

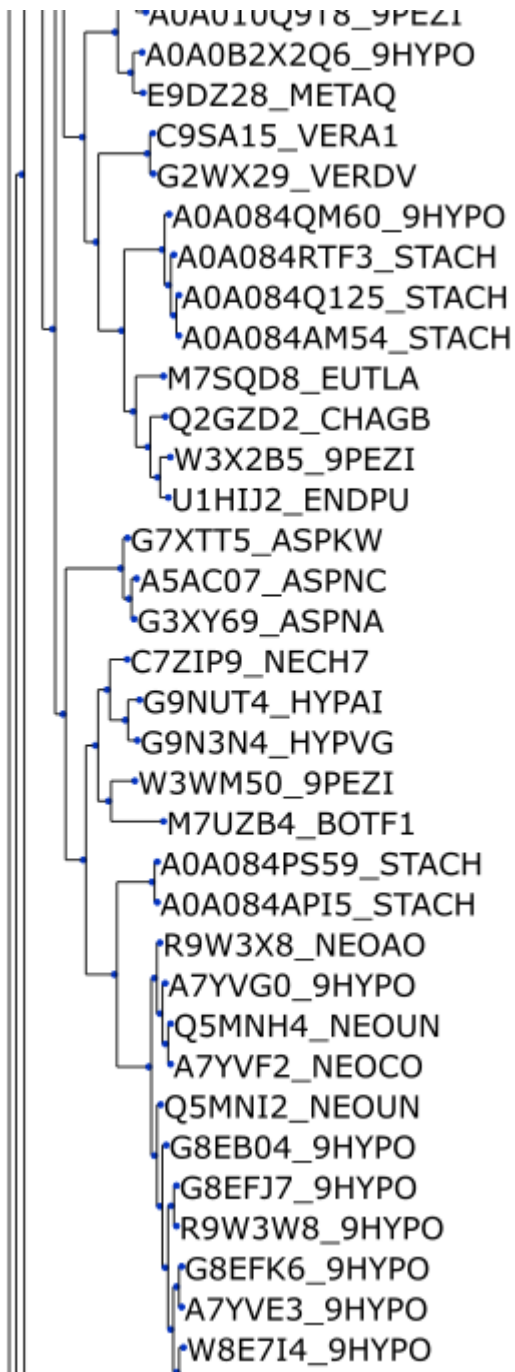
Why is it useful to know a programming language when doing phylogenomics?



## Sequences obtained from UniProt

```
>tr|C8VAU1|C8VAU1_EMENI MAT2 proteinMating type HMG-box protein [Source:UniProtKB/TrEMBLAcc:Q7Z8M2] OS=Emericella nidulans (strain FGSC A4 / ATCC 38163 / CBS 112.46 / NRRL 194 / M139) GN=MAT2 PE=4 SV=1
MAAVSIAMKSPTQSPDSITELLWKDALRHLGSTNDEVLLPTNVVDIIGQDNVEKIKSRLS
ALLGAPVVSFVDESINALRVLRTPTFSGSSISVASPSRALDSWPSEPPNKPASMKPAK
IPRPPNAFILYRQHHPKVKEARPDLSNNEISVIIGKKWRAEPEEGKLHFKNLAEFFKKK
HAE EYPDYQYTPRKPSEKKRRAASRISPKNSKRTVALENP GSMTAPSSNVFTPQMYPGIQ
NGQLAGAGYIGYLDGLNSMVNTGGLTDEPTNFGTNAFNSLFQQPQSDYGR TALFPQLEFA
GPSLGD SLEFPEFAADYF
>tr|Q7Z8M2|Q7Z8M2_EMEND MAT2 protein OS=Emericella nidulans GN=matA-2 PE=2 SV=1
MAAVSIAMKSPTQSPDSITELLWKDALRHLGSTNDEVLLPTNVVDIIGQDNVEKIKSRLS
ALLGAPVVSFVDESINALRVLRTPTFSGSSISVASPSRALDSWPSEPPNKPASMKPAK
IPRPPNAFILYRQHHPKVKEARPDLSNNEISVIIGKKWRAEPEEGKLHFKNLAEFFKKK
HAE EYPDYQYTPRKPSEKKRRAASRISPKNSKRTVALENP GSMTAPSSNVFTPQMYPGIQ
NGQLAGAGYIGYLDGLNSMVNTGGLTDEPTNFGTNAFNSLFQQPQSDYGR TALFPQLEFA
GPSLGD SLEFPEFAADYF
>tr|Q5B3Z6|Q5B3Z6_EMENI Uncharacterized protein OS=Emericella nidulans (strain FGSC A4 / ATCC 38163 / CBS 112.46 / NRRL 194 / M139) GN=AN4734.2 PE=4 SV=1
MAAVSIAMKSPTQSPDSITELLWKDALRHLGSTNDEVLLPTNVVDIIGQDNVEKIKSRLS
ALLGAPVVSFVDESINALRVLRTPTFSGSSISVASPSRALDSWPSEPPNKPASMKPAK
IPRPPNAFILYRQHHPKVKEARPDLSNNEISEEFKKKHAE EYPDYQYTPRKPSEKKRRA
ASRISPKNSKRTVALENP GSMTAPSSNVFTPQMYPGIQNGQLAGAGYIGYLDGLNSMVNT
GGLTDEPTNFGTNAFNSLFQQPQSDYGR TALFPQLEFAGPSLGD SLEFPEFAADYF
>tr|A0A0U5G7S6|A0A0U5G7S6_9EURO Putative Mating type protein MAT1-2 OS=Aspergillus calidoustus GN=ASPCAL10920 PE=4 SV=1
MATVSIAMKSTAQSPENVMERLWQDÄLRHLGSTNNEVLLPTNVVDVIGQNNVEEIKSRLC
ALLGAPVVAFIDESINALRVMRTPAFSGTTISVASHQGMAGVKSVEAPGKPRAAIKPPK
IPRPPNAFILYRQHHPKVKEAYPDLSNNEISVILGQWKSEPDEARLRFKSMAEEFFKKK
HAE EYPDYQYTPRKPSEKKRRATSRQSPKSKRSIPLESPPSIAAPSPAFTPSMYSELQG
NDTMIEGYPNPLDNFDNLNFMIDSGALADEHTGYDMNSFDALLHQAPNDYGRGTFFQHLG
IADQSIGDSFEFPDYAANCF
>tr|A0A0F8XMY0|A0A0F8XMY0_9EURO Uncharacterized protein OS=Aspergillus rambellii GN=ARAM_000112 PE=4 SV=1
MTAVSIAMKSVTTSTDNLTELLWQDÄLRHLGATNNEVLLPTNVMDLIGQDNVDKIKSRLC
VLLGAPVVAFVDESINVLRLMRTPVFSGSSISIASHDMFGNKSNEASSKVRAASTKSTK
IPRPPNAFILYRQHHPKPRVKEAYPDLSNNQISIILGQWKAEESEVKVHFKD LAEEFFKKK
HAEDHPDYQYTPRKPSEKKRRATSRSPKHVKSSQDLKYPTPTSASPNAFTPTTYPDVQN
GNIAITGYTDDLTDNLNMFDS SSVSERHSSFD SNAFDTLFQQPQDEYARATLYPHMNLAE
QSFGDSFEFPNFP GDCF
```

How can I get my sequences to have a shorter, more  
informative, header?

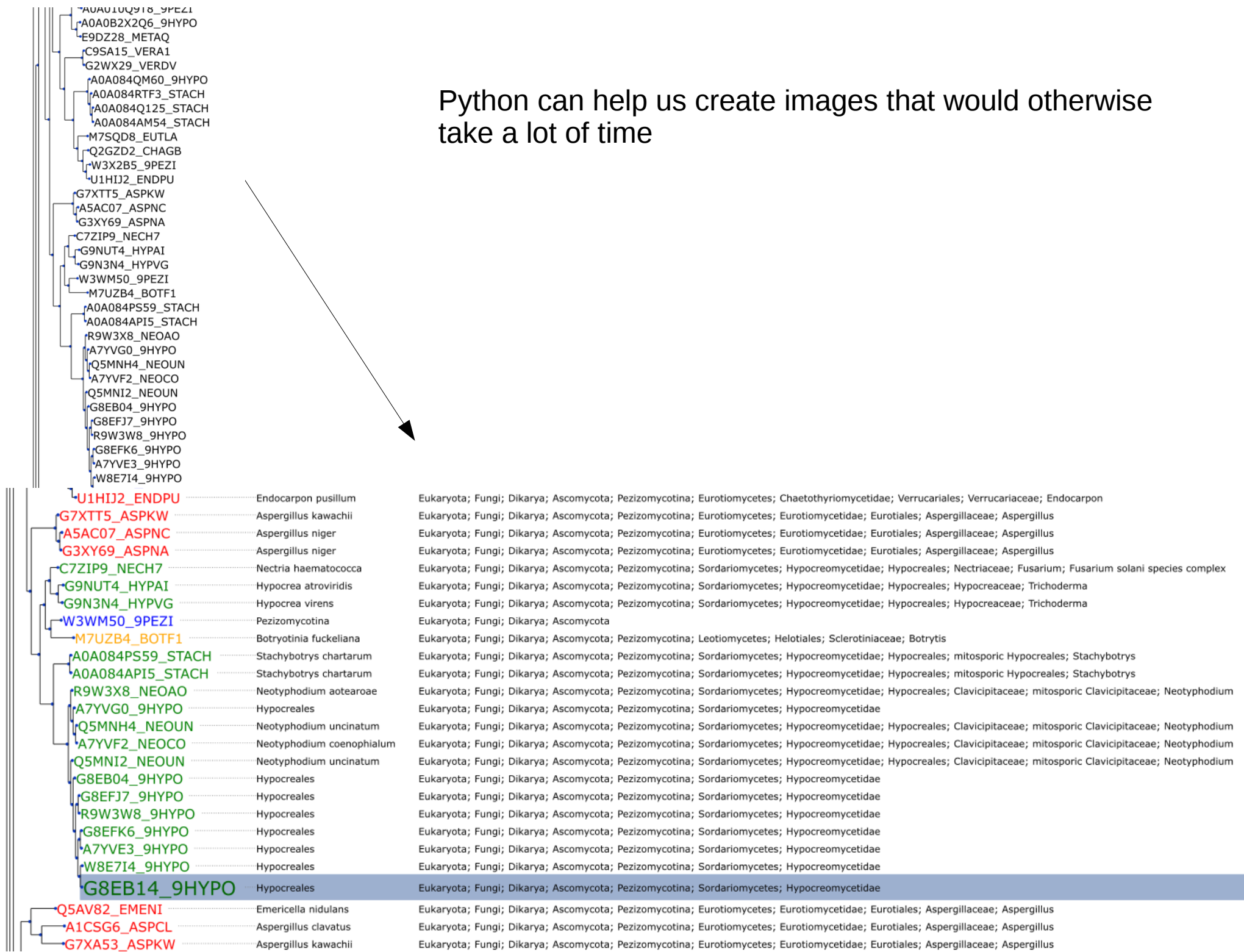


You're interested in a group of genes that perform a certain metabolic function.

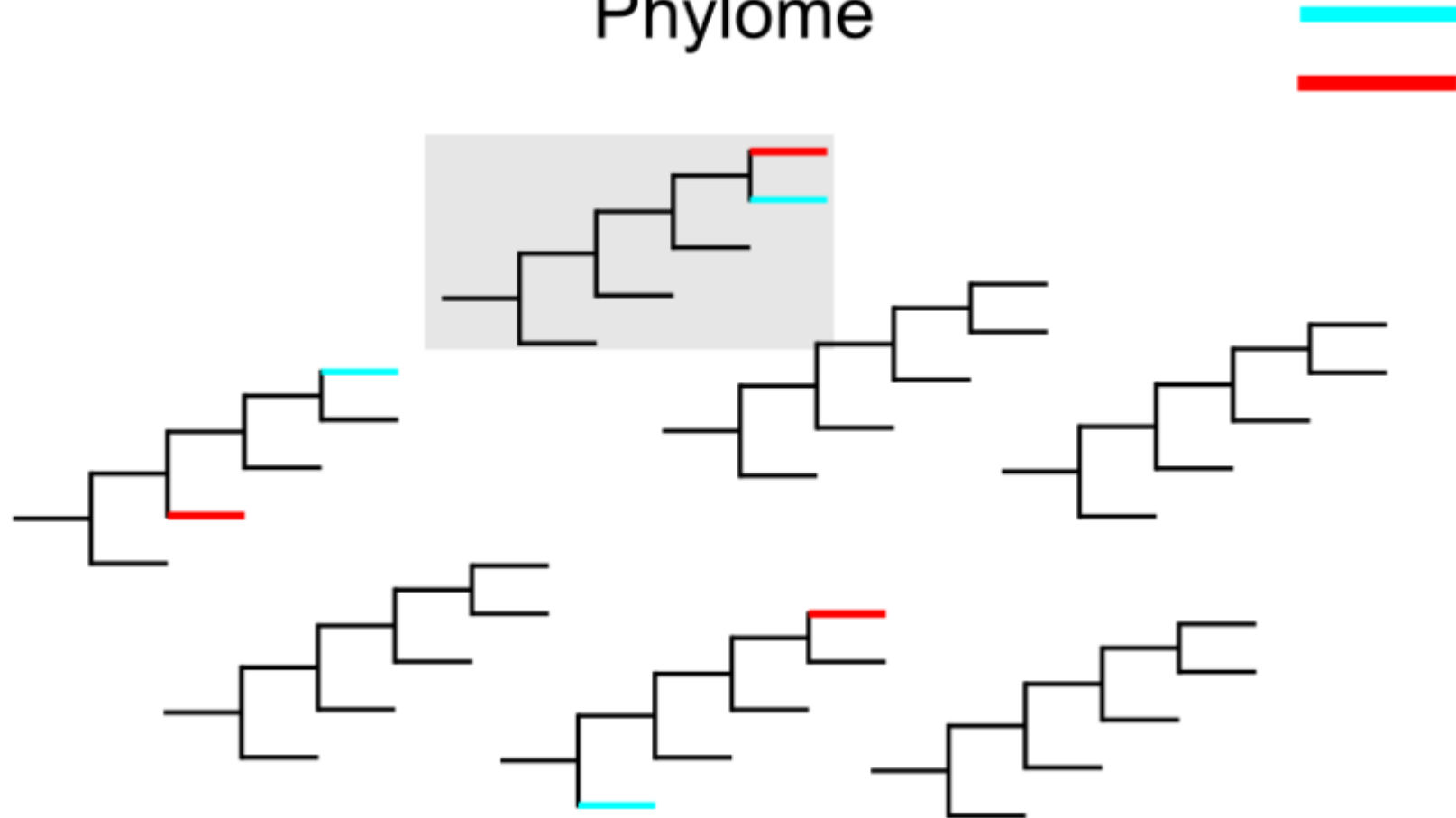
To study their evolution you have reconstructed a gene tree for every gene in the pathway.

You have found something interesting in the trees and now you want to put them nicely for your paper

Python can help us create images that would otherwise take a lot of time



# Phylome



A phylome is the complete collection of phylogenetic trees for each gene encoded in a genome. We can obtain a lot of data from it, but it implies working with a huge amount of trees.

You will probably need a tree manipulating software to get any kind of information from it.

For those of you that would like to learn python, here are two tutorials:

<http://www.cmbi.ru.nl/pythoncourse/>

<http://cscircles.cemc.uwaterloo.ca/>

For those of you already know how to program in python: ETE

<http://etetoolkit.org/>

**ETE Toolkit**

A Python framework to work with trees

[Home](#)

[Gallery](#)

[Documentation](#) ▾

[TreeView](#)

[Support](#)

[About](#)

[Download](#)

latest v3.0.0b35

A Python framework for the analysis and visualization of trees.



Download



Python API



Cookbook



Phylogenomic tools



Contribute

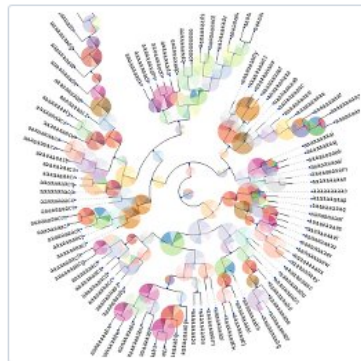
```
from ete3 import Tree
tree = Tree('((A,B), D);')

print tree
#      /-A
#     /-|
#    --| \-B
#       \-D

A = tree & "A"
A.up.show()
```

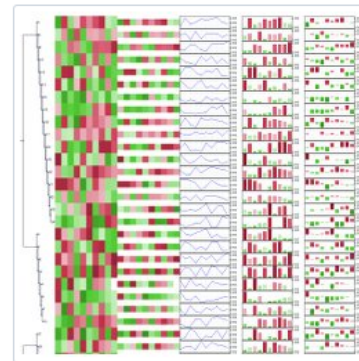
## Trees as Python objects

Load, create, traverse, search, prune, or modify hierarchical tree structures with ease using the ETE Python API.



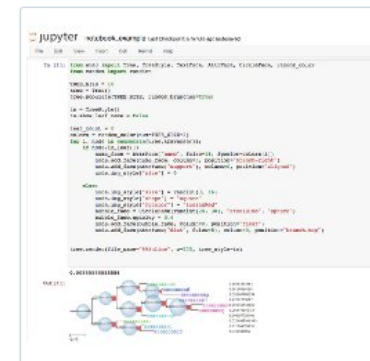
## Programmatic tree visualization

Get full control of your tree images. Browse them interactively or render SVG, PNG or PDF images.



## Tree annotation

Custom node attributes can be rendered as graphical elements. Choose among external images, charts, symbols, text labels, and more!



## Jupyter notebook support

Prototype your methods using the Jupyter notebook framework including inline visualization of trees.

## **What is ETE?**

- ETE (Environment for Tree Exploration) is a package of tools to work with trees in a programmatic way.
- We need some simple concepts of scripting programming
- There are other programming toolkits with more specific scopes: (DendroPy, BioPython, BioPerl).



ETE will allow you to:

- Read and write newick format
- Allows for node annotation
- Manipulate large tree structures
- Calculate distances among nodes
- Re-root trees
- Search nodes and partitions
- Iterate along the tree topology
- Create nice tree images
- And a lot more...

# Tree Basics

A phylogenetic tree can be uploaded in ETE and this will create a tree object.

In ETE a tree object is nothing more than a group of tree nodes connected in a hierarchical way

To create a new tree object in ETE you only need to call the `Tree()` command.

```
import ete3  
t = ete3.Tree()
```

This will just create an empty tree node.

You can upload trees from either a file or a string:

```
t = ete3.Tree("file_name")  
t = ete3.Tree("((A,B),C);")
```

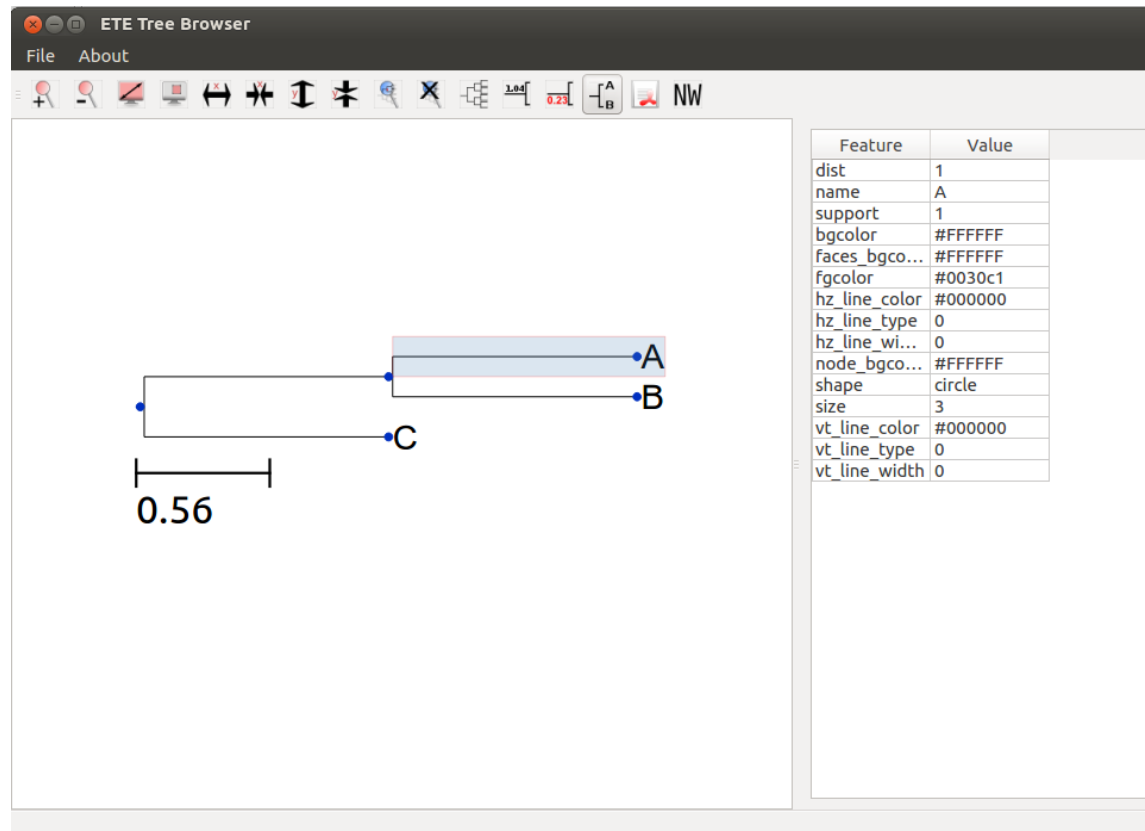
You can also write the tree either on the screen or into a file

```
t = ete3.write()  
t = ete3.write(outfile=file_name)
```

# Tree Basics

ETE also includes a visualizer that allows you to look at your tree:

`t.show()`



The window is interactive and will allow you to manually check out information and modify the tree if you want.

# Tree Basics

What is the difference between a Tree and a PhyloTree in ETE?

A PhyloTree includes all the options of a Tree but includes many more that are strictly designed to work with phylogenetic trees.

For a PhyloTree we will need to provide the species information. By default ETE will assume that the first three letters of each leaf indicate the species.

```
t= ete3.PhyloTree("((Prot1_Human,Prot1_Chimp),Prot1_Rat);")
```

Species will be: Pro for all leaves.

To define a new way to get the species names:

```
def species_name(node):  
    return node.split("_")[1]
```

This small function says that each name should be split by the “\_” and the second element should be returned.

```
t=ete3.PhyloTree("((Prot1_Human,Prot1_Chimp),Prot1_Rat);",sp_naming_function=  
species_name)
```

