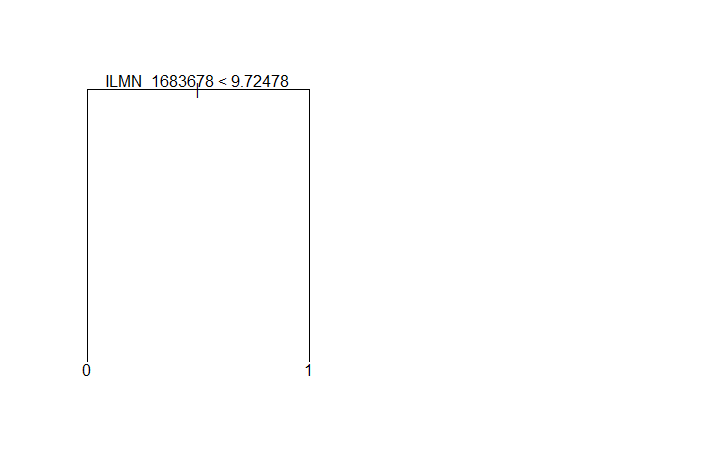
Random Forest results. (to be copied into final paper)

We also utilized Random Forests to analyze the data. This is another process that can be computationally intensive. We focused on large forests (each forest with ten thousand trees) but we only considered the default number of factors for each tree.

Working with more than forty-eight thousand factors creates a unique challenge. This is more than RStudio can handle with the randomForest function without crashing. To address this problem we had to choose a form of factor reduction. We chose to use importance from randomForest to take a subset of the “more important” factors from subsets of one thousand (or less) factors.

Once the “more important” factors have been identified, randomForest creates trees amongst these factors to identify the “most important” factors across the entire dataset.

We were surprised to note very limited overlap from Random Forests to other methods used by ourselves and the paper. It’s not surprising that ILMN\_1683678 (SPATS2L) always makes the top slot (it has 100 percent accuracy in predicting the end result by itself, see fig ###), but other factors not predicted elsewhere are also ranked very highly in terms of importance for Random Forests (see fig ###).

Creating individual trees from these “most important” factors has fairly good accuracy. We only see 100% accuracy in the case of ILMN\_1683678 (see fig ###), but other trees use two factors for 96% accuracy (for example, see fig ### and ####).

