Algorithm:

```
def OriginLikelihood(Tree):

    TM       = Tree.resolvedtree
    bp       = Tree.branchpositions
    bp       = bp[rows(TM):]
    branches = Tree.filledtimeFractions*Tree.depth*1000

    LL       = np.zeros(rows(TM))
    LL_D     = np.zeros(rows(TM))
    NN       = np.zeros(rows(TM))
    TT       = branches[:,-1]
    DD       = np.zeros(rows(TM))
    Live     = np.zeros(rows(TM))

    D         = np.copy(Tree.D)
    np.fill_diagonal(D, 1)
    lnD       = - np.log(D)
    lnD       = np.zeros(np.shape(lnD))

    for bb in bp:
        r, c    = bb[0], bb[1]
        id      = TM[r, c]
        tu      = np.where(TM[:, c] == id)[0]
        bhat    = branches[tu, c]
        THat    = TM[tu, c:]
        NNCount = NN[tu] + 1
        TTCount = TT[tu] + branches
        LLHat   = LL[tu]

        for p in tu:
            if Live[p] == 1:
                LL[p]   = LL[p] + NN[p] * (np.log(NN[p]) - np.log(TT[p]))
                LL_D[p] = LL_D[p] + DD[p]

        LL_DHat = LL_D[tu]

        DHat    = (lnD[tu, :])[:, tu]

        z = 0
        while True:
            ids  = uniq(THat[:, z])
            nums = THat[:, z]
            z    = z + 1
            if len(ids) > 1:
                break

        TTHat = np.zeros(len(nums))
        NNHat = np.zeros(len(nums))
        DDHat = np.zeros(len(nums))

        for m in ids:
            posi   = np.where(nums == m)[0]
            posni  = np.where(nums != m)[0]
            toAdd  = TTCount[posni]
            for q in posi:
                maxFinder = NNCount[posni]*(np.log(NNCount[posni]) \
                    - np.log(toAdd)) + LL_DHat[posni] + DHat[q, posni] + LLHat[posni]
                maxm      = np.argmax(maxFinder)
                TTHat[q]  = toAdd[maxm]
                NNHat[q]  = NNCount[posni[maxm]]
                DDHat[q]  = DHat[q, posni[maxm]] + LL_DHat[posni[maxm]]

        for p in range(0, rows(tu)):
            TT[tu[p]] = TTHat[p]
            NN[tu[p]] = NNHat[p]
            DD[tu[p]] = DDHat[p]
            if Live[tu[p]] == 0:
                Live[tu[p]] = 1

    for p in tu:
        if Live[p] == 1:
            LL[p]   = LL[p] + NN[p] * (np.log(NN[p]) - np.log(TT[p]))
            LL_D[p] = LL_D[p] + DD[p]
```

```
return LL_D + LL
```

1 Example

The execution of the initial block of code:

```
TM       = Tree.resolvedtree
bp       = Tree.branchpositions
bp       = bp[rows(TM):]
branches = Tree.filledtimeFractions*Tree.depth*1000

LL       = np.zeros(rows(TM))
LL_D     = np.zeros(rows(TM))
NN       = np.zeros(rows(TM))
TT       = branches[:,-1]
DD       = np.zeros(rows(TM))
Live     = np.zeros(rows(TM))

D         = np.copy(Tree.D)
np.fill_diagonal(D, 1)
lnD       = - np.log(D)
lnD       = np.zeros(np.shape(lnD))
```

Gives us the following entities:

$$TM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 & 4 \\ 0 & 1 & 2 & 3 & 5 \\ 0 & 1 & 2 & 4 & 6 \\ 0 & 1 & 2 & 4 & 7 \end{bmatrix}$$

With branch matrix:

$$branches = \begin{bmatrix} b_1 & b_2 & b_4 & . & b_8 \\ b_1 & b_2 & b_4 & . & b_9 \\ b_1 & b_2 & b_5 & . & b_{10} \\ b_1 & b_2 & b_5 & b_6 & b_{11} \\ b_1 & b_2 & b_5 & b_6 & b_{12} \\ b_1 & b_3 & . & b_7 & b_{13} \\ b_1 & b_3 & . & b_7 & b_{14} \\ b_1 & b_3 & . & . & b_{15} \end{bmatrix}$$

The list of (interior) branch positions is:

$$bp = \begin{bmatrix} [4,3] \\ [6,3] \\ [1,2] \\ [4,2] \\ [4,1] \\ [7,1] \\ [7,0] \end{bmatrix}$$

That is all we need to start the iterations, along with a distance matrix. we keep track of things in a table:

LL	LL_D	TT	NN	DD	Live
0	0	b_8	0	0	0
0	0	b_9	0	0	0
0	0	b_10	0	0	0
0	0	b_11	0	0	0
0	0	b_12	0	0	0
0	0	b_13	0	0	0
0	0	b_14	0	0	0
0	0	b_15	0	0	0

The distance matrix is:

$$\ln(D) = \begin{bmatrix} 0 & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} & d_{18} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} & d_{28} \\ d_{31} & d_{32} & 0 & d_{34} & d_{35} & d_{36} & d_{37} & d_{38} \\ d_{41} & d_{42} & d_{43} & 0 & d_{45} & d_{46} & d_{47} & d_{48} \\ d_{51} & d_{52} & d_{53} & d_{54} & 0 & d_{56} & d_{57} & d_{58} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & 0 & d_{67} & d_{68} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & 0 & d_{78} \\ d_{81} & d_{82} & d_{83} & d_{84} & d_{85} & d_{86} & d_{87} & 0 \end{bmatrix}$$

1.1 Looping over branches

We now enter the loop, which, in the first iteration, is fairly simple. We have $r, c = 4, 3$:

$$\texttt{NNCount} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\texttt{bhat} = \begin{bmatrix} b_6 \\ b_6 \end{bmatrix}$$

and then:

$$\texttt{TTCount} = \begin{bmatrix} b_6 + b_{10} \\ b_6 + b_{11} \end{bmatrix}$$

The rest of the values are all zero.

1.1.1 Looping over ids at a given branch

We then enter the loop that splits the subtree. Coming out of the loop, we have something like:

LL	LL_D	TT	NN	DD	Live
0	0	b_8	0	0	0
0	0	b_9	0	0	0
0	0	$b_1 0$	0	0	0
0	0	$b_6 + b_{12}$	1	0	1
0	0	$b_6 + b_{11}$	1	0	1
0	0	$b_1 3$	0	0	0
0	0	$b_1 4$	0	0	0
0	0	$b_1 5$	0	0	0

1.1.2 Similarly...

After popping the next two branches, the algorithm is really just starting to get rolling, so we will have:

LL	LL_D	TT	NN	DD	Live
0	0	$b_4 + b_9$	1	d_{12}	1
0	0	$b_4 + b_8$	1	d_{21}	1
0	0	$b_1 0$	0	0	0
0	0	$b_6 + b_{12}$	1	d_{45}	1
0	0	$b_6 + b_{11}$	1	d_{54}	1
0	0	$b_7 + b_{14}$	1	d_{67}	1
0	0	$b_7 + b_{13}$	1	d_{76}	1
0	0	$b_1 5$	0	0	0

And this is basically it for the low-hanging fruit.

1.1.3 Interior branches

So, we now pop a triple branch, to get:

$$\texttt{NNCount} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

$$\texttt{bhat} = \begin{bmatrix} b_5 \\ b_5 \\ b_5 \end{bmatrix}$$

and then:

$$\texttt{TTCount} = \begin{bmatrix} b_5 + b_{10} \\ b_5 + b_6 + b_{10} \\ b_5 + b_6 + b_{11} \end{bmatrix}$$

Note since two of the branches were already "live", the likelihood has to be updated and we get:

LL	LL_D	TT	NN	DD	Live
0		$b_4 + b_9$	1	d_{12}	1
0	0	$b_4 + b_8$	1	d_{21}	1
0	0	b_{10}	0	0	0
$-\ln(b_6 + b_{12})$	d_{45}	$b_6 + b_{12}$	1	d_{45}	1
$-\ln(v_6 + b_{11})$	d_{54}	$b_6 + b_{11}$	1	d_{54}	1
0	0	$b_7 + b_{14}$	1	d_{67}	1
0	0	$b_7 + b_{13}$	1	d_{76}	1
0	0	$b_1 5$	0	0	0

Now, we loop over the in and out group entries. In our case, we first have to deal with the other two entries. As written, we will go through the loop and arrive at:

LL	LL_D	TT	NN	DD	Live
0	0	$b_4 + b_9$	1	d_{12}	1
0	0	$b_4 + b_8$	1	d_{21}	1
0	0	$b_5 + b_6 + b_{11}$	2	d_{34}	1
$-\ln(b_6 + b_{12})$	d_{45}	$b_5 + b_{10}$	1	d_{43}	1
$-\ln(v_6 + b_{11})$	d_{54}	$b_5 + b_{10}$	1	d_{53}	1
0	0	$b_7 + b_{14}$	1	d_{67}	1
0	0	$b_7 + b_{13}$	1	d_{76}	1
0	0	b_{15}	0	0	0

1.2 Next branch

Now, we pop branch b_2. This is a big one and we are at the place where the rubber really hits the road. Our two most important entities are:

$$\texttt{NNCount} = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 2 \\ 2 \end{bmatrix}$$

$$\texttt{bhat} = \begin{bmatrix} b_2 \\ b_2 \\ b_2 \\ b_2 \\ b_2 \end{bmatrix}$$

and then:

$$\texttt{TTCount} = \begin{bmatrix} b_2 + b_4 + b_9 \\ b_2 + b_4 + b_8 \\ b_2 + b_5 + b_6 + b_{11} \\ b2 + b5 + b_{10} \\ b2 + b_5 + b_{10} \end{bmatrix}$$

Everyone is now live. So, now, we have the following interim update:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9)$	d_{12}	$b_4 + b_9$	1	d_{12}	1
$-\ln(b_4 + b_8)$	d_{21}	$b_4 + b_8$	1	d_{21}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11}))$	d_{34}	$b_5 + b_6 + b_{11}$	2	d_{34}	1
$-\ln(b_6 + b_{12}) + \ln(b_5 + b_{10})$	$d_{45} + d_{43}$	$b_5 + b_{10}$	1	d_{43}	1
$-\ln(v_6 + b_{11}) + \ln(b_5 + b_{10})$	$d_{54} + d_{53}$	$b_5 + b_{10}$	1	d_{53}	1
0	0	$b_7 + b_{14}$	1	d_{67}	1
0	0	$b_7 + b_{13}$	1	d_{76}	1
0	0	b_{15}	0	0	0

At first, we have the in group at positions $0, 1$ and the out group at positions $2, 3, 4$. So, we suppose the branch with three entities along it is selected as the max. For the other group, we will suppose that the first of the top two is selected. This means that in the end, we have:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9)$	d_{12}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{13}	1
$-\ln(b_4 + b_8)$	d_{21}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{23}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11}))$	d_{34}	$b_2 + b_4 + b_9$	2	d_{31}	1
$-\ln(b_6 + b_{12}) + \ln(b_5 + b_{10})$	$d_{45} + d_{43}$	$b_2 + b_4 + b_9$	2	d_{41}	1
$-\ln(v_6 + b_{11}) + \ln(b_5 + b_{10})$	$d_{54} + d_{53}$	$b_2 + b_4 + b_9$	2	d_{51}	1
0	0	$b_7 + b_{14}$	1	d_{67}	1
0	0	$b_7 + b_{13}$	1	d_{76}	1
0	0	b_{15}	0	0	0

And that is how things should stand after the next iteration. The next branch to be popped is the bottom three groups. So, have an interim update, that gives:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9)$	d_{12}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{13}	1
$-\ln(b_4 + b_8)$	d_{21}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{23}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11}))$	d_{34}	$b_2 + b_4 + b_9$	2	d_{31}	1
$-\ln(b_6 + b_{12}) + \ln(b_5 + b_{10})$	$d_{45} + d_{43}$	$b_2 + b_4 + b_9$	2	d_{41}	1
$-\ln(b_6 + b_{11}) + \ln(b_5 + b_{10})$	$d_{54} + d_{53}$	$b_2 + b_4 + b_9$	2	d_{51}	1
$-\ln(b_7 + b_{14})$	d_{67}	$b_7 + b_{14}$	1	d_{67}	1
$-\ln(b_7 + b_{13}$	d_{76}	$b_7 + b_{13}$	1	d_{76}	1
0	0	b_{15}	0	0	0

The important entities that we then create are:

$$\texttt{NNCount} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$

$$\texttt{bhat} = \begin{bmatrix} b_2 \\ b_2 \\ b_2 \end{bmatrix}$$

and then:

$$\texttt{TTCount} = \begin{bmatrix} b_3 + b_7 + b_{14} \\ b_3 + b_7 + b_{14} \\ b_3 + b_{15} \end{bmatrix}$$

Accordingly, at the end of the loop, we have:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9)$	d_{12}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{13}	1
$-\ln(b_4 + b_8)$	d_{21}	$b_2 + b_5 + b_6 + b_{11}$	3	d_{23}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11}))$	d_{34}	$b_2 + b_4 + b_9$	2	d_{31}	1
$-\ln(b_6 + b_{12}) + \ln(b_5 + b_{10})$	$d_{45} + d_{43}$	$b_2 + b_4 + b_9$	2	d_{41}	1
$-\ln(b_6 + b_{11}) + \ln(b_5 + b_{10})$	$d_{54} + d_{53}$	$b_2 + b_4 + b_9$	2	d_{51}	1
$-\ln(b_7 + b_{14})$	d_{67}	$b_3 + b_{15}$	1	d_{68}	1
$-\ln(b_7 + b_{13}$	d_{76}	$b_3 + b_{15}$	1	d_{78}	1
0	0	$b_3 + b_7 + b_{14}$	2	d_{86}	1

1.3 Final pass through the loop

On the final pass through the loop, we have an interim update across the board - all groups are live and included in the last branch. So:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9) + 3(\ln(3) - \ln(b_2 + b_5 + b_6 + b_{11}))$	$d_{12} + d_{13}$	$b_2 + b_5 + b_6 + b_{11}$	3	d_{13}	1
$-\ln(b_4 + b_8) + 3(\ln(3) - \ln(b_2 + b_5 + b_6 + b_{11}))$	$d_{21} + d_{23}$	$b_2 + b_5 + b_6 + b_{11}$	3	d_{23}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11})) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{34} + d_{31}$	$b_2 + b_4 + b_9$	2	d_{31}	1
$-\ln(b_6 + b_{12}) - \ln(b_5 + b_{10}) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{45} + d_{43} + d_{41}$	$b_2 + b_4 + b_9$	2	d_{41}	1
$-\ln(b_6 + b_{11}) - \ln(b_5 + b_{10}) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{54} + d_{53} + d_{51}$	$b_2 + b_4 + b_9$	2	d_{51}	1
$-\ln(b_7 + b_{14}) - \ln(b_3 + b_{15}$	$d_{67} + d_{68}$	$b_3 + b_{15}$	1	d_{68}	1
$-\ln(b_7 + b_{13} - \ln(b_3 + b_{15}$	$d_{76} + d_{68}$	$b_3 + b_{15}$	1	d_{78}	1
$2(\ln(2) - \ln(b_3 + b_7 + b14))$	d_{86}	$b_3 + b_7 + b_{14}$	2	d_{86}	1

Now, there is no need to rewrite anything. On our final pass, we get:

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9) + 3(\ln(3) - \ln(b_2 + b_5 + b_6 + b_{11}))$	$d_{12} + d_{13}$	$b_1 + b_3 + b_{15}$	2	d_{16}	1
$-\ln(b_4 + b_8) + 3(\ln(3) - \ln(b_2 + b_5 + b_6 + b_{11}))$	$d_{21} + d_{23}$	$b_1 + b_3 + b_{15}$	2	d_{26}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11})) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{34} + d_{31}$	$b_1 + b_3 + b_{15}$	2	d_{36}	1
$-\ln(b_6 + b_{12}) - \ln(b_5 + b_{10}) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{45} + d_{43} + d_{41}$	$b_1 + b_3 + b_{15}$	2	d_{46}	1
$-\ln(b_6 + b_{11}) - \ln(b_5 + b_{10}) + 2(\ln(2) - \ln(b_2 + b_4 + b_9))$	$d_{54} + d_{53} + d_{51}$	$b_1 + b_3 + b_{15}$	2	d_{56}	1
$-\ln(b_7 + b_{14}) - \ln(b_3 + b_{15}$	$d_{67} + d_{68}$	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{61}	1
$-\ln(b_7 + b_{13} - \ln(b_3 + b_{15}$	$d_{76} + d_{68}$	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{71}	1
$2(\ln(2) - \ln(b_3 + b_7 + b14))$	d_{86}	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{81}	1

And we are at the end of the loop. We next have to perform one more update to get everything in the right place.

LL	LL_D	TT	NN	DD	Live
$-\ln(b_4 + b_9) + 3(\ln(3) -$ $\ln(b_2 + b_5 + b_6 + b_{11})) +$ $2(\ln(2) - \ln(b_1 + b_3 + b_{15})$	$d_{12} + d_{13}$	$b_1 + b_3 + b_{15}$	2	d_{16}	1
$-\ln(b_4 + b_8) + 3(\ln(3) -$ $\ln(b_2 + b_5 + b_6 + b_{11}) +$ $2(\ln(2) - \ln(b_1 + b_3 + b_{15}))$	$d_{21} + d_{23}$	$b_1 + b_3 + b_{15}$	2	d_{26}	1
$2(\ln 2 - \ln(b_5 + b_6 + b_{11})) +$ $2(\ln(2) - \ln(b_2 + b_4 + b_9)) +$ $2(\ln(2) - \ln(b_1 + b_3 + b_{15})$	$d_{34} + d_{31}$	$b_1 + b_3 + b_{15}$	2	d_{36}	1
$-\ln(b_6 + b_{12}) - \ln(b_5 + b_{10}) +$ $2(\ln(2) - \ln(b_2 + b_4 + b_9)) +$ $2(\ln(2) - \ln(b_1 + b_3 + b_{15})$	$d_{45} + d_{43} + d_{41}$	$b_1 + b_3 + b_{15}$	2	d_{46}	1
$-\ln(b_6 + b_{11}) - \ln(b_5 + b_{10}) +$ $2(\ln(2) - \ln(b_2 + b_4 + b_9)) +$ $2(\ln(2) - \ln(b_1 + b_3 + b_{15})$	$d_{54} + d_{53} + d_{51}$	$b_1 + b_3 + b_{15}$	2	d_{56}	1
$-\ln(b_7 + b_{14}) - \ln(b_3 + b_{15} +$ $4(\ln(4) - \ln(b_1 + b_2 + b_5 + b_6 + b_{11}$	$d_{67} + d_{68}$	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{61}	1
$-\ln(b_7 + b_{13} - \ln(b_3 + b_{15} +$ $4(\ln(4) - \ln(b_1 + b_2 + b_5 + b_6 + b_{11}$	$d_{76} + d_{68}$	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{71}	1
$2(\ln(2) - \ln(b_3 + b_7 + b14)) +$ $4(\ln(4) - \ln(b_1 + b_2 + b_5 + b_6 + b_{11}$	d_{86}	$b_1 + b_2 + b_5 + b_6 + b_{11}$	4	d_{81}	1