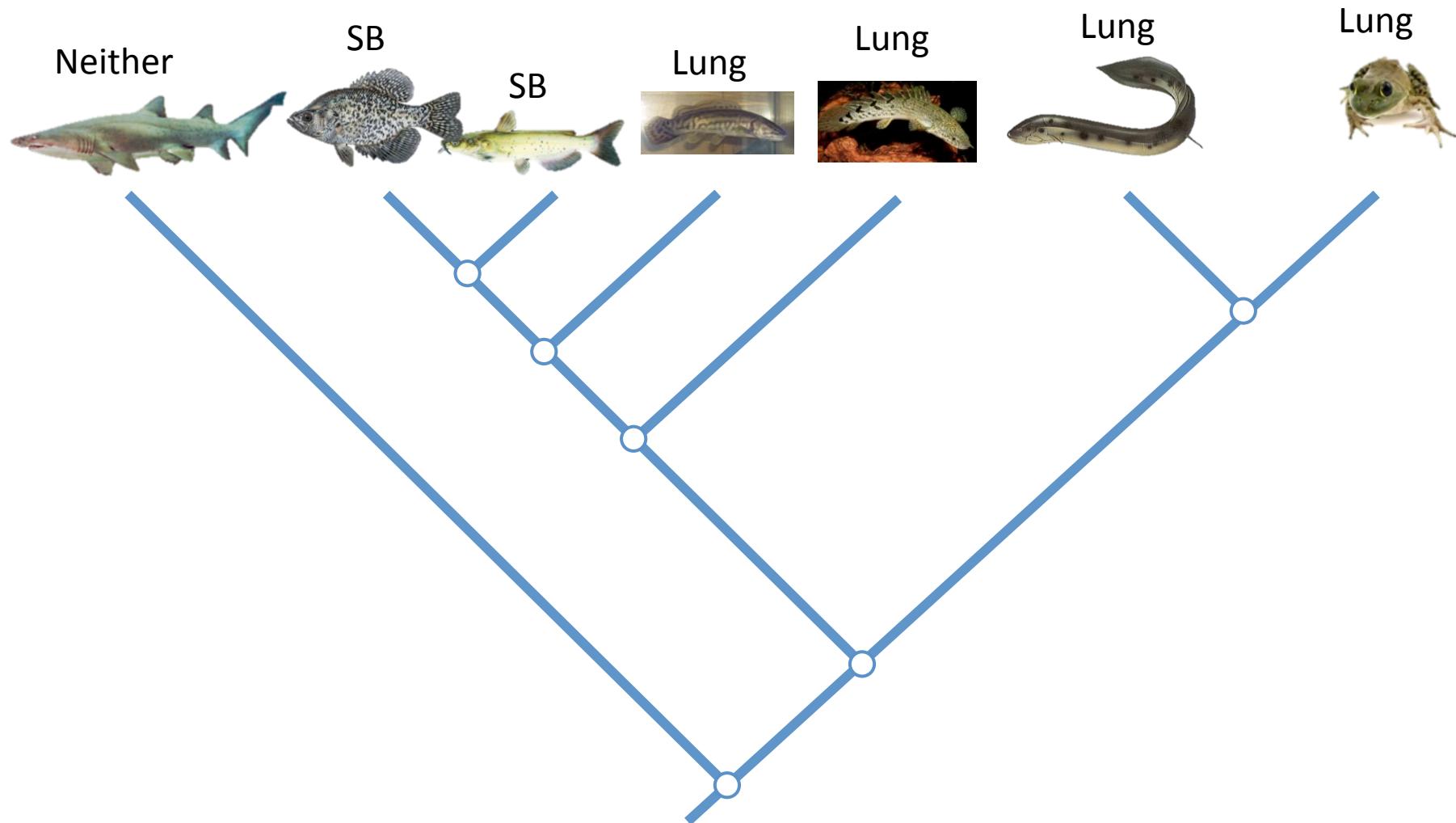


# The air-breathing fishes

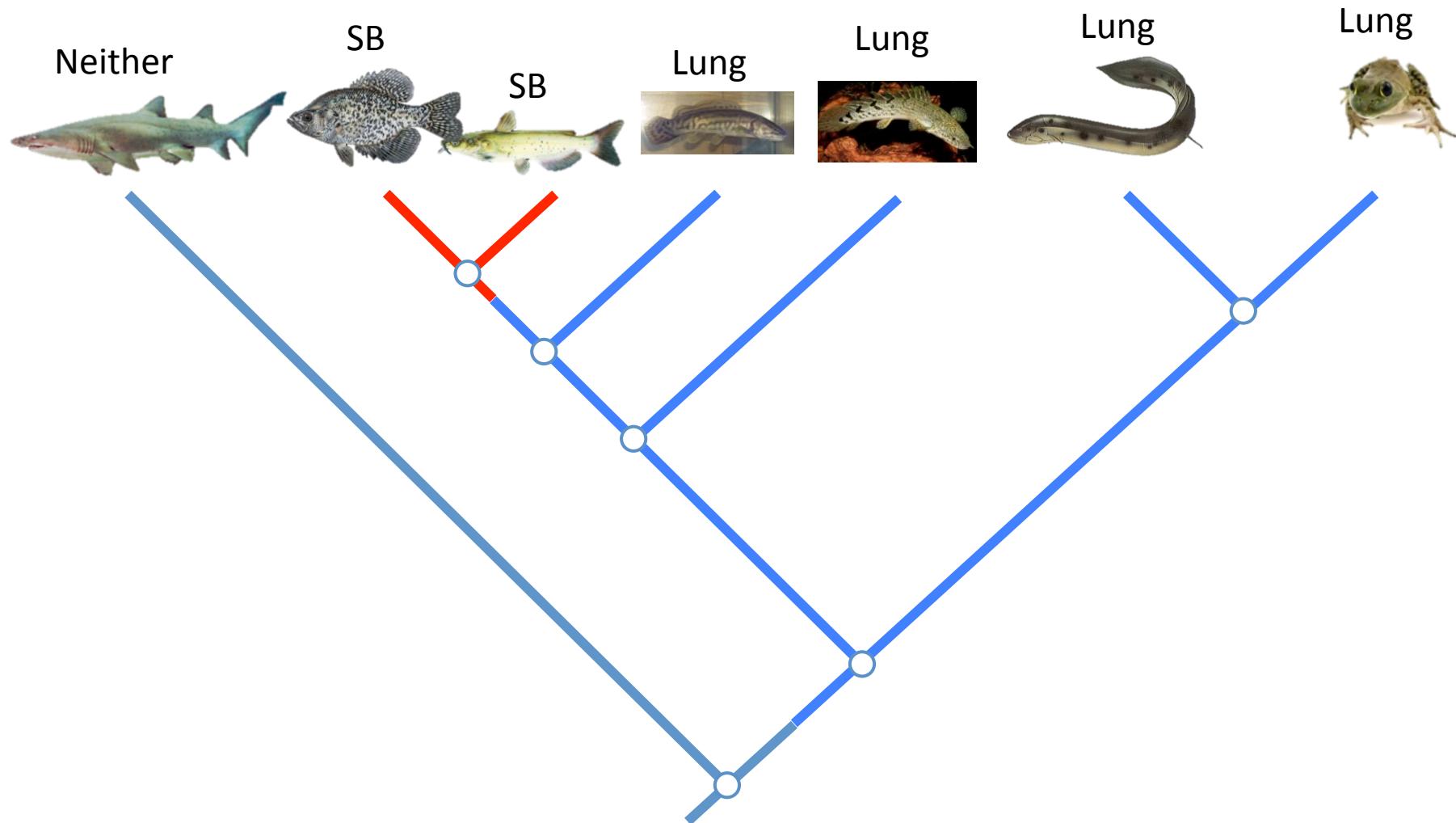
- Air-breathing is very common among fishes



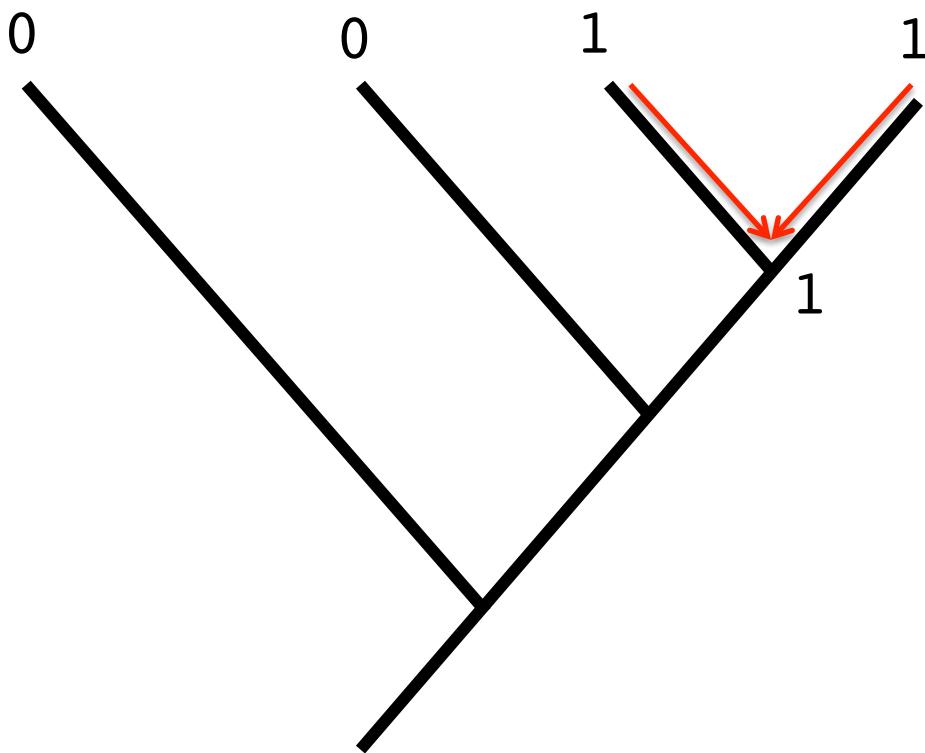
# Which evolved first: the swim bladder or the lung?



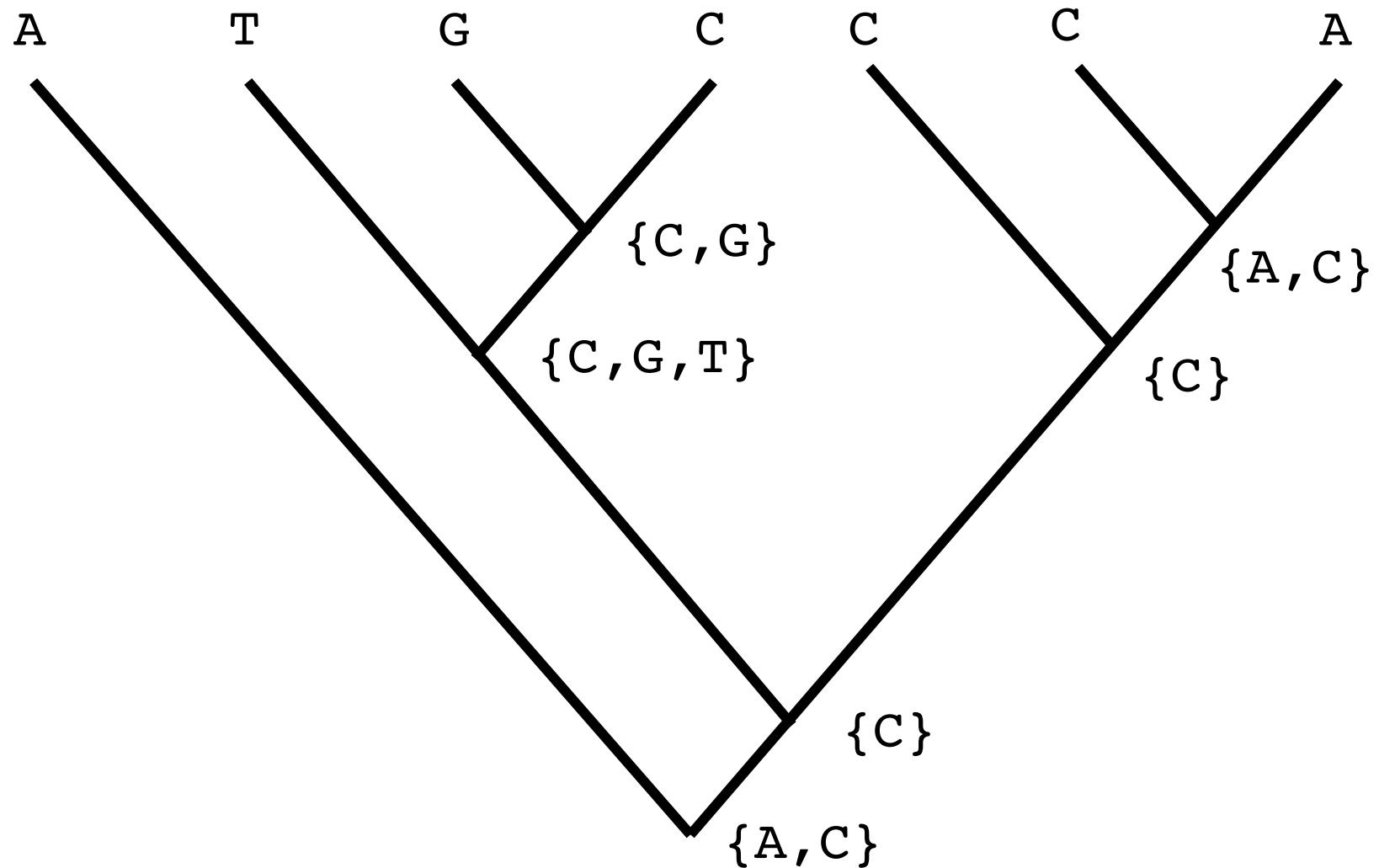
# Which evolved first: the swim bladder or the lung?



# Part I. Parsimony (Semi-formally)



# Fitch downpass (postorder)



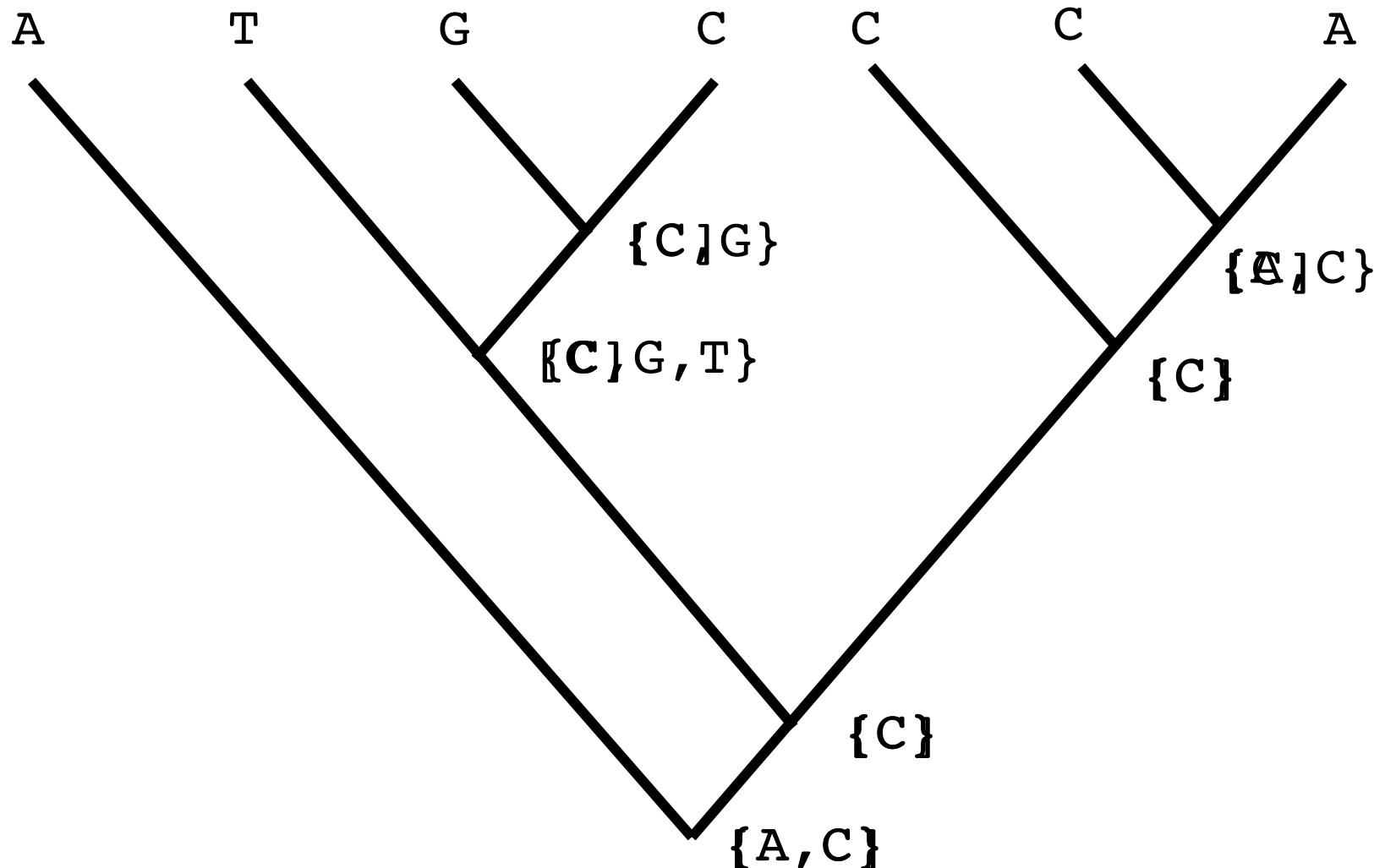
# Fitch downpass (postorder)

- Construct preliminary ancestral state sequences
- If descendant branches have common states:
  - Reconstruction is the *intersection* of those states
- Otherwise:
  - Reconstruction is the *union* of those states

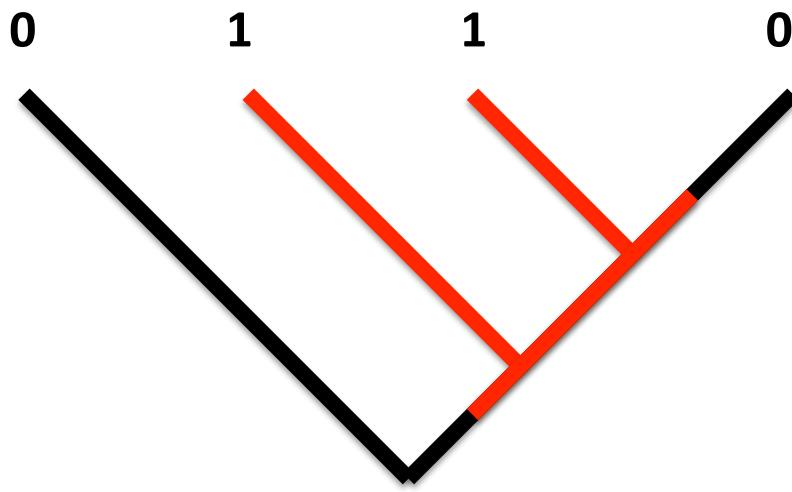
# Fitch downpass (postorder)

- Only provides a preliminary set of ancestral state reconstructions
- Doesn't account for information at deeper nodes (because the downpass hasn't reconstructed them yet)
- Final reconstructions requires bringing information up from the root.

# Final estimate: Fitch uppass (preorder)

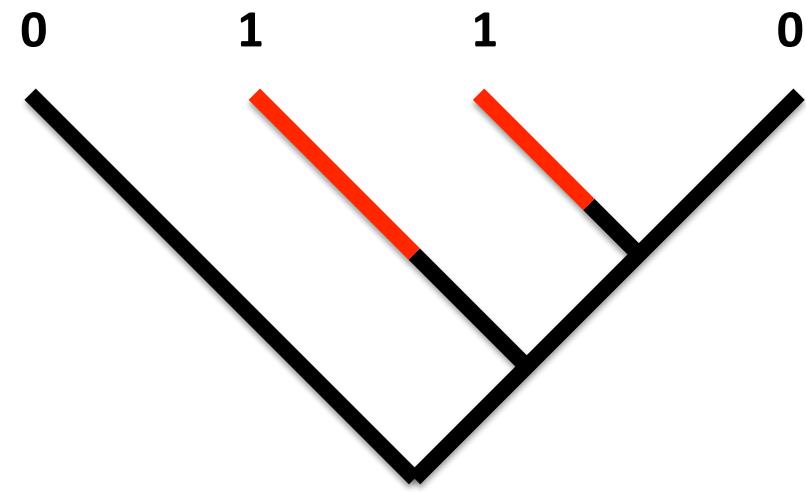


# Ambiguity



2 steps  
(gain, loss)

“Accelerated  
transformation”



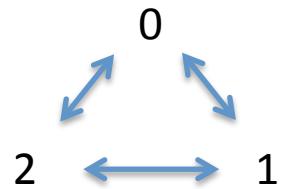
2 steps  
(gain, gain)

“Delayed  
transformation”

# Ordered (Wagner) parsimony

- Sometimes characters are ordered along a cline

Fitch



Wagner

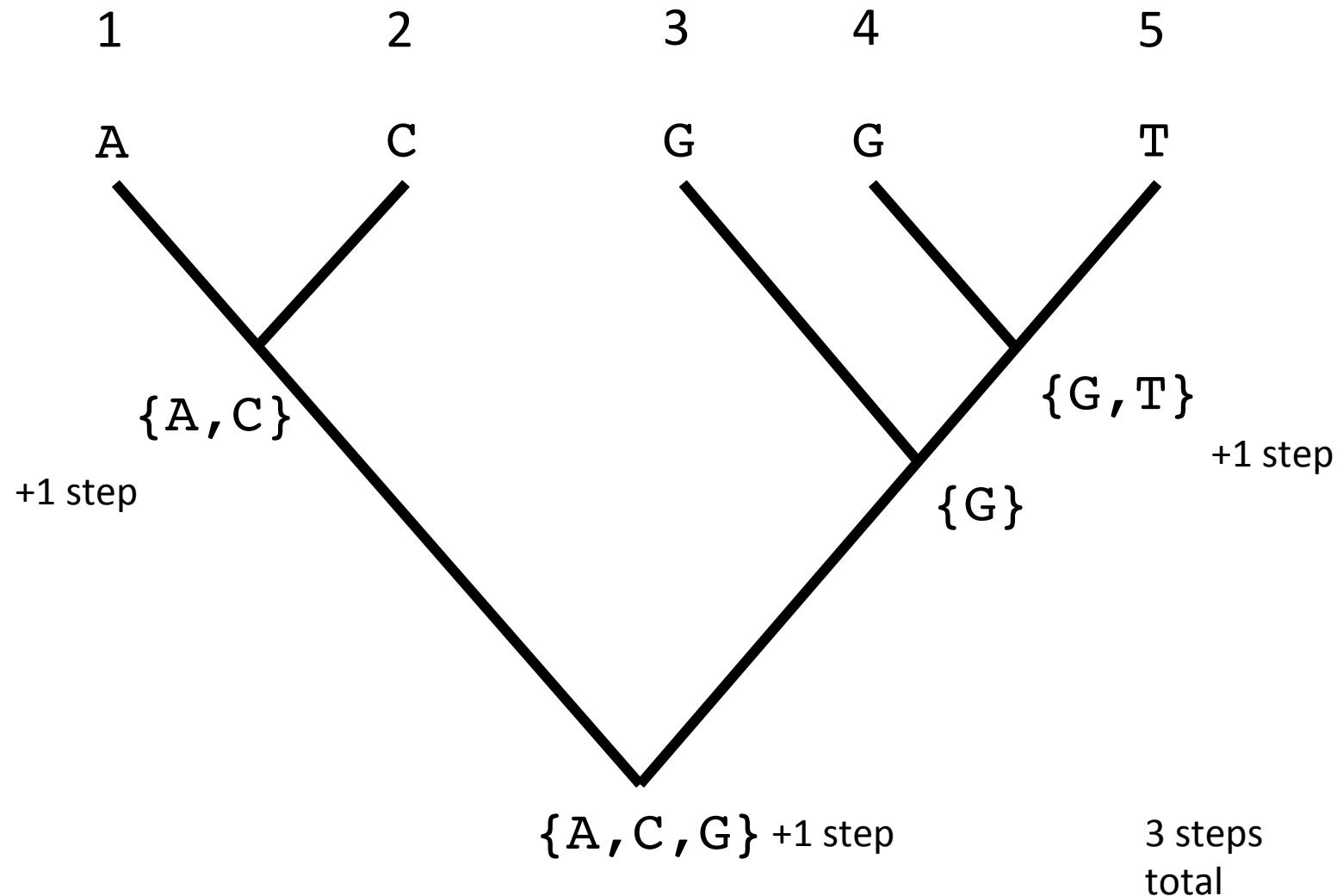


All transitions are symmetrical, evenly weighted

Transformations are ordered (additive).

# Part II. Parsimony (formally)

# Fitch (unordered) parsimony



# The Sankoff method

- More explicit
- Matrix of transformation weights ('costs')
- The **Q** matrix

	A	C	G	T
A	-	1	1	1
C	1	-	1	1
G	1	1	-	1
T	1	1	1	-

# Minimising changes under Sankoff parsimony

- The minimum length  $S$  of a character on a tree is

$$S = \min_i S_0(i)$$

- Which we compute for each node  $a$  using

$$S_a = \min_j [c_{ij} + S_l(j)] + \min_k [c_{ik} + S_r(k)]$$

- Where:
- $l$  and  $r$  denote left and right descendants, respectively.
- $S_0$  is the root node.
- $c_{ij}$  and  $c_{ik}$  are the cost of transitions from states  $i$  to  $j$  and  $i$  to  $k$ , respectively.

1

A

2

C

3

G

4

G

5

T

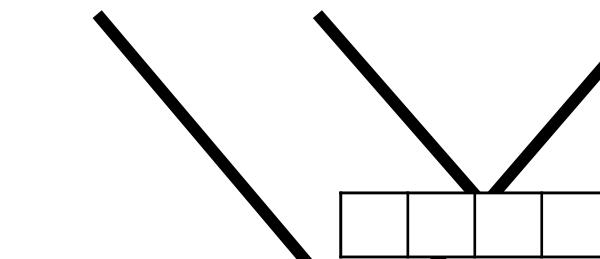
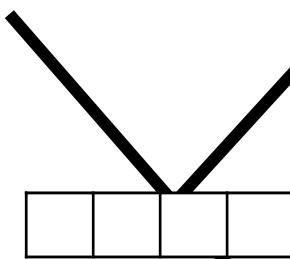
0	$\infty$	$\infty$	$\infty$
---	----------	----------	----------

$\infty$	0	$\infty$	$\infty$
----------	---	----------	----------

$\infty$	$\infty$	0	$\infty$
----------	----------	---	----------

$\infty$	$\infty$	0	$\infty$
----------	----------	---	----------

$\infty$	$\infty$	$\infty$	0
----------	----------	----------	---



A	C	G	T
---	---	---	---

	A	C	G	T
A	-	1	1	1
C	1	-	1	1
G	1	1	-	1
T	1	1	1	-

# The Sankoff (matrix) method

- We can, in effect, score a tree by considering *all* possible reconstructions at every single node.
- Computationally intensive, but has some advantages.
- For instance, varying character weights

# Ordering expressed as a matrix

Unordered (Fitch)

	0	1	2	3
0	-	1	1	1
1	1	-	1	1
2	1	1	-	1
3	1	1	1	-

Ordered (Wagner)

	0	1	2	3
0	-	1	2	3
1	1	1	-	1
2	2	2	1	-
3	3	3	2	1

1

A

2

C

3

G

4

G

5

T

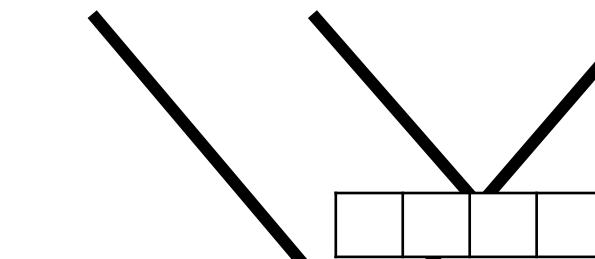
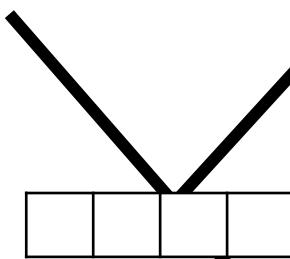
0	$\infty$	$\infty$	$\infty$
---	----------	----------	----------

$\infty$	0	$\infty$	$\infty$
----------	---	----------	----------

$\infty$	$\infty$	0	$\infty$
----------	----------	---	----------

$\infty$	$\infty$	0	$\infty$
----------	----------	---	----------

$\infty$	$\infty$	$\infty$	0
----------	----------	----------	---



A	C	G	T
---	---	---	---



	A	C	G	T
A	-	2.5	1	2.5
C	2.5	-	2.5	1
G	1	2.5	-	2.5
T	2.5	1	2.5	-

# Fitch vs. Sankoff

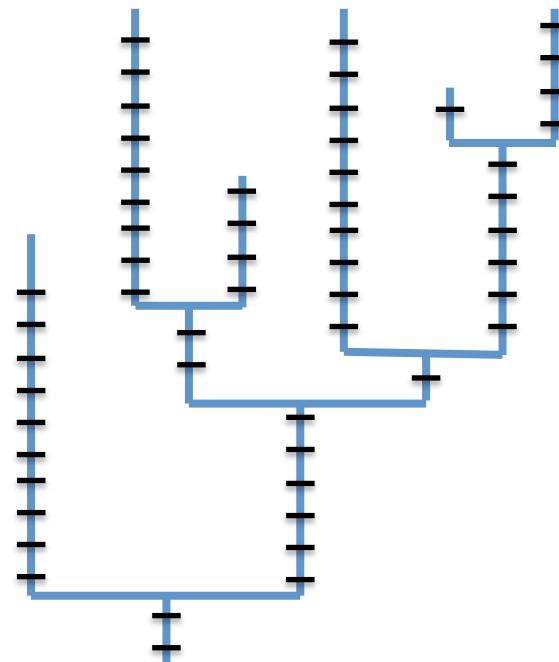
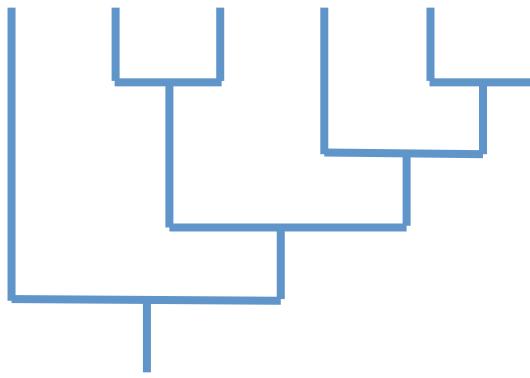
- Fitch is intuitive
- Fitch is computationally easier
- Sankoff is more explicit
- Allows for varying of weights

# Part III. Models of Character Change

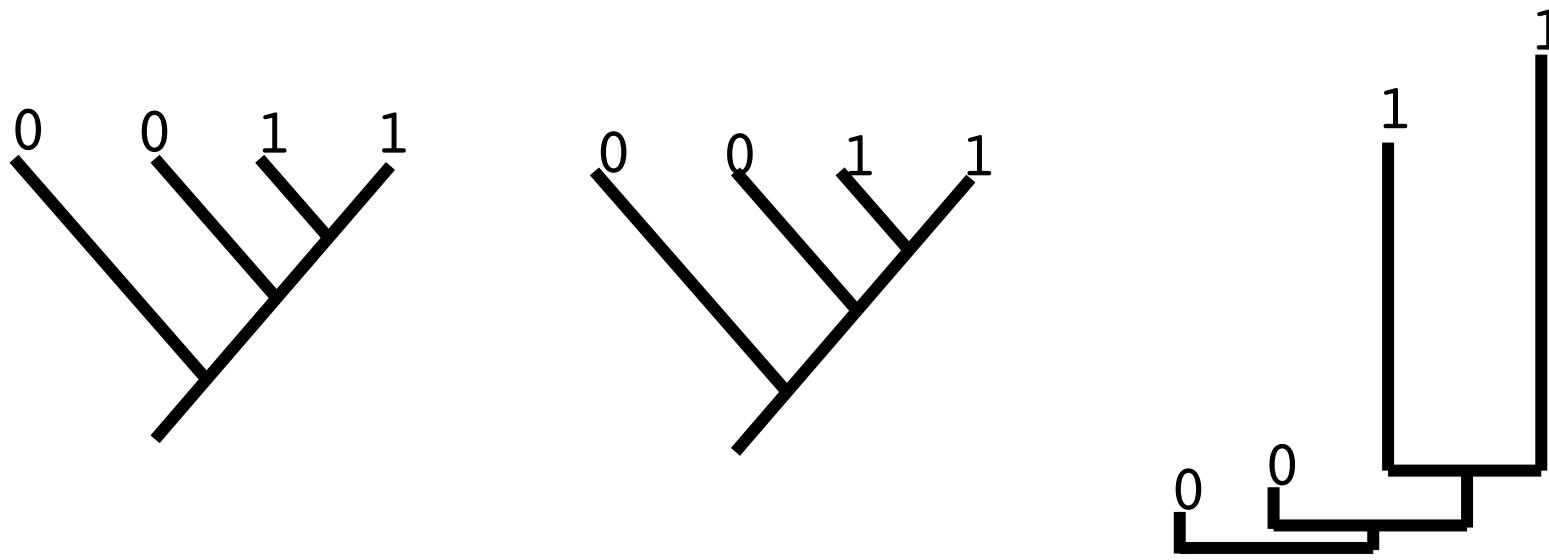
An intuitive look at going beyond parsimony

# Branch lengths

- The (mean) number of substitutions/transformations along a branch
- i.e. the number of synapomorphies supporting a node



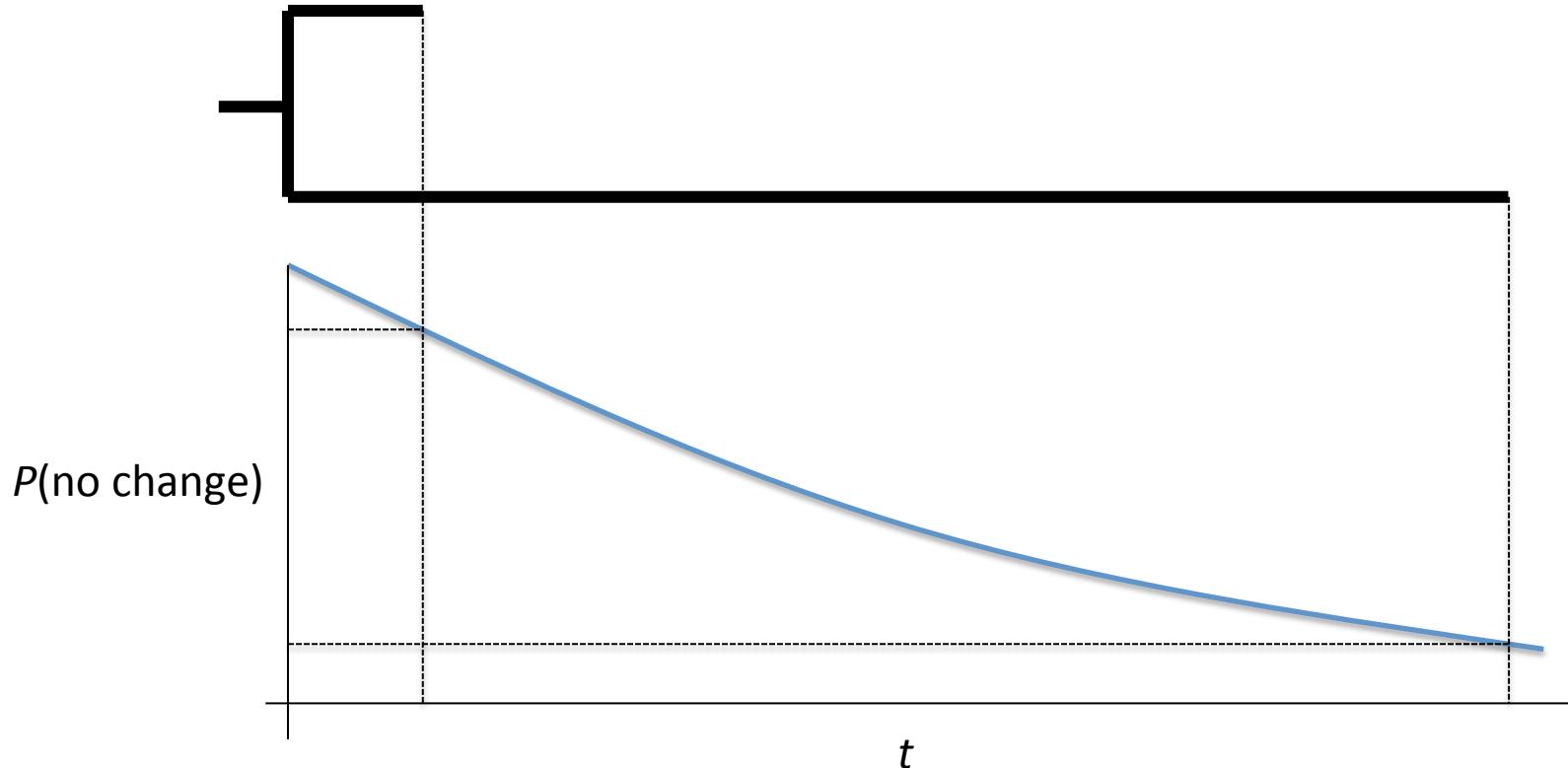
# Parsimony does not model time



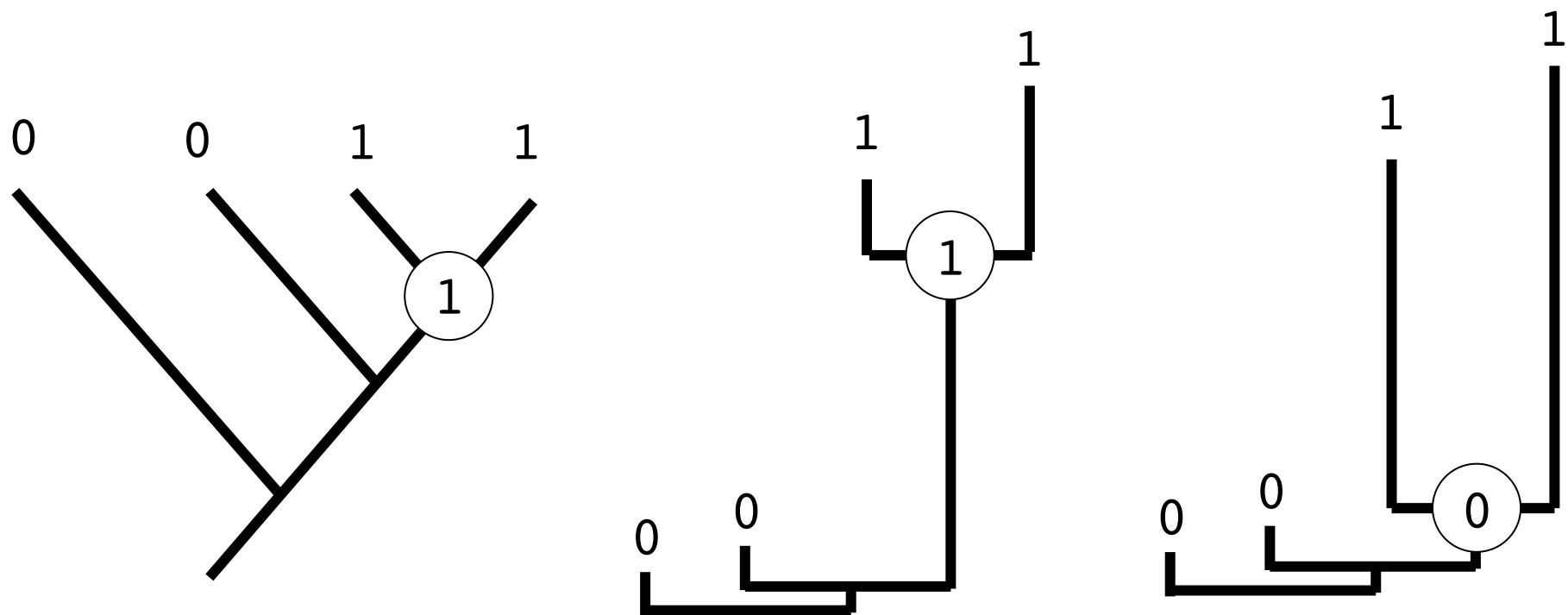
In effect: a character's weight diminishes over branch length

# Probability of change accumulates over time...

- Classic Poisson process



Parsimony will therefore ‘miss’ (or overestimate) changes



# Models of evolutionary change

- There might be some things we can know about evolution
- We can set up ‘rules’ to constrain character evolution
- Consider third codon positions

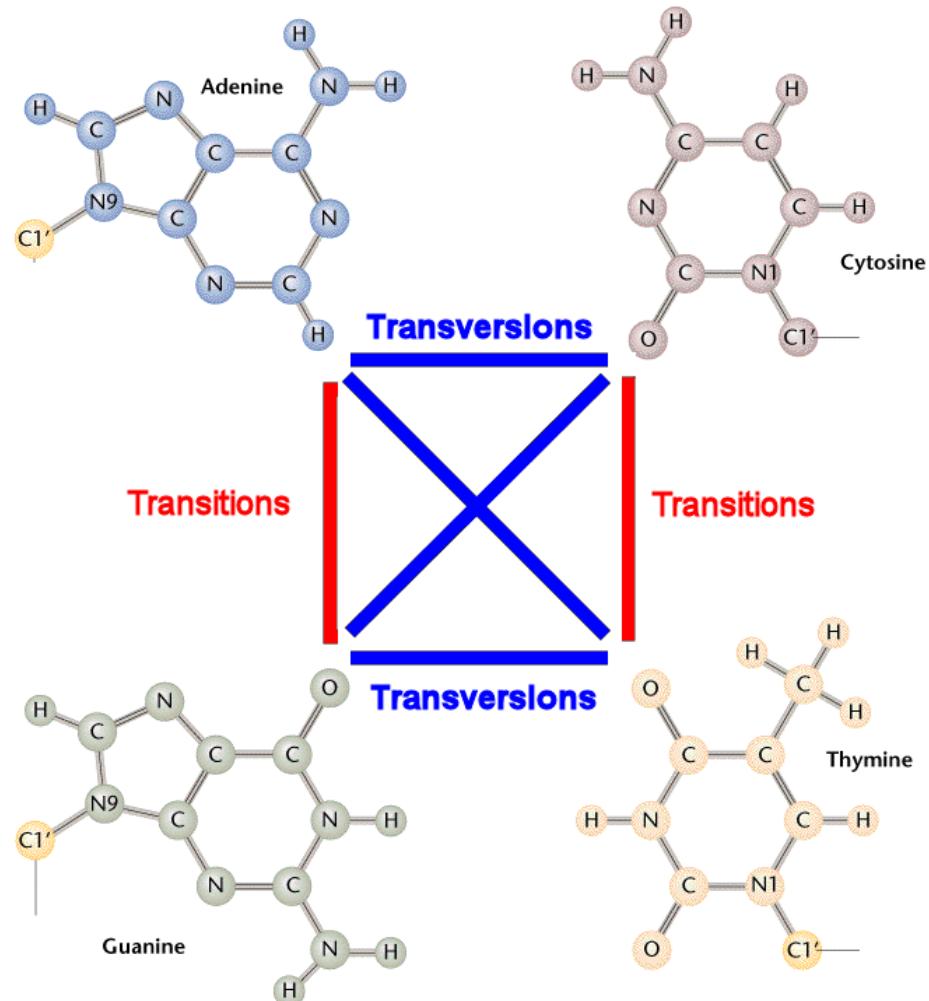
		Second Position							
		U		C		A		G	
		code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino
First Position	U	UUU	phe	UCU	ser	UAU	tyr	UGU	cy
		UUC		UCC		UAC		UGC	
		UUA	leu	UCA		UAA	STOP	UGA	
		UUG		UCG		UAG	STOP	UGG	
C	C	CUU	leu	CCU	pro	CAU	his	CGU	ar
		CUC		CCC		CAC		CGC	
		CUA		CCA		CAA		CGA	
		CUG		CCG		CAG		CGG	
A	A	AUU	ile	ACU	thr	AAU	asn	AGU	se
		AUC		ACC		AAC		AGC	
		AUA		ACA		AAA		AGA	

# Non-coding DNA

- Non-coding DNA is more likely to evolve neutrally
- e.g introns vs. exons
- So, it will evolve according to different “rules” than coding DNA

# Transition/transversion ratios

- Approx. 2/3 of single base polymorphisms are transitions
- Changes within types are more frequent
- i.e. pyr -> pyr more frequent than pyr->pur
- And vice versa.



# The Q matrix revisited

- Now we can specify a maximum likelihood model of evolutionary change
- Weights are, in effect, rates of change

	A	G	C	T		A	G	C	T	
A	-	1	1	1		A	-	0.5	0.2	0.5
G	1	-	1	1		G	0.5	-	0.5	0.2
C	1	1	-	1		C	0.2	0.5	-	0.5
T	1	1	1	-		T	0.5	0.2	0.5	-

# Real probabilities are rarely known

- We therefore set constraints
- Need to estimate parameter values

	A	G	C	T
A	-	0.2	0.5	0.2
G	0.2	-	0.2	0.5
C	0.5	0.2	-	0.2
T	0.5	0.2	0.5	-

# Real probabilities are rarely known

- We therefore set constraints
- Need to estimate parameter values
- E.g. Kimura two-parameter model

	A	G	C	T
A	-	$\kappa$	1	1
G	$\kappa$	-	1	1
C	1	1	-	$\kappa$
T	1	1	$\kappa$	-

$\kappa$  = transition/  
transversion rate  
ratio

# More complicated models

$$\mu\pi_k$$

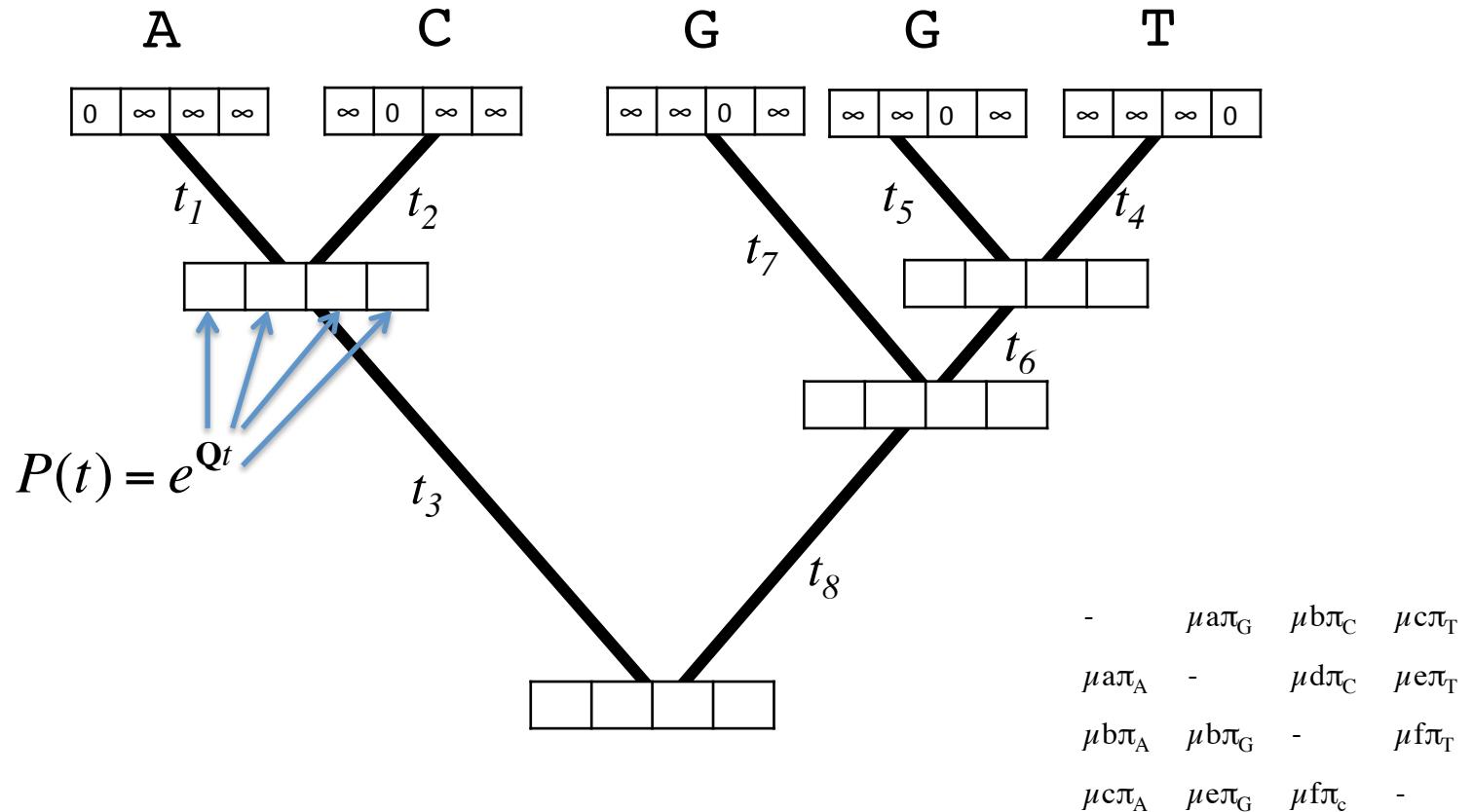
- $\mu$  is the instantaneous rate probability (usually drawn from a random distribution, e.g. gamma)
- And  $\pi_k$  is the frequency of base  $k$  (i.e. A, C, G or T)

# More complicated models

	A	G	C	T
A	-	$\mu a\pi_G$	$\mu b\pi_C$	$\mu c\pi_T$
G	$\mu a\pi_A$	-	$\mu d\pi_C$	$\mu e\pi_T$
C	$\mu b\pi_A$	$\mu b\pi_G$	-	$\mu f\pi_T$
T	$\mu c\pi_A$	$\mu e\pi_G$	$\mu f\pi_c$	-

# Calculating the likelihood of a tree

- Calculate probabilities for all possible nucleotides (or states) at each node
- Add them up.



# Conclusions

- Ancestral state reconstructions allow us to use trees to infer past evolutionary states/episodes
- Parsimony and likelihood methods intimately related
- Likelihood allows explicit modelling of time dimensions
- Likelihood methods allow for explicit models involving rates and the estimation of those parameters