

# DPT

immediate

November 18, 2021

## Contents

Diffusion Pseudo Time (DPT) is a pseudo time metric based on the transition probability of a diffusion process ?.

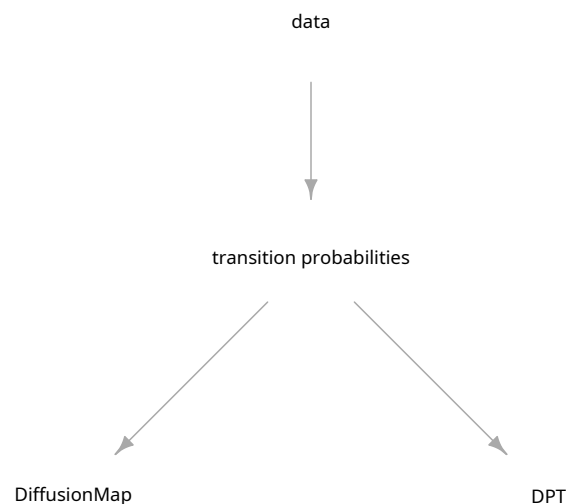
*destiny* supports DPT in addition to its primary function of creating *DiffusionMaps* from data.

```
[2]: library(destiny)    # load destiny...  
data(guo)               # ...and sample data  
library(gridExtra)      # Also we need grid.arrange
```

Registered S3 method overwritten by 'xts':

```
method      from  
as.zoo.xts  zoo
```

DPT is in practice independent of Diffusion Maps:



However in order not to overcomplicate things, in *destiny*, you have to create DPT objects from *DiffusionMap* objects.

(If you really only need the DPT, skip Diffusion Component creation by specifying `n_eigs = 0`)

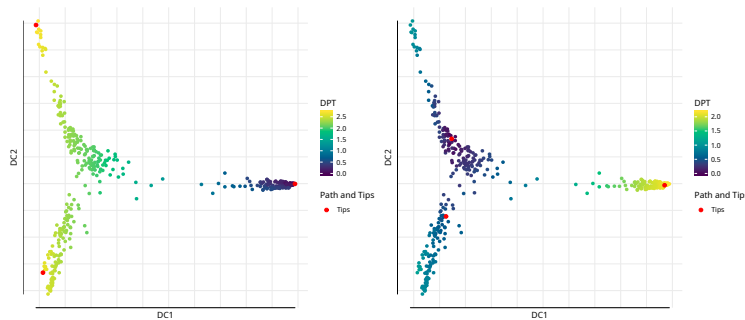
```
[4]: dm <- DiffusionMap(guo)
     dpt <- DPT(dm)
```

The resulting object of a call like this will have three automatically chosen tip cells. You can also specify tip cells:

```
[5]: set.seed(4)
     dpt_random <- DPT(dm, tips = sample(ncol(guo), 3L))
```

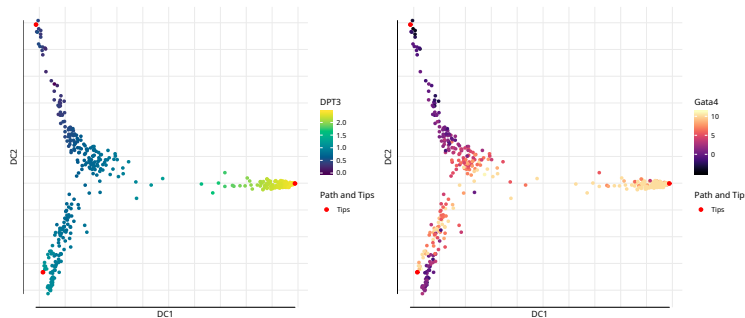
Plotting without parameters results in the DPT of the first root cell:

```
[7]: grid.arrange(plot(dpt), plot(dpt_random), ncol = 2)
```



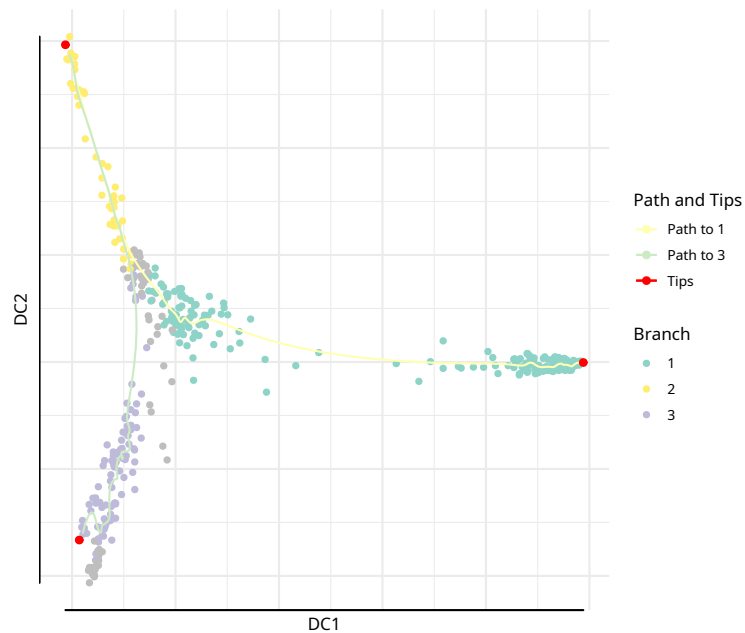
Other possibilities include the DPT from the other tips or everything supported by `plot.DiffusionMap`:

```
[8]: grid.arrange(
     plot(dpt, col_by = 'DPT3'),
     plot(dpt, col_by = 'Gata4', pal = viridis::magma),
     ncol = 2
)
```



The DPT object also contains a clustering based on the tip cells and DPT, and you can specify where to draw paths from and to:

```
[10]: plot(dpt, root = 2, paths_to = c(1,3), col_by = 'branch')
```



You can further divide branches. First simply plot branch colors like we did above, then identify the number of the branch you intend to plot, and then specify it in a subsequent plot call. In order to see the new branches best, we specify a dcs argument that visually spreads out all four branches.

```
[11]: plot(dpt, col_by = 'branch', divide = 3, dcs = c(-1,-3,2), pch = 20)
```

Warning message in title(main, sub, ...):

"\"legend\_name\" ist kein Grafikparameter"

Warning message in plot.xy(xy.coords(x, y), type = type, ...):

"\"legend\_name\" ist kein Grafikparameter"

