PhyloNetworks and SNaQ Tutorial

Claudia Solís-Lemus

April 7th, 2016

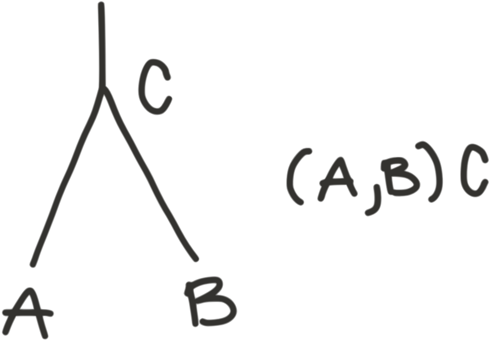
PhyloNetworks

- Julia package for phylogenetic networks
- `www.github.com/crsl4/PhyloNetworks`

- Read/write networks in parenthetical format
- Plot networks
- Root networks
- Estimate max pseudolikelihood network, bootstrap: SNaQ
- Future: trait evolution models on networks, better parallelization

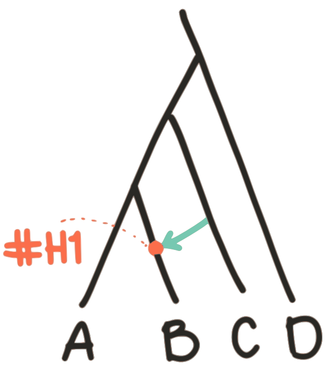
Extended Newick format

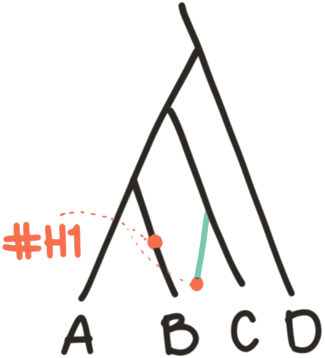
Recall that internal nodes can have name:



Extended Newick format

Split the hybrid node into two nodes with the same name:



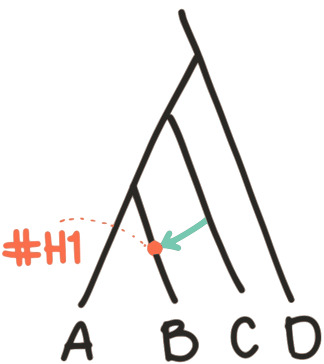


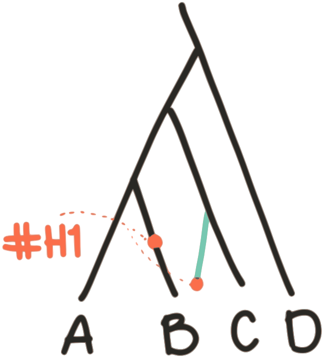
By convention:

- hybrid name: `# + H,LGT,R + number`
- minor hybrid edge leads to leaf

Extended Newick format

Split the hybrid node into two nodes with the same name:

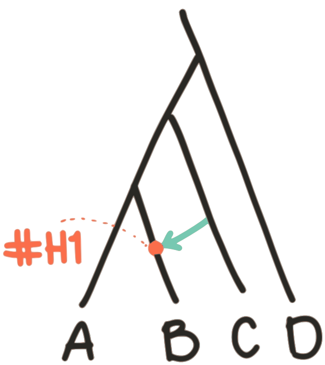


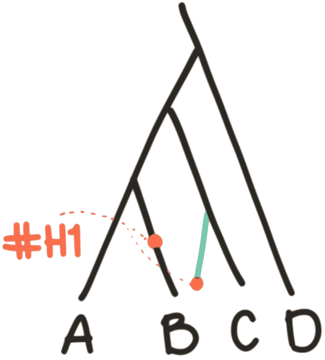


```
(((A,(B)#H1),(C,#H1)),D);
```

Extended Newick format

Split the hybrid node into two nodes with the same name:

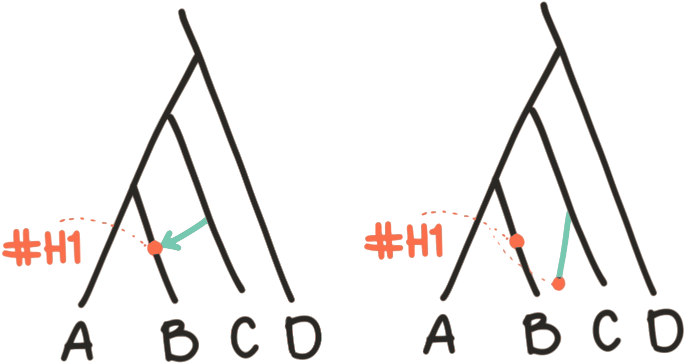




```
(((A,(B)#H1),(C,#H1)),D);
```

Extended Newick format

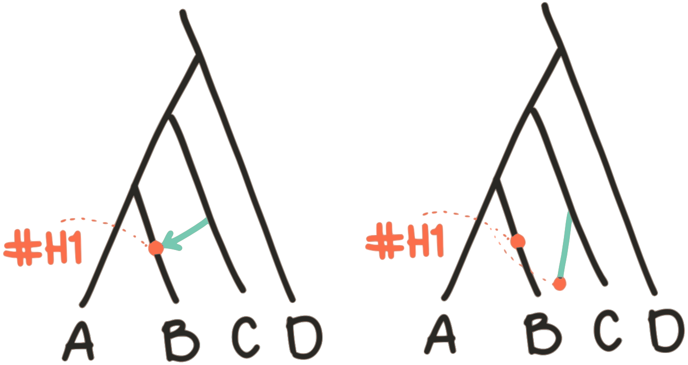
Split the hybrid node into two nodes with the same name:



`(C,#H1):branch length:bootstrap support:gamma`

Extended Newick format

Split the hybrid node into two nodes with the same name:



```
(((A,(B)#H1:::0.9),(C,#H1:::0.1)),D);
```

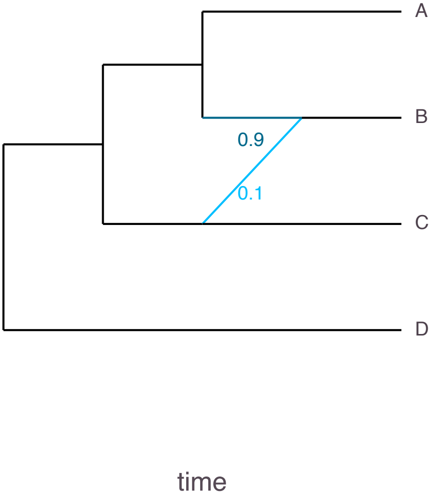
Read/write networks in parenthetical format

```
net =
readTopology("(((A,(B)#H1:::0.9),(C,#H1:::0.1)),D);")

writeTopology(net)
```

`net` is a Julia object of type `HybridNetwork`. It can be a tree.

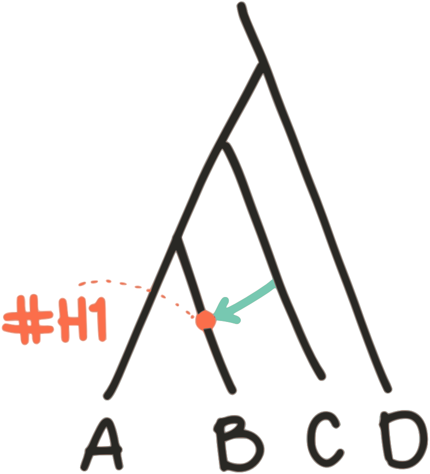
Plot networks

```
plot(net)
```



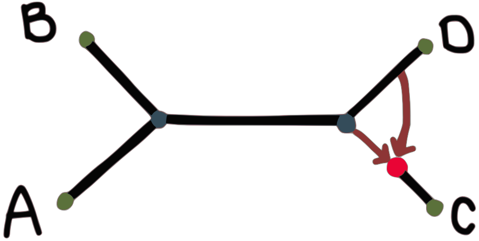
Root networks

Careful with root position



Root networks

Careful with root position:
semi-directed networks



Root networks

```
rootonedge!(net,edge#)
rootatnode!(net,node#)
```

```
plot(net, showEdgeNumber=true, showEdgeLength=false)
plot(net, showNodeNumber=true, showEdgeLength=false)
```

Also: `rootatnode!(net,outgroup)`
Julia convention: `!` in function means it modifies arguments

Estimate max pseudolik network with SNaQ

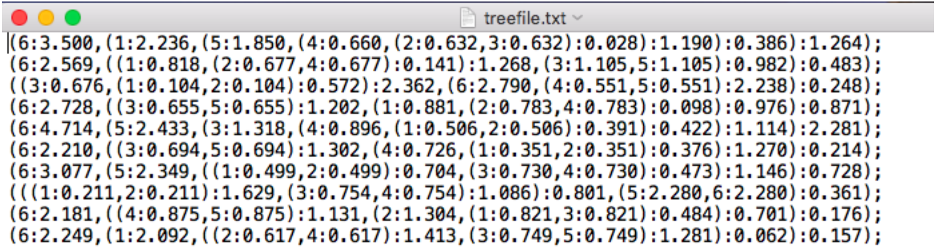
You need:

1. - List of gene trees or
   - Table of CF

2. Starting topology for the search

Input: list of gene trees

Plain text file: one tree in parenthetical format per line



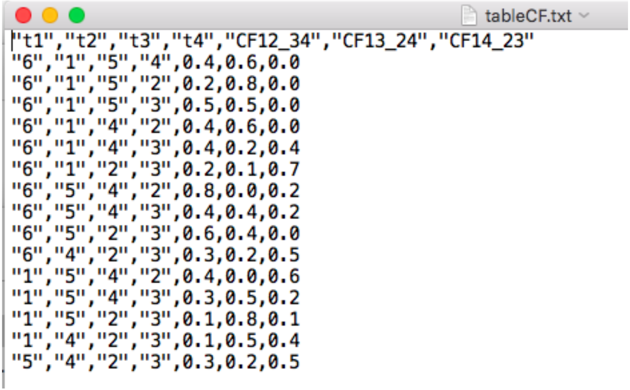
```
(6:3.500,(1:2.236,(5:1.850,(4:0.660,(2:0.632,3:0.632):0.028):1.190):0.386):1.264);
(6:2.569,((1:0.818,(2:0.677,4:0.677):0.141):1.268,(3:1.105,5:1.105):0.982):0.483);
((3:0.676,(1:0.104,2:0.104):0.572):2.362,(6:2.790,(4:0.551,5:0.551):2.238):0.248);
(6:2.728,((3:0.655,5:0.655):1.202,(1:0.881,(2:0.783,4:0.783):0.098):0.976):0.871);
(6:4.714,(5:2.433,(3:1.318,(4:0.896,(1:0.506,2:0.506):0.391):0.422):1.114):2.281);
(6:2.210,((3:0.694,5:0.694):1.302,(4:0.726,(1:0.351,2:0.351):0.376):1.270):0.214);
(6:3.077,(5:2.349,((1:0.499,2:0.499):0.704,(3:0.730,4:0.730):0.473):1.146):0.728);
(((1:0.211,2:0.211):1.629,(3:0.754,4:0.754):1.086):0.801,(5:2.280,6:2.280):0.361);
(6:2.181,((4:0.875,5:0.875):1.131,(2:1.304,(1:0.821,3:0.821):0.484):0.701):0.176);
(6:2.249,(1:2.092,((2:0.617,4:0.617):1.413,(3:0.749,5:0.749):1.281):0.062):0.157);
```

RAxML or MrBayes will put gene trees in separate files, you need to combine these files into one (PhyloNetwork does not do this!)

Input: table of CF

Plain text file, comma separated (or ;), 7 columns:

1. Taxon 1
2. Taxon 2
3. Taxon 3
4. Taxon 4
5. $CF_{12|34}$
6. $CF_{13|24}$
7. $CF_{14|23}$

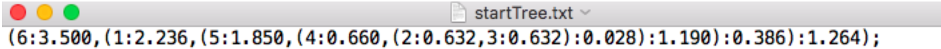


```
"t1","t2","t3","t4","CF12_34","CF13_24","CF14_23"
"6","1","5","4",0.4,0.6,0.0
"6","1","5","2",0.2,0.8,0.0
"6","1","5","3",0.5,0.5,0.0
"6","1","4","2",0.4,0.6,0.0
"6","1","4","3",0.4,0.2,0.4
"6","1","2","3",0.2,0.1,0.7
"6","5","4","2",0.8,0.0,0.2
"6","5","4","3",0.4,0.4,0.2
"6","5","2","3",0.6,0.4,0.0
"6","4","2","3",0.3,0.2,0.5
"1","5","4","2",0.4,0.0,0.6
"1","5","4","3",0.3,0.5,0.2
"1","5","2","3",0.1,0.8,0.1
"1","4","2","3",0.1,0.5,0.4
"5","4","2","3",0.3,0.2,0.5
```

This involves running BUCKy on every 4-taxon subset of the taxa, and get the CF from the three possible splits from the .concordance file into a table (PhyloNetworks does not do this!)

Input: starting topology

Plain text file: tree (or networks)in parenthetical format



```
(6:3.500,(1:2.236,(5:1.850,(4:0.660,(2:0.632,3:0.632):0.028):1.190):0.386):1.264);
```

BUCKy can give you a primary concordance tree (but recall that you ran BUCKy analyses on subsets of taxa before, not all taxa)

Input: list of gene trees or table of CF

There is a lot of pre-processing before running SNaQ

Input: list of gene trees or table of CF

There is a lot of pre-processing before running SNaQ

Luckily for us: `https://github.com/nstenz/TICR`

Input: table of CF and starting tree with TICR

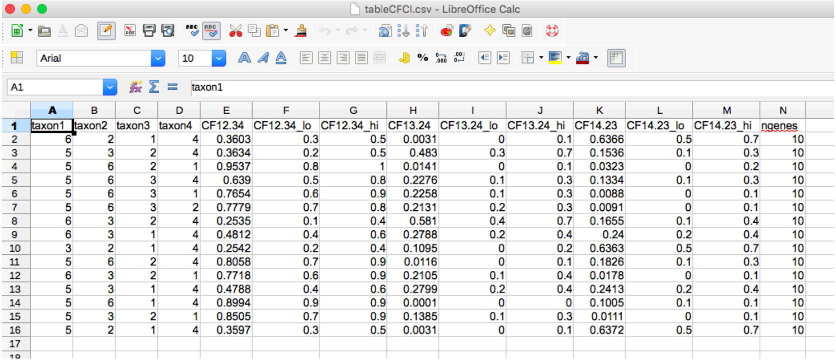
`https://github.com/nstenz/TICR`

- MDL to delimit loci
- MrBayes to perform individual gene analyses
- BUCKy to estimate quartet concordance factors
- Quartet MaxCut to estimate a binary population tree

`TICR` gives us everything we need!

Input: table of CF with TICR

https://github.com/nstenz/TICR



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	taxon1	taxon2	taxon3	taxon4	CF12.34	CF12.34_lo	CF12.34_hi	CF13.24	CF13.24_lo	CF13.24_hi	CF14.23	CF14.23_lo	CF14.23_hi	ngenes
2	6	2	1	4	0.3603	0.3	0.5	0.0031	0	0.1	0.6366	0.5	0.7	10
3	5	3	2	4	0.3634	0.2	0.5	0.483	0.3	0.7	0.1536	0.1	0.3	10
4	5	6	2	1	0.9537	0.8	1	0.0141	0	0.1	0.0323	0	0.2	10
5	5	6	3	4	0.639	0.5	0.8	0.2276	0.1	0.3	0.1334	0.1	0.3	10
6	5	6	3	1	0.7654	0.6	0.9	0.2258	0.1	0.3	0.0088	0	0.1	10
7	5	6	3	2	0.7779	0.7	0.8	0.2131	0.2	0.3	0.0091	0	0.1	10
8	6	3	2	4	0.2535	0.1	0.4	0.581	0.4	0.7	0.1655	0.1	0.4	10
9	6	3	1	4	0.4812	0.4	0.6	0.2788	0.2	0.4	0.24	0.2	0.4	10
10	3	2	1	4	0.2542	0.2	0.4	0.1095	0	0.2	0.6363	0.5	0.7	10
11	5	6	2	4	0.8058	0.7	0.9	0.0116	0	0.1	0.1826	0.1	0.3	10
12	6	3	2	1	0.7718	0.6	0.9	0.2105	0.1	0.4	0.0178	0	0.1	10
13	5	3	1	4	0.4788	0.4	0.6	0.2799	0.2	0.4	0.2413	0.2	0.4	10
14	5	6	1	4	0.8994	0.9	0.9	0.0001	0	0	0.1005	0.1	0.1	10
15	5	3	2	1	0.8505	0.7	0.9	0.1385	0.1	0.3	0.0111	0	0.1	10
16	5	2	1	4	0.3597	0.3	0.5	0.0031	0	0.1	0.6372	0.5	0.7	10
17														

Network estimation with SNaQ

```
d = readTableCF("tableCFCI.csv")
T = readTopologyLevel1("startTree.txt")
```

Estimate best network with $h_{max} = 1, runs = 10$ (default):

```
estNet1 = snaq!(T,d)
plot(estNet1)
```

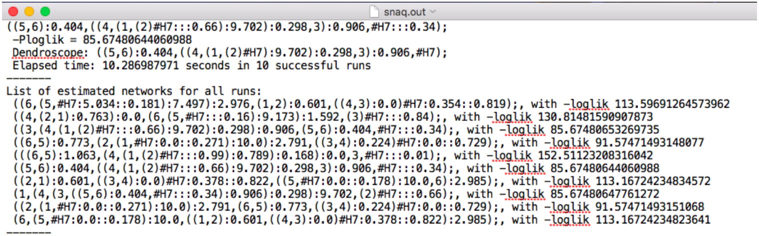
You can also change the number of independent runs:

```
estNet1 = snaq!(T,d, runs=5)
```

Suggestion for big datasets: Start with `runs=1`

SNaQ output files: .out file

`snaq.out`



```
((5,6):0.404,((4,(1,(2)#H7:::0.66):9.702):0.298,3):0.906,#H7:::0.34);
 -Ploglik = 85.67480644060988
 Dendroscope: ((5,6):0.404,((4,(1,(2)#H7):9.702):0.298,3):0.906,#H7);
 Elapsed time: 10.286987971 seconds in 10 successful runs
-------
List of estimated networks for all runs:
 ((6,(5,#H7:5.034::0.181):7.497):2.976,(1,2):0.601,((4,3):0.0)#H7:0.354::0.819);, with -loglik 113.59691264573962
 ((4,(2,1):0.763):0.0,(6,(5,#H7:::0.16):9.173):1.592,(3)#H7:::0.84);, with -loglik 130.81481590907873
 ((3,(4,(1,(2)#H7:::0.66):9.702):0.298):0.906,(5,6):0.404,#H7:::0.34);, with -loglik 85.67480653269735
 ((6,5):0.773,(2,(1,#H7:0.0::0.271):10.0):2.791,((3,4):0.224)#H7:0.0::0.729);, with -loglik 91.57471493148077
 (((6,5):1.063,(4,(1,(2)#H7:::0.99):0.789):0.168):0.0,3,#H7:::0.01);, with -loglik 152.51123208316042
 ((5,6):0.404,((4,(1,(2)#H7:::0.66):9.702):0.298,3):0.906,#H7:::0.34);, with -loglik 85.67480644060988
 ((2,1):0.601,((3,4):0.0)#H7:0.378::0.822,((5,#H7:0.0::0.178):10.0,6):2.985);, with -loglik 113.16724234834572
 (1,(4,(3,((5,6):0.404,#H7:::0.34):0.906):0.298):9.702,(2)#H7:::0.66);, with -loglik 85.67480647761272
 ((2,(1,#H7:0.0::0.271):10.0):2.791,(6,5):0.773,((3,4):0.224)#H7:0.0::0.729);, with -loglik 91.57471493151068
 (6,(5,#H7:0.0::0.178):10.0,((1,2):0.601,((4,3):0.0)#H7:0.378::0.822):2.985);, with -loglik 113.16724234823641
-------
```

Default rootname: snaq (can be set with the option `filename=`)

−*Pseudolik*: the smaller, the better!

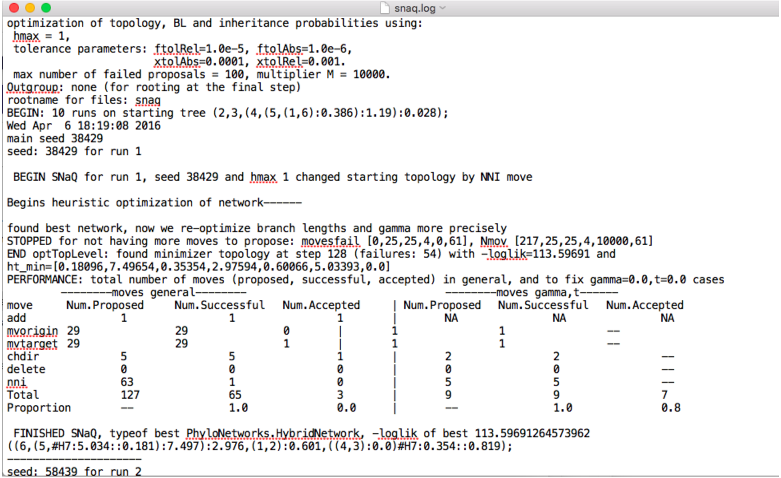
You can read the best network overall, and the best network per run with:

`nets = readInputTrees("snaq.out")`

and plot one of them: `plot(nets[2])`

SNaQ output files: .log file

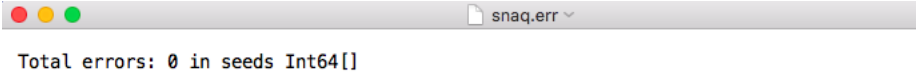
snaq.log



```
optimization of topology, BL and inheritance probabilities using:
 hmax = 1,
 tolerance parameters: ftolRel=1.0e-5, ftolAbs=1.0e-6,
                       xtolAbs=0.0001, xtolRel=0.001.
 max number of failed proposals = 100, multiplier M = 10000.
Outgroup: none (for rooting at the final step)
rootname for files: snaq
BEGIN: 10 runs on starting tree (2,3,(4,(5,(1,6):0.386):1.19):0.028);
Wed Apr  6 18:19:08 2016
main seed 38429
seed: 38429 for run 1

 BEGIN SNaQ for run 1, seed 38429 and hmax 1 changed starting topology by NNI move

Begins heuristic optimization of network------

found best network, now we re-optimize branch lengths and gamma more precisely
STOPPED for not having more moves to propose: movesfail [0,25,25,4,0,61], Nmov [217,25,25,4,10000,61]
END optTopLevel: found minimizer topology at step 128 (failures: 54) with -loglik=113.59691 and
ht_min=[0.18096,7.49654,0.35354,2.97594,0.60066,5.03393,0.0]
PERFORMANCE: total number of moves (proposed, successful, accepted) in general, and to fix gamma=0.0,t=0.0 cases
        --------moves general--------                               --------moves gamma,t------
move     Num.Proposed     Num.Successful   Num.Accepted     | Num.Proposed   Num.Successful   Num.Accepted
add               1                1                1       |       NA               NA               NA
mvorigin 29               29               0        |       1                1                --
mvtarget 29               29               1        |       1                1                --
chdir             5                5                1       |       2                2                --
delete            0                0                0       |       0                0                --
nni               63               1                0       |       5                5                --
Total             127              65               3       |       9                9                7
Proportion        --               1.0              0.0     |       --               1.0              0.8

 FINISHED SNaQ, typeof best PhyloNetworks.HybridNetwork, -loglik of best 113.59691264573962
((6,(5,#H7:5.034::0.181):7.497):2.976,(1,2):0.601,((4,3):0.0)#H7:0.354::0.819);
---------------------
seed: 58439 for run 2
```

SNaQ output files: .err file

`snaq.err`



```
Total errors: 0 in seeds Int64[]
```

Example:
Total errors: 1 in seeds [4545]
You need to run the following function with the same settings that caused the error:
`snaqDebug(T,d,seed=4545)`

Estimating bigger networks

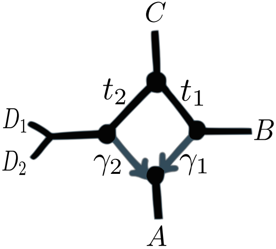
```
estNet2 = snaq!(estNet1,d,hmax=2,runs=3)
```

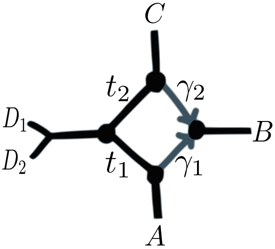
Suggestion 1: Start the estimation for $h_{max} = h$ in the best network at $h_{max} = h - 1$

Suggestion 2: Keep in mind the **level-1 restriction**, maybe no more hybridizations can be added if the number of taxa is small!

Troubleshooting: my network does not make sense!

- Recall that the root is meaningless, use root functions
- Recall the identifiability issue with direction of hybrid edges:





use `topologyMaxQPseudolik!(newNet,d)`

- Start the estimation for $h_{max} = h$ in the best network at $h_{max} = h - 1$
- Recall **level-1** assumption: maybe no more hybridizations can be added
- Recall that the bigger the network, the more gene trees are needed

Bugs and errors

- Use PhyloNetworks users google group (link in Github)
- Use "Issues" in Github
- Send me an email `claudia@stat.wisc.edu`, specially with output from `snaqDebug`

Getting help

- Use PhyloNetworks users google group (link in Github)
- README file in Github has step-by-step commands
- `?plot` inside Julia
- Soon: `http://phylonetworks.readthedocs.org/en/latest/`
- Send me an email `claudia@stat.wisc.edu`

Running julia as a script

```
julia myscript.jl arguments
```

Arguments: vector separated by spaces `ARGS`, e.g., `ARGS[1]`

Bootstrap

```
using DataFrames
df = readtable("tableCFCI.csv")
bootNet = bootsnaq(T,df,hmax=1,nrep=10,bestNet=estNet1,
runs=1)
```

`nrep` is the number of bootstrap replicates
`bestNet` is the estimated network with original data
`bootNet` is a vector of size `nrep` with estimated network per replicate
Output files: .out, .log, .err per bootstrap replicate, and `bootsnaq.out`, `bootsnaq.log`

Bootstrap: summary

- `df0,tree0 = treeEdgesBootstrap(bootNet,estNet1)` will calculate the bootstrap support of tree edges by edge number in `df0`, and the main underlying tree in `tree0`
- `plot(tree0, showEdgeNumber=true, showEdgeLength=false)`

Bootstrap: summary

`HFmat,discTrees = hybridDetection(bootNet,estNet1,outgroup)` to summarize bootstrap on hybrid nodes: `HFmat` will have one row per bootstrap network, and number of columns depending on number of hybrids in `estNet1`: for example, if `estNet1` had 3 hybrids for example, `HFmat` will have 6 columns:

- first 3 columns indicate the presence (1.0) or absense (0.0) of each hybrid (column) for each bootstrap network (row)
- the last 3 columns indicate the estimated gamma in the bootstrap network if the hybrid was found (0.0 if not found)

Bootstrap: summary

Only makes sense to summarize networks that have the same underlying tree: `discTrees` has different trees to the underlying tree in `estNet1`
`length(discTrees)`

`dfhyb = summarizeHFdf(HFmat)`
`dfhyb` has one row per hybrid, and 5 columns:

- hybrid index
- number of trees that match the underlying tree in estNet1 (same for all hybrids)
- number of networks with that hybrid
- mean estimated gamma among networks with the hybrid
- sd estimated gamma among networks with the hybrid

Note that last row contains in 3er column the number of networks that have all same hybrids as estNet1 (hybrid index, mean gamma and sd gamma are meaningless)