A Simulation of the Spread of Type 1 Diabetes in a Family Tree using Multitype Branching Processes

Alexia Spoon

Introduction

Type 1 Diabetes is a chronic health condition where an individual's body no longer creates its own insulin. This is presumed to be the result of some internal autoimmune attack on the body that results in permanent damage to the pancreas (CDC, 2022). When the pancreas no longer produces insulin, glucose is no longer able to be converted into energy, and as a result the blood glucose level of the individual rises, sometimes to dangerous levels (National Institute of Diabetes and Digestive and Kidney Diseases, 2017). Individuals that have been diagnosed with Type 1 Diabetes must take insulin multiple times a day in order to stay alive.

According to the CDC, only 0.55% of U.S. adults have been diagnosed be diagnosed with Type 1 Diabetes in 2016 (CDC, 2018). Many times, Type 1 Diabetes will present itself in individuals with no family history of the condition. However, the presence of Type 1 Diabetes in a direct family member does increase the likelihood of the diagnosis. In this case, the term direct family member is limited to a parent or sibling. If the father has Type 1 Diabetes, the likelihood of a child also getting it is increased to $\frac{1}{17}$, or approximately 5.88% (America Diabetes Association). If the mother has Type 1 Diabetes, and has a child before she turns 25, the odds of the child also having it is $\frac{1}{25}$, or 4%, and if she has a child after she turns 25, the odds of the child having it is $\frac{1}{100}$, or 1% (American Diabetes Association). If both parents have Type 1 Diabetes, then the probability of a child having it as well increase to approximately $\frac{1}{4}$, or 25% (Moore, 2021). Finally, if neither parent has Type 1 Diabetes, but a sibling has it, then the likelihood of other siblings having it as well is about $\frac{1}{20}$, or 5% (Moore, 2021).

Question/Goal

While the likelihood of a child being diagnosed with Type 1 Diabetes does increase a significant amount with the presence of the condition in a direct family member, these probabilities still remain quite low. Regardless of this, it is sometimes seen that families do have consecutive generations with the presence of Type 1 Diabetes, which leads one to believe that if a parent has Type 1 Diabetes, it is very likely that the child will also have it. This leads to the question of how likely the presence of Type 1 Diabetes in a family tree *actually* is. Can a simulation of a family tree produce trees that show a significant spread of Type 1 Diabetes when it is present, as some real-world examples of families suggest?

Methods

The concept of Branching Processes are often used to simulate family trees. This concept can be extended using Multitype Branching Processes. Multitype Branching Processes are processes that allow each individual representing in the tree to have a different set of probabilities. The Branching Processes that are usually used to simulate family trees have one significant property: the distribution of the offspring must always be independent and identically distributed, or IID. This means that the probability of having a certain amount of children will always be the same. This property changes for Multitype Branching Processes. In the case of Multitype Branching Processes, the distribution of offspring is no longer IID, but rather the distribution will depend on the type of the parent (Yeo, 2014). This allows for a more complex branching process, since more factors are able to be accounted for.

In this project, Multitype Branching Processes will be utilized. There will be two types of individuals represented in the processes: individuals where Type 1 Diabetes is present, and individuals where Type 1 Diabetes is not present. The distribution of offspring when the parent does have Type 1 Diabetes will follow a Poisson distribution with $\lambda = 1$. Additionally, the distribution of offspring when the parent does not have Type 1 Diabetes will also follow a Poisson distribution, but in this case the parameter will be $\lambda = 1.5$.

These λ's were chosen following the assumption that parents with Type 1 Diabetes will, on average, have slightly less children than parents where Type 1 Diabetes is not present. It is common for individuals with Type 1 Diabetes to think about whether or not they should have children, given that their children will have higher chances of being diagnosed with Type 1 Diabetes as well. Based on this, the fear of passing this condition onto children could easily lead the average number of children with parents who have Type 1 Diabetes to be slightly less than their non-diabetic counterparts.

Once the number of offspring have been determined for a given parent, based on the distribution given the presence (or lack of presence) of Type 1 Diabetes, the presence of Type 1 Diabetes in each offspring must then be determined. This will be determined using the aforementioned likelihoods based on the presence of the condition in a direct family member, with some slight modifications. Specifically, the parent will always be assumed to be the father, meaning the probability of an offspring having the condition given one parent has it will be $\frac{1}{17}$. The next generation of offspring will have varying distributions depending on the "type" of each current generation's offspring, who will become the parent's in the next generation. This will continue for numerous generations, resulting in a family tree that identifies the overall percentage of the presence of Type 1 Diabetes, as well as a visual depiction that can show the spread of the condition in a family.

Results

The following function produces the offspring generation, along with the "type" of each offspring based on varying likelihoods. If Type 1 Diabetes is present in one parent, then the number of offspring will be a random value pulled from a Poisson distribution with $\lambda = 1$. If Type 1 Diabetes is not present in one parent, then the number of offspring will be a random value pulled from a Poisson distribution with $\lambda = 1.5$. Next, the presence of Type 1 Diabetes is determined for the other parent using the national percentage of Type 1 Diabetes, 0.55%, as the probability for the condition being present, and the remaining 99.45% being the probability that the condition is not present.

Next, for each offspring, the "type" is determined. If Type 1 Diabetes is present in one parent but not the other, the probability of the condition being present is $\frac{1}{17}$ and the probability of it not being present is $\frac{16}{17}$. If Type 1 Diabetes is present in both parents, the probability of the condition being present in a given offspring is $\frac{1}{4}$, with the probability of the condition not being present is $\frac{3}{4}$. If Type 1 Diabetes is not present in either parent, but it is already present in the set of offspring, then the probability of it being present of another offspring is $\frac{1}{20}$, and the probability of it not being present is $\frac{19}{20}$. If Type 1 diabetes is not present in either parent or in the set of offspring thus far, the probability of it being present in a given offspring is 0.55% and the probability of it not being present is 99.45%.

Additionally, if the number of offspring is 0, then a blank node is created to show the end of a family line. Finally, Each node is given a name to keep track of them. If it is determined that Type 1 Diabetes is present in a given offspring, it's name will be "Present" followed by the number offspring it is. Similarly, if Type 1 Diabetes is not present in a given offspring, it's name will be "Not Present" followed by the number offspring it is.

```r
add.children <- function(node, diab){
  if(diab == 0) {
    num_children <- rpois(1, lambda = 1.5)
  } else {
    num_children <- rpois(1, lambda = 1)
  }
  diabetes_present <- c()
  other_parent <- sample(c("Present", "Not Present"), 1, prob = c(0.0055, 0.9945))
  for(i in 1:num_children) {
    if(diab == 1) {
      if(other_parent == "Not Present") {
        diabetes_present <- c(diabetes_present,
                              sample(c("Present","Not Present"), 1, prob = c(1/17, 16/17)))
      } else {
          diabetes_present <- c(diabetes_present,
                                sample(c("Present","Not Present"), 1, prob = c(1/4, 3/4)))
        }
    } else {
      if("Present" %in% diabetes_present) {
        diabetes_present <- c(diabetes_present,
                              sample(c("Present","Not Present"), 1, prob = c(1/20, 19/20)))
      } else {
        diabetes_present <- c(diabetes_present,
                              sample(c("Present","Not Present"), 1, prob = c(0.0055, 0.9945)))
      }
    }
  }
  if (num_children == 0){
    node$AddChild(" ")
  } else {
    for (i in 1:num_children){
      name.new <- paste(diabetes_present[i]," ", i, sep = "")
      node$AddChild(name.new)
    }
  }
}
```

Now, the first tree being created is when Type 1 Diabetes is initially present at generation 0. The parent node is created with the name “Present”. From there, the tree is created. Since Type 1 Diabetes is initially present, the first set of offspring will follow the Poisson distribution with $\lambda = 1$. Note that the depth is set to 10, meaning that there will be a total of 11 generations (including the initial generation 0) in each tree. From there, the full tree is created, with the offspring distribution differing based on the presence of Type 1 Diabetes in the parent. After that, the tree is traversed, and depending on the “type” of each node, the nodes will be a different color. If Type 1 Diabetes is present, the node will be yellow, and if Type 1 Diabetes is not present, the node will be pink.

```
depth <- 10

# Diabetes initially present
## If sibling has it, then 5% chance
## If one parent has it, then 1/17 chance
## If two parent have it, then 1/4 chance
## If no one has it, then 0.55% chance
parent_diab <- Node$new("Present")
add.children(parent_diab, diab = 1)
node <- parent_diab$children[[1]]
if(node$name == " "){
  node <- parent_diab
}

while (node$name != parent_diab$name){
  if (node$level>depth || node$name == " "){
    node <- node$parent
    while(!(node$isLeaf) & node$name != parent_diab$name){
      temp <- node$parent
      if(length(node$siblings) > 0) {
        for(i in 1:length(node$siblings)) {
          node.sib <- node$siblings[[i]]
          if(node.sib$isLeaf) {
            temp <- node.sib
          }
        }
      }
      node <- temp
    }
  }
  if (node$level<=depth & node$isLeaf & node$name != parent_diab$name){
    if(node$name %in% c("Not Present","Not Present 1", "Not Present 2",
                        "Not Present 3", "Not Present 4", "Not Present 5",
                        "Not Present 6", "Not Present 7", "Not Present 8")) {
      add.children(node, diab = 0)
    } else {
      add.children(node, diab = 1)
    }
    if (node$leafCount > 0){
      node <- node$children[[1]]
    }
  }
}
present <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                                        c("Present", "Present 1", "Present 2",
                                          "Present 3","Present 4","Present 5",
                                          "Present 6", "Present 7", "Present 8"))
notpresent <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                            c("Not Present", "Not Present 1", "Not Present 2",
                              "Not Present 3", "Not Present 4", "Not Present 5",
                              "Not Present 6", "Not Present 7", "Not Present 8"))
extinct <- Traverse(parent_diab, filterFun = function(x) x$name %in% c(" "))
Do(present, SetNodeStyle, style = "filled", fillcolor = "#fff200",
   fontcolor = "black", inherit = FALSE)
Do(notpresent, SetNodeStyle, style = "filled", fillcolor = "#feadc9",
   fontcolor = "black", inherit = FALSE)
Do(extinct, SetNodeStyle, style = "filled", fillcolor = 'white',
   fontcolor = 'white', color = 'white', inherit = FALSE)
```

Observe the sample tree below when Type 1 Diabetes is initially present at generation 0. We can see that from generation 0 to generation 1, the parent has 1 offspring, and that offspring does have Type 1 Diabetes as well. However, after that the next generation has 2 offspring, neither of which have the condition. Thus, that family line of Type 1 Diabetes is extinct, since only the parent or a sibling can increase the odds of an offspring having the condition. Now, observe that there is a single node with Type 1 Diabetes on the left at generation 7, but this node's offspring once again do not have the condition. Finally, observe that also at generation 7, a node has Type 1 Diabetes, and passes it on the one offspring at generation 8, but this once again dies out since there are offspring with the condition. This gives a total of 5 individuals in the whole family tree that have Type 1 Diabetes. Overall, only 1.37457% of individuals in this family tree have Type 1 Diabetes, while 98.62543% do not have it. This percentage is higher than the national average, but recall that this is only one sample tree.

```
plot(parent_diab)
```

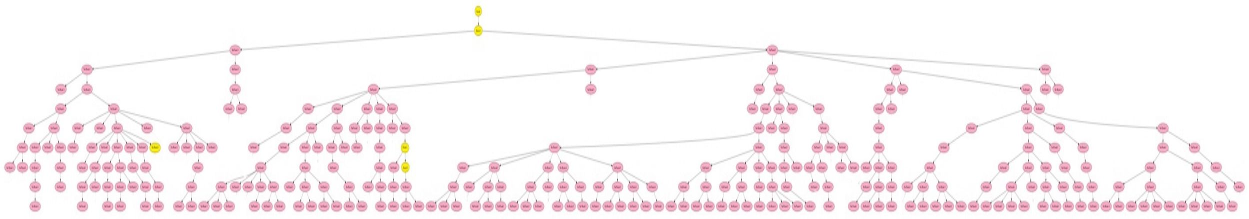

Figure 1: Type 1 Diabetes initially present

```
length(present)/(length(present)+length(notpresent))
```

```
## [1] 0.0137457
```

```
length(notpresent)/(length(present)+length(notpresent))
```

```
## [1] 0.9862543
```

Now this process will be run for 500 iterations. Therefore, 500 trees are made in the same way, and the mean of the percentages of Type 1 Diabetes presence from the 500 trees is found. This will give a better idea of the true ratio of individuals with Type 1 Diabetes in a family tree in 10 generations. From these 500 Multitype Branching Processes, it is found that, on average, 0.86021% of individuals will have Type 1 Diabetes, while 99.13979% will not.

```
pres <- c()
notpres <- c()
for(i in  1:500) {
  parent_diab <- Node$new("Present")
  add.children(parent_diab, diab = 1)
  node <- parent_diab$children[[1]]
  if(node$name == " "){
    node <- parent_diab
  }

  while (node$name != parent_diab$name){
    if (node$level>depth || node$name == " "){
      node <- node$parent
      while(!(node$isLeaf) & node$name != parent_diab$name){
        temp <- node$parent
        if(length(node$siblings) > 0) {
          for(i in 1:length(node$siblings)) {
            node.sib <- node$siblings[[i]]
            if(node.sib$isLeaf) {
              temp <- node.sib
            }
          }
        }
        node <- temp
      }
    }
    if (node$level<=depth & node$isLeaf & node$name != parent_diab$name){
      if(node$name %in% c("Present","Present 1", "Present 2",
                          "Present 3", "Present 4", "Present 5",
                          "Present 6", "Present 7", "Present 8")) {
        add.children(node, diab = 1)
      } else {
        add.children(node, diab = 0)
      }
      if (node$leafCount > 0){
        node <- node$children[[1]]
      }
    }
  }

  present <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                                      c("Present", "Present 1", "Present 2",
                                        "Present 3","Present 4","Present 5",
                                        "Present 6", "Present 7", "Present 8"))
  notpresent <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                                c("Not Present", "Not Present 1", "Not Present 2",
                                  "Not Present 3", "Not Present 4", "Not Present 5",
                                  "Not Present 6", "Not Present 7", "Not Present 8"))
  extinct <- Traverse(parent_diab, filterFun = function(x) x$name %in% c(" "))

  pres <- c(pres, length(present)/(length(present)+length(notpresent)))
  notpres <- c(notpres, length(notpresent)/(length(present)+length(notpresent)))
}
```

```
mean(pres)
```

```
## [1] 0.0086021
```

```
mean(notpres)
```

```
## [1] 0.9913979
```

Now the same process is repeated for a sample Multitype Branching Process where Type 1 Diabetes is *not* present at generation 0. This means that the number of offspring at generation 1 will follow a Poisson distribution with $\lambda = 1.5$. The same steps are followed in creating the full tree, with the offspring distribution changing depending on the parent's "type". Observe that in this tree only has 3 individual with Type 1 Diabetes, all at different generations and with no presence of the condition from the parent or any sibling. Overall, on this sample tree, Type 1 Diabetes is present in only 0.84985% of individuals, while it is not present in the remaining 99.15015%.

```
# Diabetes not initially present
## If sibling has it, then 5% chance
## If parent has it, then 1.17 chance
## If no one has it, then 0.55% chance
parent_nodiab <- Node$new("Not Present")
add.children(parent_nodiab, diab = 0)
node <- parent_nodiab$children[[1]]
```

```r
if(node$name == " "){
  node <- parent_nodiab
}

while (node$name != parent_nodiab$name){
  if (node$level>depth || node$name == " "){
    node <- node$parent
    while(!(node$isLeaf) & node$name != parent_nodiab$name){
      temp <- node$parent
      if(length(node$siblings) > 0) {
        for(i in 1:length(node$siblings)) {
          node.sib <- node$siblings[[i]]
          if(node.sib$isLeaf) {
            temp <- node.sib
          }
        }
      }
      node <- temp
    }
  }
  if (node$level<=depth & node$isLeaf & node$name != parent_nodiab$name){
    if(node$name %in% c("Present","Present 1", "Present 2",
                        "Present 3", "Present 4", "Present 5",
                        "Present 6", "Present 7", "Present 8")) {
      add.children(node, diab = 1)
    } else {
      add.children(node, diab = 0)
    }
    if (node$leafCount > 0){
      node <- node$children[[1]]
    }
  }
}

present <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                                        c("Present", "Present 1", "Present 2",
                                          "Present 3","Present 4","Present 5",
                                          "Present 6", "Present 7", "Present 8"))
notpresent <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                            c("Not Present", "Not Present 1", "Not Present 2",
                              "Not Present 3", "Not Present 4", "Not Present 5",
                              "Not Present 6", "Not Present 7", "Not Present 8"))
extinct <- Traverse(parent_nodiab, filterFun = function(x) x$name %in% c(" "))
Do(present, SetNodeStyle, style = "filled", fillcolor = "#fff200",
   fontcolor = "black", inherit = FALSE)
Do(notpresent, SetNodeStyle, style = "filled", fillcolor = "#feadc9",
   fontcolor = "black", inherit = FALSE)
Do(extinct, SetNodeStyle, style = "filled", fillcolor = 'white',
   fontcolor = 'white', color = 'white', inherit = FALSE)
```

```
plot(parent_nodiab)
```

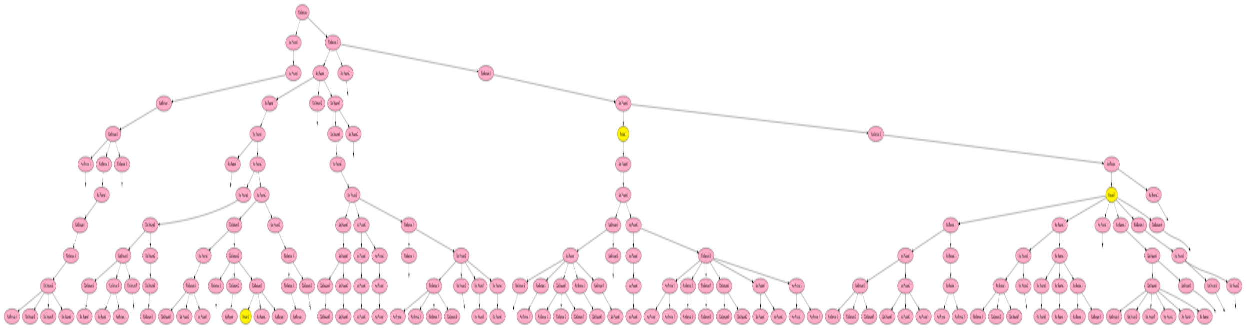

Figure 2: Type 1 Diabetes not initially present

```
length(present)/(length(present)+length(notpresent))
```

```
## [1] 0.0084985
```

```
length(notpresent)/(length(present)+length(notpresent))
```

```
## [1] 0.9915015
```

Once again, the simulation is run for 500 iterations. In this case where Type 1 Diabetes is *not* present at generation 0, Type 1 Diabetes is present, on average, in 0.44743% of individuals, and is not present in 99.55257% of individuals.

```
pres2 <- c()
notpres2 <- c()
for(i in 1:500) {
  parent_nodiab <- Node$new("Not Present")
  add.children(parent_nodiab, diab = 0)
  node <- parent_nodiab$children[[1]]
  if(node$name == " "){
    node <- parent_nodiab
  }

  while (node$name != parent_nodiab$name){
    if (node$level>depth || node$name == " "){
      node <- node$parent
      while(!(node$isLeaf) & node$name != parent_nodiab$name){
        temp <- node$parent
        if(length(node$siblings) > 0) {
          for(i in 1:length(node$siblings)) {
            node.sib <- node$siblings[[i]]
            if(node.sib$isLeaf) {
              temp <- node.sib
            }
          }
        }
        node <- temp
      }
    }
    if (node$level<=depth & node$isLeaf & node$name != parent_nodiab$name){
      if(node$name %in% c("Not Present","Not Present 1", "Not Present 2",
                          "Not Present 3", "Not Present 4", "Not Present 5",
                          "Not Present 6", "Not Present 7", "Not Present 8")) {
        add.children(node, diab = 0)
      } else {
        add.children(node, diab = 1)
      }
      if (node$leafCount > 0){
        node <- node$children[[1]]
      }
    }
  }

  present <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                                        c("Present", "Present 1", "Present 2",
                                          "Present 3","Present 4","Present 5",
                                          "Present 6", "Present 7", "Present 8"))
  notpresent <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                            c("Not Present", "Not Present 1", "Not Present 2",
                              "Not Present 3", "Not Present 4", "Not Present 5",
                              "Not Present 6", "Not Present 7", "Not Present 8"))
  extinct <- Traverse(parent_nodiab, filterFun = function(x) x$name %in% c(" "))

  pres2 <- c(pres2, length(present)/(length(present)+length(notpresent)))
  notpres2 <- c(notpres2, length(notpresent)/(length(present)+length(notpresent)))
}
```

```
mean(pres2)
```

```
## [1] 0.0044743
```

```
mean(notpres2)
```

```
## [1] 0.9955257
```

Discussion

Recall that from the Multitype Branching Processes with Type 1 Diabetes being present (and not present) at generation 0 were that, on average, Type 1 Diabetes was present only 0.86021% of the time when initially present, and only 0.44743% of the time when not initially present. These percentages seem relatively close together compared to the increased probabilities if the condition is present in a direct family member.

Let us try the same simulation with an increased maximum depth. With a maximum depth of 20, we will run the simulation once again 500 times for each case (where Type 1 Diabetes is initially present, and where it is *not* initially present). Now, when the Multitype Branching Processes are run, the average percentage of individuals with Type 1 Diabetes through 20 generations is 0.6169083% when Type 1 Diabetes is initially present, and 0.5849457% when it is *not* initially present. Observe that this is much closer to the national average of 0.55%.

```r
depth = 20

pres3 <- c()
notpres3 <- c()
for(i in  1:500) {
  parent_diab <- Node$new("Present")
  add.children(parent_diab, diab = 1)
  node <- parent_diab$children[[1]]
  if(node$name == " "){
    node <- parent_diab
  }

  while (node$name != parent_diab$name){
    if (node$level>depth || node$name == " "){
      node <- node$parent
      while(!(node$isLeaf) & node$name != parent_diab$name){
        temp <- node$parent
        if(length(node$siblings) > 0) {
          for(i in 1:length(node$siblings)) {
            node.sib <- node$siblings[[i]]
            if(node.sib$isLeaf) {
              temp <- node.sib
            }
          }
        }
        node <- temp
      }
    }
    if (node$level<=depth & node$isLeaf & node$name != parent_diab$name){
      if(node$name %in% c("Present","Present 1", "Present 2",
                          "Present 3", "Present 4", "Present 5",
                          "Present 6", "Present 7", "Present 8")) {
        add.children(node, diab = 1)
      } else {
        add.children(node, diab = 0)
      }
      if (node$leafCount > 0){
        node <- node$children[[1]]
      }
    }
  }

  present <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                                        c("Present", "Present 1", "Present 2",
                                          "Present 3","Present 4","Present 5",
                                          "Present 6", "Present 7", "Present 8"))
  notpresent <- Traverse(parent_diab, filterFun = function(x) x$name %in%
                            c("Not Present", "Not Present 1", "Not Present 2",
                              "Not Present 3", "Not Present 4", "Not Present 5",
                              "Not Present 6", "Not Present 7", "Not Present 8"))
  extinct <- Traverse(parent_diab, filterFun = function(x) x$name %in% c(" "))

  pres3 <- c(pres3, length(present)/(length(present)+length(notpresent)))
  notpres3 <- c(notpres3, length(notpresent)/(length(present)+length(notpresent)))
}
mean(pres3)
```

```
## [1] 0.006169083
```

```
pres4 <- c()
notpres4 <- c()
for(i in 1:500) {
  parent_nodiab <- Node$new("Not Present")
  add.children(parent_nodiab, diab = 0)
  node <- parent_nodiab$children[[1]]
  if(node$name == " "){
    node <- parent_nodiab
  }

  while (node$name != parent_nodiab$name){
    if (node$level>depth || node$name == " "){
      node <- node$parent
      while(!(node$isLeaf) & node$name != parent_nodiab$name){
        temp <- node$parent
        if(length(node$siblings) > 0) {
          for(i in 1:length(node$siblings)) {
            node.sib <- node$siblings[[i]]
            if(node.sib$isLeaf) {
              temp <- node.sib
            }
          }
        }
        node <- temp
      }
    }
    if (node$level<=depth & node$isLeaf & node$name != parent_nodiab$name){
      if(node$name %in% c("Not Present","Not Present 1", "Not Present 2",
                          "Not Present 3", "Not Present 4", "Not Present 5",
                          "Not Present 6", "Not Present 7", "Not Present 8")) {
        add.children(node, diab = 0)
      } else {
        add.children(node, diab = 1)
      }
      if (node$leafCount > 0){
        node <- node$children[[1]]
      }
    }
  }

  present <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                                        c("Present", "Present 1", "Present 2",
                                          "Present 3","Present 4","Present 5",
                                          "Present 6", "Present 7", "Present 8"))
  notpresent <- Traverse(parent_nodiab, filterFun = function(x) x$name %in%
                            c("Not Present", "Not Present 1", "Not Present 2",
                              "Not Present 3", "Not Present 4", "Not Present 5",
                              "Not Present 6", "Not Present 7", "Not Present 8"))
  extinct <- Traverse(parent_nodiab, filterFun = function(x) x$name %in% c(" "))

  pres4 <- c(pres4, length(present)/(length(present)+length(notpresent)))
  notpres4 <- c(notpres4, length(notpresent)/(length(present)+length(notpresent)))
}

mean(pres4)
```

```
## [1] 0.005849457
```

Therefore, it is reasonable to assume that as the depth of the Multitype Branching Process increases, whether or not Type 1 Diabetes is initially present at generation 0 has no significant influence on the spread of Type 1 Diabetes in a family tree. In either case, it seems that the percentage of individuals in the family tree that have Type 1 Diabetes converges to the national average. While there are some real-world examples of families that pass Type 1 Diabetes through multiple generations with many family members having the condition, this simulation shows that this is not the norm. Throughout the trees shown above, there does not seem to be a significant spread of Type 1 Diabetes through a family tree. Even if Type 1 Diabetes does show up in consecutive generations from parent to child (as seen in Figure 1), this does not seem to spread to further generations. It seems that even if Type 1 Diabetes shows up in a family, the likelihood of it being spread through the family line is very low.

References

“Genetics of Diabetes.” *America Diabetes Association*, https://www.diabetes.org/diabetes/genetics-diabetes.

Moore, William. “Is Diabetes Genetic? The Hereditary Risk of Type 1 Diabetes.” *WebMD*, WebMD, https://www.webmd.com/diabetes/diabetes-type-1-genetics.

“Prevalence of Diagnosed Diabetes in Adults by Diabetes Type - United States, 2016.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 29 Mar. 2018, https://www.cdc.gov/mmwr/volumes/67/wr/mm6712a2.htm#.

“Type 1 Diabetes.” *National Institute of Diabetes and Digestive and Kidney Diseases*, U.S. Department of Health and Human Services, https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes/type-1-diabetes#.

“What Is Type 1 Diabetes?” Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 11 Mar. 2022, https://www.cdc.gov/diabetes/basics/what-is-type-1-diabetes.html.

Yeo, Dominic. “Multitype Branching Processes.” *Eventually Almost Everywhere*, 26 Feb. 2014, https://eventuallyalmosteverywhere.wordpress.com/2014/02/25/multitype-branching-processes/.