

# visualize lineage barcodes

Andy Sposato

**The purpose of this notebook is to visualize lineage barcodes individually on UMAP and in a line graph.**

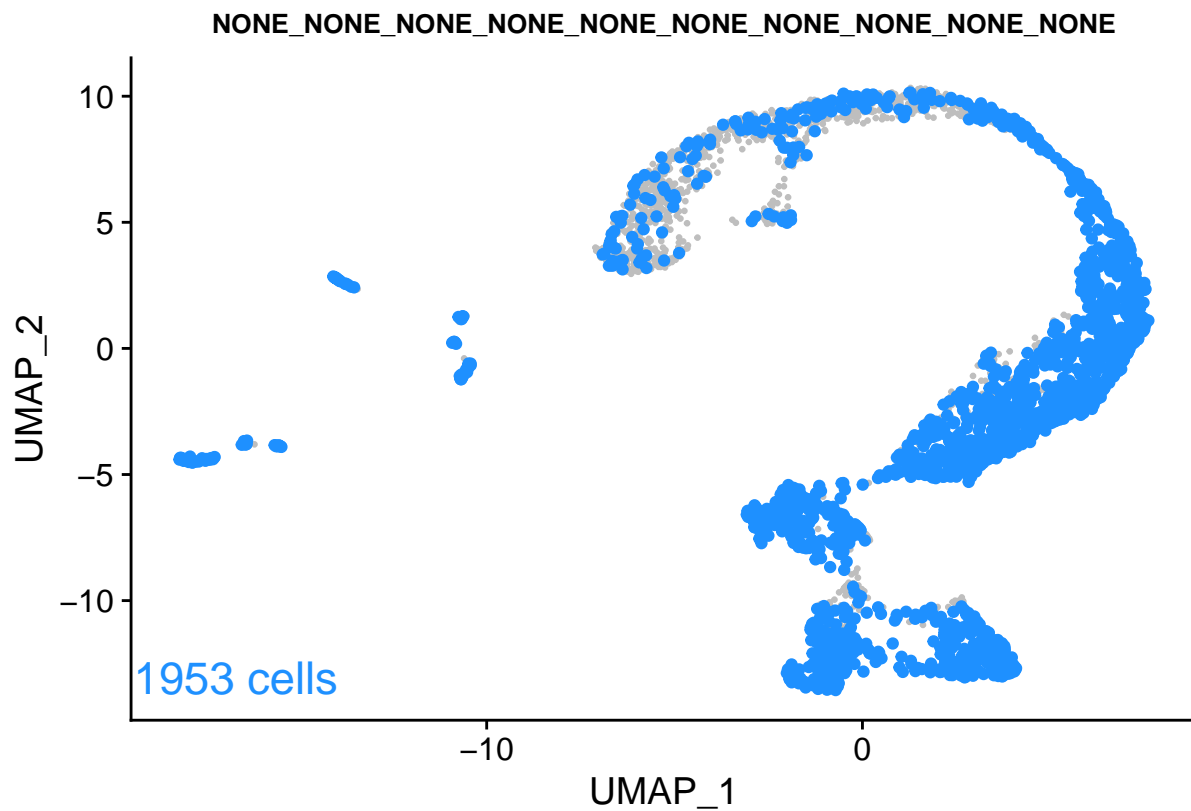
For individual barcodes on UMAP, we can generate UMAP images where all cells with the same lineage barcode are highlighted.

For the line graph, we can plot normalized barcode abundance across differentiation within the object.

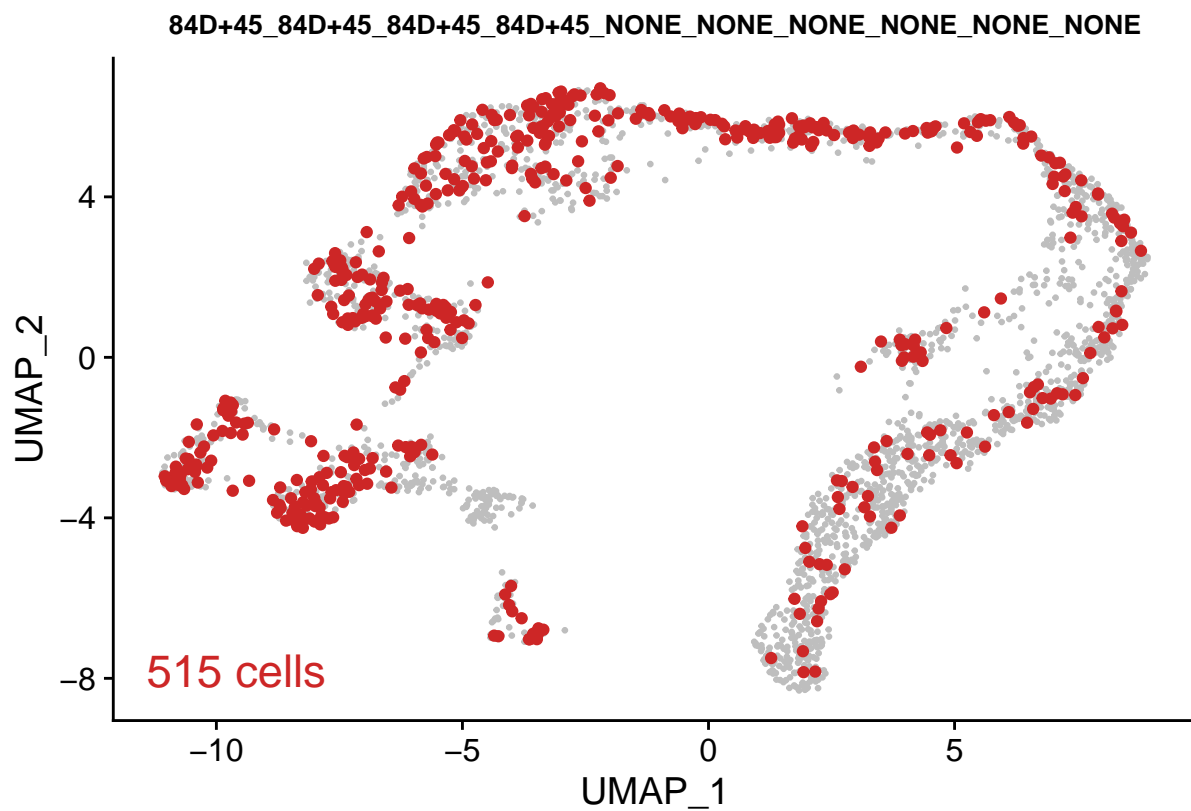
## Plotting individual barcodes on UMAP

```
edited <- readRDS("data/objects/edited_w_diffscores_lineage.rds")
unedited <- readRDS("data/objects/unedited_w_diffscores_lineage.rds")

# this is function is from bin/plot_barcode_on_UMAP.R
plot_barcode("NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE", unedited,
             "dodgerblue")
```



```
plot_barcode("84D+45_84D+45_84D+45_84D+45_NONE_NONE_NONE_NONE_NONE_NONE", edited,
             "firebrick3")
```



Generating a line graph for all the barcodes in your object

```
edited <- readRDS("data/objects/edited_w_diffscores_lineage.rds")
unedited <- readRDS("data/objects/unedited_w_diffscores_lineage.rds")
```

edited data

```
meta <- edited@meta.data
meta$cell.id <- row.names(meta)
meta <- as.data.frame(cbind(meta$cell.id, meta$diff.score, meta$barcode,
                             meta$site1, meta$site2, meta$site3, meta$site4,
                             meta$site5, meta$site6, meta$site7, meta$site8,
                             meta$site9, meta$site10))
write.table(meta, "data/tables/edited_diffscores.tsv",
            sep = "\t", col.names = F, row.names = F, quote = F)
```

```
data_folder = "./data/tables/"
```

```
cells_ranges_dict= {}
file = open(data_folder + 'edited_diffscores.tsv', "r")
# for line in edited_diffscores
for line in file.readlines():
    # split line by tab
    line = line.split("\t")
    # cell id is the first index
    cellid = line[0].rstrip("\t")
    # followed by differentiation score
    diff = float(line[1].rstrip("\t"))
    # the full barcode
    barcode = line[2].rstrip("\t")
    # and the barcode broken up by site
    site1 = line[3].rstrip("\t")
    site2 = line[4].rstrip("\t")
    site3 = line[5].rstrip("\t")
    site4 = line[6].rstrip("\t")
    site5 = line[7].rstrip("\t")
    site6 = line[8].rstrip("\t")
    site7 = line[9].rstrip("\t")
    site8 = line[10].rstrip("\t")
    site9 = line[11].rstrip("\t")
    site10 = line[12].rstrip("\n\t")
    # group according to differentiation range for cells_ranges_dict
    if diff <= 0.1:
        cells_ranges_dict[cellid] = ['0-0.1', barcode, site1, site2, site3, site4,
                                     site5, site6, site7, site8, site9, site10]
    elif diff > 0.1 and diff <= 0.2:
        cells_ranges_dict[cellid] = ['0.1-0.2', barcode, site1, site2, site3, site4,
                                     site5, site6, site7, site8, site9, site10]
```

```

elif diff > 0.2 and diff <= 0.3:
    cells_ranges_dict[cellid] = ['0.2-0.3', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.3 and diff <= 0.4:
    cells_ranges_dict[cellid] = ['0.3-0.4', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.4 and diff <= 0.5:
    cells_ranges_dict[cellid] = ['0.4-0.5', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.5 and diff <= 0.6:
    cells_ranges_dict[cellid] = ['0.5-0.6', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.6 and diff <= 0.7:
    cells_ranges_dict[cellid] = ['0.6-0.7', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.7 and diff <= 0.8:
    cells_ranges_dict[cellid] = ['0.7-0.8', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.8 and diff <= 0.9:
    cells_ranges_dict[cellid] = ['0.8-0.9', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.9:
    cells_ranges_dict[cellid] = ['0.9-1.0', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
file.close()

# add the differentiation range as a column to the metadata of the object
edited_diffranges <- read.table('data/tables/edited_diffranges.tsv',
                               sep = '\t', header = T)
edited <- AddMetaData(edited, as.list(subset(edited_diffranges, select = c('diff.range'))),
                     col.name = 'diff.range')

# get the list of unique edited barcodes
unique_edited_barcodes <- as.data.frame(unique(edited@meta.data$barcode))
write.table(unique_edited_barcodes, "data/tables/unique_edited_barcodes.tsv",
           sep = "\t", col.names = F, row.names = F, quote = F)

# collect unique barcodes from edited testes
unique_edited_barcodes = open(data_folder + "unique_edited_barcodes.tsv", "r")
barcodes = []
# read in the list of unique barcodes and save to barcodes
for barcode in unique_edited_barcodes:
    barcode_string = barcode.strip("\n")
    if barcode_string != "NA":
        barcodes.append(barcode_string)
unique_edited_barcodes.close()

barcodes_dict = {}
# create a differentiation range dictionary where the value is an empty list for each range
# do that for each barcode (nested dictionaries)
for barcode in barcodes:
    barcodes_dict[str(barcode)] = {'0-0.1': [], '0.1-0.2': [], '0.2-0.3': [],
    '0.3-0.4': [], '0.4-0.5': [], '0.5-0.6': [], '0.6-0.7': [], '0.7-0.8': [],
    '0.8-0.9': [], '0.9-1': []}

```

```

# example barcode dictionary
barcodes_dict['1I+45+C_15D+66_NONE_9D+123_NONE_NONE_NONE_NONE_NONE_NONE']

## {'0-0.1': [], '0.1-0.2': [], '0.2-0.3': [], '0.3-0.4': [], '0.4-0.5': [], '0.5-0.6': [], '0.6-0.7':

# create barcode dictionaries
file = open(data_folder + "edited_diffranges.tsv", "r")
file.readline()

## 'cell.id\tdiff.range\tbarcode\tsite1\tsite2\tsite3\tsite4\tsite5\tsite6\tsite7\tsite8\tsite9\tsite10'
# for line in edited_diffranges.tsv
for line in file.readlines():
    # split line by tab character
    line = line.split("\t")
    # first index is cell id
    cell = line[0].strip("\t")
    # second index is differentiation range
    diff_range = line[1].strip("\t")
    # third index is the barcode
    barcode = line[2].rstrip("\t")
    # if the barcode already exists in barcodes_dict (they should)
    if barcode in barcodes_dict.keys():
        # if the diff_range equals a key in the barcode dictionary
        if diff_range in barcodes_dict[barcode].keys():
            # assign the cell to that barcode dictionary's range dictionary
            barcodes_dict[barcode][diff_range].append(cell)
file.close()

# example
barcodes_dict['1I+45+C_15D+66_NONE_9D+123_NONE_NONE_NONE_NONE_NONE_NONE']

## {'0-0.1': ['AAACGAAAGACTTCCA-1', 'AATGGCTCACCTCAGG-1', 'TAGTGCATCGAACCTA-1', 'TGGTAGTGTGAGGTGA-1', '

# get a count for how many cells are in the range and store this as the last value
for ranges_dictionary in barcodes_dict.values():
    for key, value in ranges_dictionary.items():
        cell_count = len(value)
        ranges_dictionary[key].append(str(cell_count))

# example
barcodes_dict['1I+45+C_15D+66_NONE_9D+123_NONE_NONE_NONE_NONE_NONE_NONE']

## {'0-0.1': ['AAACGAAAGACTTCCA-1', 'AATGGCTCACCTCAGG-1', 'TAGTGCATCGAACCTA-1', 'TGGTAGTGTGAGGTGA-1', '

# write a table with cell counts included
output = open(data_folder + "edited_diffranges_cellcounts.tsv", 'w')
output.write('barcode\t'+ '0-0.1'+ '\t'+ '0.1-0.2'+ '\t'+ '0.2-0.3'+ '\t'+ '0.3-0.4'+ '\t'+
'0.4-0.5'+ '\t'+ '0.5-0.6'+ '\t'+ '0.6-0.7'+ '\t'+ '0.7-0.8'+ '\t'+ '0.8-0.9'+ '\t'+ '0.9-1'+ '\n')

## 84

# for barcode in barcodes_dict
for barcode in barcodes_dict.keys():
    # write the cell count for each diff range dictionary to a tsv
    output.write(barcode+
'\t'+barcodes_dict[barcode]['0-0.1'][-1]+
'\t'+barcodes_dict[barcode]['0.1-0.2'][-1]+
'\t'+barcodes_dict[barcode]['0.2-0.3'][-1]+

```

```
'\t'+barcodes_dict[barcode]['0.3-0.4'][-1]+
'\t'+barcodes_dict[barcode]['0.4-0.5'][-1]+
'\t'+barcodes_dict[barcode]['0.5-0.6'][-1]+
'\t'+barcodes_dict[barcode]['0.6-0.7'][-1]+
'\t'+barcodes_dict[barcode]['0.7-0.8'][-1]+
'\t'+barcodes_dict[barcode]['0.8-0.9'][-1]+
'\t'+barcodes_dict[barcode]['0.9-1'][-1]+'\\n')
```

```
## 79
## 88
## 111
## 79
## 80
## 150
## 114
## 114
## 131
## 86
## 108
## 109
## 76
## 82
```

```
output.close()
```

```
# clonality should be a dataframe where for each barcode, there is a list of cell counts
# for each differentiation range where that barcode is found
clonality <- read.table("data/tables/edited_diffranges_cellcounts.tsv",
                        sep = "\\t", header = T)
# here we normalize the cell counts to a percent value
# the normalization accounts for the fact that not every cell
# within a differentiation range will have a high confidence lineage barcode
clonality$X0.0.1 <- as.vector(
  sapply(clonality$X0.0.1,normalize,
         length(subset(edited@meta.data,
                        barcode.count != 0 & diff.range == '0-0.1')$barcode)))
clonality$X0.1.0.2 <- as.vector(
  sapply(clonality$X0.1.0.2,normalize,
         length(subset(edited@meta.data,
                        barcode.count != 0 & diff.range == '0.1-0.2')$barcode)))
clonality$X0.2.0.3 <- as.vector(
  sapply(clonality$X0.2.0.3,normalize,
         length(subset(edited@meta.data,
                        barcode.count != 0 & diff.range == '0.2-0.3')$barcode)))
clonality$X0.3.0.4 <- as.vector(
  sapply(clonality$X0.3.0.4,normalize,
         length(subset(edited@meta.data,
                        barcode.count != 0 & diff.range == '0.3-0.4')$barcode)))
clonality$X0.4.0.5 <- as.vector(
  sapply(clonality$X0.4.0.5,normalize,
         length(subset(edited@meta.data,
                        barcode.count != 0 & diff.range == '0.4-0.5')$barcode)))
clonality$X0.5.0.6 <- as.vector(
  sapply(clonality$X0.5.0.6,normalize,
```

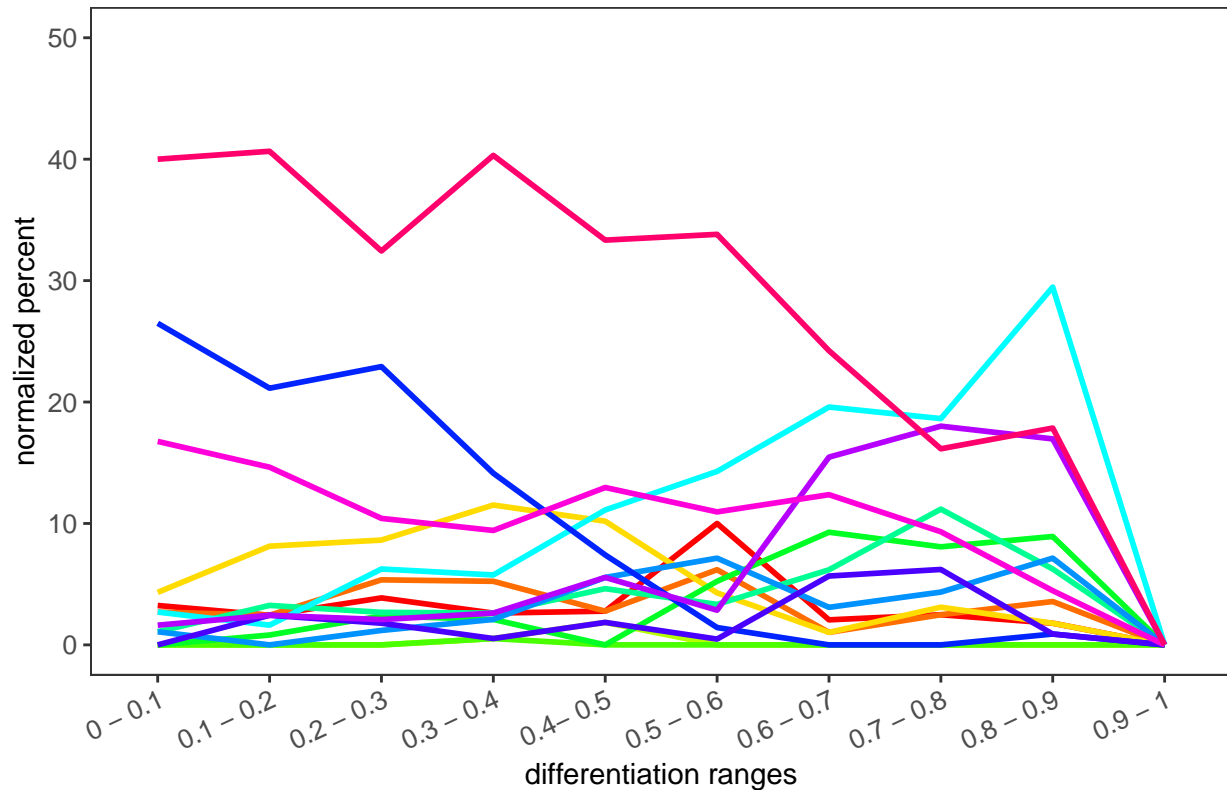
```

        length(subset(edited@meta.data,
                      barcode.count != 0 & diff.range == '0.5-0.6')$barcode)))
clonality$X0.6.0.7 <- as.vector(
  sapply(clonality$X0.6.0.7, normalize,
        length(subset(edited@meta.data,
                      barcode.count != 0 & diff.range == '0.6-0.7')$barcode)))
clonality$X0.7.0.8 <- as.vector(
  sapply(clonality$X0.7.0.8, normalize,
        length(subset(edited@meta.data,
                      barcode.count != 0 & diff.range == '0.7-0.8')$barcode)))
clonality$X0.8.0.9 <- as.vector(
  sapply(clonality$X0.8.0.9, normalize,
        length(subset(edited@meta.data,
                      barcode.count != 0 & diff.range == '0.8-0.9')$barcode)))
clonality$X0.9.1 <- as.vector(
  sapply(clonality$X0.9.1, normalize,
        length(subset(edited@meta.data,
                      barcode.count != 0 & diff.range == '0.9-1.0')$barcode)))
# convert the names to something readable
names(clonality)[names(clonality) == "X0.0.1"] <- "0 - 0.1"
names(clonality)[names(clonality) == "X0.1.0.2"] <- "0.1 - 0.2"
names(clonality)[names(clonality) == "X0.2.0.3"] <- "0.2 - 0.3"
names(clonality)[names(clonality) == "X0.3.0.4"] <- "0.3 - 0.4"
names(clonality)[names(clonality) == "X0.4.0.5"] <- "0.4 - 0.5"
names(clonality)[names(clonality) == "X0.5.0.6"] <- "0.5 - 0.6"
names(clonality)[names(clonality) == "X0.6.0.7"] <- "0.6 - 0.7"
names(clonality)[names(clonality) == "X0.7.0.8"] <- "0.7 - 0.8"
names(clonality)[names(clonality) == "X0.8.0.9"] <- "0.8 - 0.9"
names(clonality)[names(clonality) == "X0.9.1"] <- "0.9 - 1"

# plot the line graph
# use melt to format the table nicely for ggplot
clonality_df <- melt(clonality, id.vars="barcode", value.name="value",
                    variable.name= c("state"))
ggplot(data=clonality_df, aes(x=state, y=value, group = barcode, colour = barcode)) +
  #geom_point() +
  geom_line(linewidth = 1) +
  theme_bw() +
  theme(axis.text.y = element_text(size = 10)) +
  theme(axis.text.x = element_text(size = 10, angle = 25, hjust = 1)) +
  theme(plot.title = element_text(size = 15)) +
  ggtitle("lineage barcode abundance across differentiation") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("differentiation ranges") +
  ylab("normalized percent") + ylim(0,50) +
  scale_color_manual(values=rainbow(14)) + theme(legend.position = "none")

```

## lineage barcode abundance across differentiation



unedited data

This will be less exciting since all barcodes found should be the unedited barcode. But it serves as a good check that it's working right. We should see the unedited barcode at 100% abundance across all differentiation ranges except 0.9-1 because we don't actually have barcoded cells in that range.

```
meta <- unedited@meta.data
meta$cell.id <- row.names(meta)
meta <- as.data.frame(cbind(meta$cell.id, meta$diff.score, meta$barcode, meta$site1,
                             meta$site2, meta$site3, meta$site4, meta$site5, meta$site6,
                             meta$site7, meta$site8, meta$site9, meta$site10))
write.table(meta, "data/tables/unedited_diffscores.tsv", sep = "\t", col.names = F,
            row.names = F, quote = F)
```

```
data_folder = "./data/tables/"
```

```
cells_ranges_dict= {}
file = open(data_folder + 'unedited_diffscores.tsv', "r")
for line in file.readlines():
    line = line.split("\t")
    cellid = line[0].rstrip("\t")
    diff = float(line[1].rstrip("\t"))
    barcode = line[2].rstrip("\t")
```



```

site1 = line[3].rstrip("\t")
site2 = line[4].rstrip("\t")
site3 = line[5].rstrip("\t")
site4 = line[6].rstrip("\t")
site5 = line[7].rstrip("\t")
site6 = line[8].rstrip("\t")
site7 = line[9].rstrip("\t")
site8 = line[10].rstrip("\t")
site9 = line[11].rstrip("\t")
site10 = line[12].rstrip("\n\t")
if diff <= 0.1:
    cells_ranges_dict[cellid] = ['0-0.1', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.1 and diff <= 0.2:
    cells_ranges_dict[cellid] = ['0.1-0.2', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.2 and diff <= 0.3:
    cells_ranges_dict[cellid] = ['0.2-0.3', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.3 and diff <= 0.4:
    cells_ranges_dict[cellid] = ['0.3-0.4', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.4 and diff <= 0.5:
    cells_ranges_dict[cellid] = ['0.4-0.5', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.5 and diff <= 0.6:
    cells_ranges_dict[cellid] = ['0.5-0.6', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.6 and diff <= 0.7:
    cells_ranges_dict[cellid] = ['0.6-0.7', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.7 and diff <= 0.8:
    cells_ranges_dict[cellid] = ['0.7-0.8', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.8 and diff <= 0.9:
    cells_ranges_dict[cellid] = ['0.8-0.9', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
elif diff > 0.9:
    cells_ranges_dict[cellid] = ['0.9-1.0', barcode, site1, site2, site3, site4,
    site5, site6, site7, site8, site9, site10]
file.close()

unedited_diffranges <- read.table('data/tables/unedited_diffranges.tsv',
                                sep = '\t', header = T)
unedited <- AddMetaData(unedited,
                        as.list(subset(unedited_diffranges, select = c('diff.range'))),
                        col.name = 'diff.range')

unique_unedited_barcodes <- as.data.frame(unique(unedited@meta.data$barcode))
write.table(unique_unedited_barcodes, "data/tables/unique_unedited_barcodes.tsv",
            sep = "\t", col.names = F, row.names = F, quote = F)

unique_unedited_barcodes = open(data_folder + "unique_unedited_barcodes.tsv", "r")
barcodes = []

```

```

for barcode in unique_unedited_barcodes:
    barcode_string = barcode.strip("\n")
    if barcode_string != "NA":
        barcodes.append(barcode_string)
unique_unedited_barcodes.close()

barcodes_dict = {}
for barcode in barcodes:
    barcodes_dict[str(barcode)] = {'0-0.1': [], '0.1-0.2': [], '0.2-0.3': [],
    '0.3-0.4': [], '0.4-0.5': [], '0.5-0.6': [], '0.6-0.7': [], '0.7-0.8': [],
    '0.8-0.9': [], '0.9-1': []}

barcodes_dict['NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE']

## {'0-0.1': [], '0.1-0.2': [], '0.2-0.3': [], '0.3-0.4': [], '0.4-0.5': [], '0.5-0.6': [], '0.6-0.7':
file = open(data_folder + "unedited_diffranges.tsv", "r")
file.readline()

## 'cell.id\tdiff.range\tbarcode\tsite1\tsite2\tsite3\tsite4\tsite5\tsite6\tsite7\tsite8\tsite9\tsite10'
for line in file.readlines():
    line = line.split("\t")
    cell = line[0].strip("\t")
    diff_range = line[1].strip("\t")
    barcode = line[2].rstrip("\t")
    if barcode in barcodes_dict.keys():
        if diff_range in barcodes_dict[barcode].keys():
            barcodes_dict[barcode][diff_range].append(cell)
file.close()

# checking this worked right
print(barcodes_dict['NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE_NONE']['0.8-0.9'])

## ['AACCTGATCACAACCA-1_1', 'ACGTAGTCAACATACC-1_1', 'AGACTCAAGTCCTGTA-1_1', 'AGAGCCCCACGAGGTA-1_1', 'AG
for ranges_dictionary in barcodes_dict.values():
    for key, value in ranges_dictionary.items():
        cell_count = len(value)
        ranges_dictionary[key].append(str(cell_count))

output = open(data_folder + 'unedited_diffranges_cellcounts.tsv', 'w')
output.write('barcode\t'+ '0-0.1'+ '\t'+ '0.1-0.2'+ '\t'+ '0.2-0.3'+ '\t'+ '0.3-0.4'+ '\t'+ '0.4-0.5'+ '\t'+
    '0.5-0.6'+ '\t'+ '0.6-0.7'+ '\t'+ '0.7-0.8'+ '\t'+ '0.8-0.9'+ '\t'+ '0.9-1'+ '\n')

## 84
for barcode in barcodes_dict.keys():
    output.write(barcode+
        '\t'+barcodes_dict[barcode]['0-0.1'][-1]+
        '\t'+barcodes_dict[barcode]['0.1-0.2'][-1]+
        '\t'+barcodes_dict[barcode]['0.2-0.3'][-1]+
        '\t'+barcodes_dict[barcode]['0.3-0.4'][-1]+
        '\t'+barcodes_dict[barcode]['0.4-0.5'][-1]+
        '\t'+barcodes_dict[barcode]['0.5-0.6'][-1]+
        '\t'+barcodes_dict[barcode]['0.6-0.7'][-1]+
        '\t'+barcodes_dict[barcode]['0.7-0.8'][-1]+
        '\t'+barcodes_dict[barcode]['0.8-0.9'][-1]+

```

```
'\t'+barcodes_dict[barcode]['0.9-1'][-1]+'\\n')
```

```
## 86
```

```
output.close()
```

```
clonality <- read.table("data/tables/unedited_diffranges_cellcounts.tsv",
  sep = "\\t", header = T)
clonality$X0.0.1 <- as.vector(sapply(clonality$X0.0.1, normalize, length(subset(unedited@meta.data, barcode == "X0.0.1"))))
clonality$X0.1.0.2 <- as.vector(sapply(clonality$X0.1.0.2, normalize, length(subset(unedited@meta.data, barcode == "X0.1.0.2"))))
clonality$X0.2.0.3 <- as.vector(sapply(clonality$X0.2.0.3, normalize, length(subset(unedited@meta.data, barcode == "X0.2.0.3"))))
clonality$X0.3.0.4 <- as.vector(sapply(clonality$X0.3.0.4, normalize, length(subset(unedited@meta.data, barcode == "X0.3.0.4"))))
clonality$X0.4.0.5 <- as.vector(sapply(clonality$X0.4.0.5, normalize, length(subset(unedited@meta.data, barcode == "X0.4.0.5"))))
clonality$X0.5.0.6 <- as.vector(sapply(clonality$X0.5.0.6, normalize, length(subset(unedited@meta.data, barcode == "X0.5.0.6"))))
clonality$X0.6.0.7 <- as.vector(sapply(clonality$X0.6.0.7, normalize, length(subset(unedited@meta.data, barcode == "X0.6.0.7"))))
clonality$X0.7.0.8 <- as.vector(sapply(clonality$X0.7.0.8, normalize, length(subset(unedited@meta.data, barcode == "X0.7.0.8"))))
clonality$X0.8.0.9 <- as.vector(sapply(clonality$X0.8.0.9, normalize, length(subset(unedited@meta.data, barcode == "X0.8.0.9"))))
clonality$X0.9.1 <- as.vector(sapply(clonality$X0.9.1, normalize, length(subset(unedited@meta.data, barcode == "X0.9.1"))))
names(clonality)[names(clonality) == "X0.0.1"] <- "0 - 0.1"
names(clonality)[names(clonality) == "X0.1.0.2"] <- "0.1 - 0.2"
names(clonality)[names(clonality) == "X0.2.0.3"] <- "0.2 - 0.3"
names(clonality)[names(clonality) == "X0.3.0.4"] <- "0.3 - 0.4"
names(clonality)[names(clonality) == "X0.4.0.5"] <- "0.4 - 0.5"
names(clonality)[names(clonality) == "X0.5.0.6"] <- "0.5 - 0.6"
names(clonality)[names(clonality) == "X0.6.0.7"] <- "0.6 - 0.7"
names(clonality)[names(clonality) == "X0.7.0.8"] <- "0.7 - 0.8"
names(clonality)[names(clonality) == "X0.8.0.9"] <- "0.8 - 0.9"
names(clonality)[names(clonality) == "X0.9.1"] <- "0.9 - 1"

clonality_df <- melt(clonality, id.vars="barcode", value.name="value", variable.name= c("state"))
ggplot(data=clonality_df, aes(x=state, y=value, group = barcode, colour = barcode)) +
  #geom_point() +
  geom_line(linewidth = 1) +
  theme_bw() +
  theme(axis.text.y = element_text(size = 10)) +
  theme(axis.text.x = element_text(size = 10, angle = 25, hjust = 1)) +
  theme(plot.title = element_text(size = 20)) +
  ggtitle("lineage barcode abundance across differentiation") +
  theme(plot.title = element_text(hjust = 0.5, size = 12)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("differentiation ranges") +
  ylab("normalized percent") + ylim(0,100) +
  scale_color_manual(values=rainbow(1)) + theme(legend.position = "none")
```

