Workflows for Visualization

An example with bigger dataset

Here the visualization workflow from Dalziel et al. 2013. Human mobility patterns predict divergent epidemic dynamics among cities. PRSB 280.

```r
rm(list = ls())
load("../data/TorontoCommuting.Rdata")
ls()
```

```
## [1] "A" "d" "N" "x" "y"
```

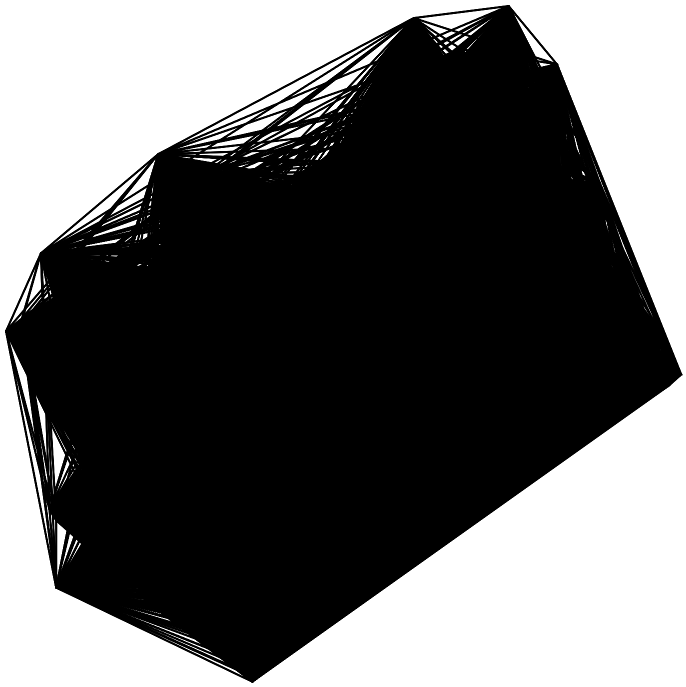
These data consist of commuting flows between 743 census tracts within the city of Toronto, connecting the home and work locations of 3.8 million workers. These are organized in the matrix A, such that element in the ith row and jth column corresponding to the number of people who live in census tract i and work in census tract j. The latitude and longitude of the census tracts are stored in the vectors x and y. The matrix d supplies the pairwise distance between census tracts. N is the population size of each census tract.

Here is a barebones network plot where each edge represents a commuting flow between a pair of locations

```r
# Assign home and work x and y locations to each element of the commmuting matrix
n <- nrow(A)
hx <- matrix(x,n,n)
hy <- matrix(y,n,n)
wx <- t(hx)
wy <- t(hy)


# Empty plot
par(fin = c(4,4))
par(mai = c(0.1,0.1,0.1,0.1))
plot(range(x), range(y),
     type="n",
     xlab="",
     ylab="",
     bty="n",
     xaxt="n",
     yaxt="n")

# Draw line segments for each commuting flow
segments(hx, hy, wx, wy)
```



What a mess. The problem is that there are too many lines and they are too thick to see what is going on. A quick cleanup step is to make each of the lines very thin, and to make them almost transparent. We control line color using the `rgb()` function, which specifies colors using the red, green, blue, alpha system, where alpha is transparency. Let's also plot the most important (ie biggest) flows last, so they don't get obscured by all the tiny ones.

```
# Assign home and work x and y locations to each element of the commmuting matrix
n <- nrow(A)
hx <- matrix(x,n,n)
hy <- matrix(y,n,n)
wx <- t(hx)
wy <- t(hy)


# Specify edge widths and colors
edge_width <- 0.1
edge_color <- rgb(red = 0, green = 0, blue = 0, alpha = 0.1)     # black, nearly transparent


# Sort by flow so thickest segments are plotted last
idx <- order(A)
idx <- idx[idx>0]


# Empty plot
par(fin = c(4,4))
par(mai = c(0.1,0.1,0.1,0.1))
plot(range(x), range(y),
     type="n",
     xlab="",
     ylab="",
     bty="n",
```

```
     xaxt="n",
     yaxt="n")

# Draw line segments for each commuting flow
# segments(hx, hy, wx, wy)

# Draw line segments for each commuting flow
segments(hx[idx], hy[idx], wx[idx], wy[idx],
         lwd = edge_width[idx],
         col = edge_color[idx])
```

And here is the whole workflow.

```
# Plotting parameters
max_edge_width <-  5
min_edge_width <- 0.1
max_alpha <- 1
min_alpha <- 0


# Assign home and work x and y locations to each element of the commmuting matrix
n <- nrow(A)
hx <- matrix(x,n,n)
hy <- matrix(y,n,n)
wx <- t(hx)
wy <- t(hy)


# Standardize commuting flows and distance in A in preparation for mapping to colors
standardized_flow <- 1-exp(-0.0015*A)
standardized_distance <- 1-exp(-0.07*d)
standardized_distance[is.na(standardized_distance)] <- 0


# Specify red, green, blue and transparency channels for the edge colors
red <- standardized_flow
green <- standardized_distance
blue <- 1 - standardized_flow
alpha <- (max_alpha - min_alpha) * standardized_flow
alpha <- alpha - min(alpha) + min_alpha


# Set the edge colors and widths
edge_color <- rgb(red, green, blue, alpha)
edge_width <- standardized_flow*(max_edge_width-min_edge_width)
edge_width <- edge_width + min_edge_width


# Sort by flow so thickest segments are plotted last
idx <- order(A)
idx <- idx[idx>0]
```

```r
# Plot
par(fin = c(4,4))
par(mai = c(0.1,0.1,0.1,0.1))
plot(range(x), range(y),
     type="n",
     xlab="",
     ylab="",
     bty="n",
     xaxt="n",
     yaxt="n")

segments(hx[idx], hy[idx], wx[idx], wy[idx],
         lwd = edge_width[idx],
         col = edge_color[idx])
```