Population Graphs

Rodney J. Dyer
Department of Biology
Virginia Commonwealth University
http://dyerlab.bio.vcu.edu

Synopsis

A population graph is a topological representation of within and among population genetic variance first introduced by Dyer & Nason (2004). It is particularly well suited to characterizing how spatial genetic variation is distributed among sites.

```
> require(gstudio)
> data(araptus_attenuatus)
> baja <- araptus_attenuatus[araptus_attenuatus$Species != "CladeB",]
```

Simple Population Graphs

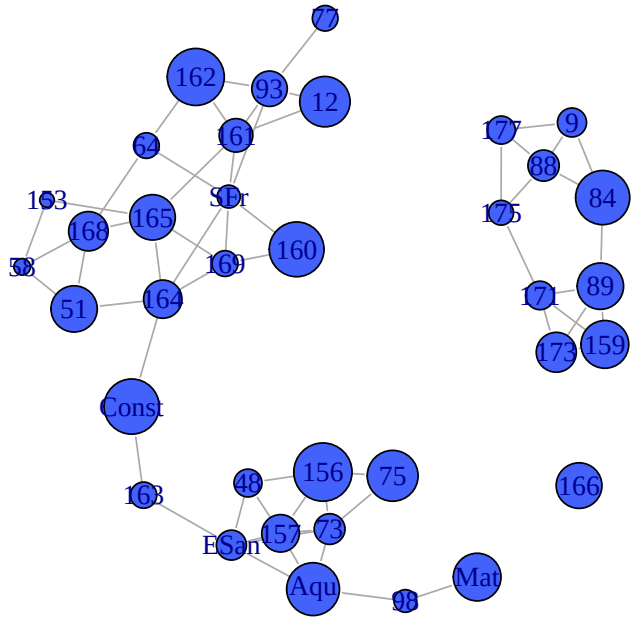
```
> graph <- population.graph(baja,"Pop")
```

```
tranforming data... done
Rotating mv genos and partitioning...  done
Estimating conditional genetic covariance... done
Making graph... done
```

```
> summary(graph)
```

```
Vertices: 36
Edges: 59
Directed: FALSE
No graph attributes.
Vertex attributes: name, size, color.
Edge attributes: weight.
```

```
> l <- layout.fruchterman.reingold(graph)
> plot(graph,layout=l,vertex.label=V(graph)$name)
```



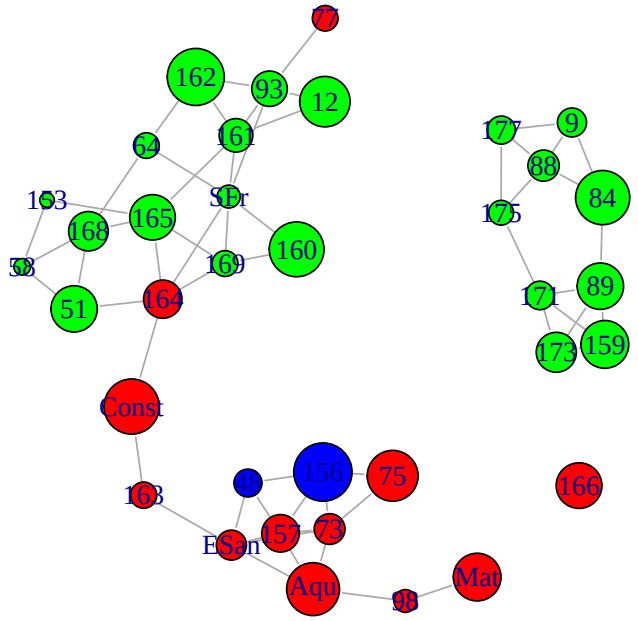
We know that these data are a mixture of two putative species denoted as `CladeA` and `CladeC`.

```
> table(baja$Species)
```

```
CladeA CladeC
    75    252
```

We can color the nodes depending upon the identity of clade representation at the node-level. If there is a mixture of species, you would expect to find that the mixed populations would be topologically intermediate between populations made up of pure samples.

```
> getCladeColor <- function(pop,data) {
+         inds <- data$Species[data$Pop==pop]
+         levels.inds <- levels(as.factor(as.character(inds)))
+         if(length(levels.inds)==2) return("red")
+         else if( levels.inds=="CladeA" ) return("blue")
+         else return("green")
+ }
> colors <- unlist(lapply(V(graph)$name, function(x) getCladeColor(x,baja)))
> plot(graph,layout=l,vertex.label=V(graph)$name,vertex.color=colors)
```



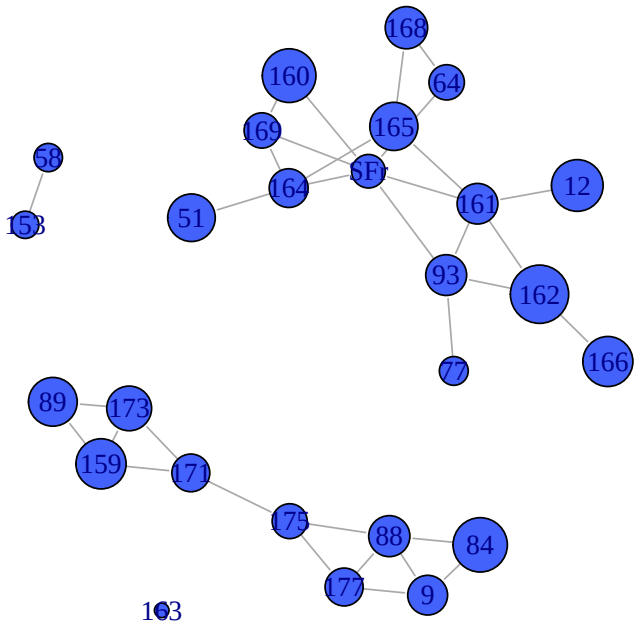
So if we only use the samples from `CladeC` we may be actually analyzing the data in a way that makes sense. Do this by:

1. Use only the `CladeC` individuals
2. Get rid of the populations with say $N < 5$ individuals
3. Make graph and examine the topology

```
> baja.cladeC <- baja[baja$Species=="CladeC",]
> inds.per.pop <- lapply( partition(baja.cladeC,"Pop"), function(x) dim(x)[1] )
> ## Examine inds per pop to figure out which have <5 individuals save in smPops
> smPops <- c("Const","ESan","157","73","Aqu","Mat","98","75")
> baja.cladeC <- baja.cladeC[ !(baja.cladeC$Pop %in% smPops) , ]
> graph.cladeC <- population.graph(baja.cladeC,"Pop")

tranforming data... done
Rotating mv genos and partitioning...  done
Estimating conditional genetic covariance... done
Making graph... done

> summary(graph.cladeC)

Vertices: 26
Edges: 33
```

```
Directed: FALSE
No graph attributes.
Vertex attributes: name, size, color.
Edge attributes: weight.
```

```
> l <- layout.fruchterman.reingold(graph.cladeC)
> plot(graph.cladeC,layout=l,vertex.label=V(graph.cladeC)$name)
```

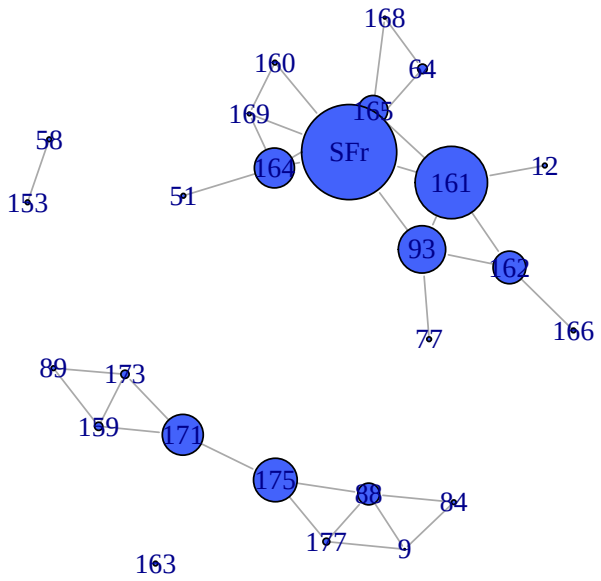


From this plot, you can see even when we only focus on the true `CladeC` individuals, there is still partitioning of genetic covariance!

Node Position

Both node and edge position in the topology can easily be determined using common network analysis tools. The `igraph` package has some as does the most excellent `sna` package. Here is a quick example where the size of the node is depicting the node's betweeness (e.g., the number of shortest paths that go through that node).

```
> pop.betweenness <- betweenness(graph.cladeC,directed=F)
> plot(graph.cladeC,layout=l,vertex.label=V(graph.cladeC)$name,vertex.size=pop.betweenness)
```



Which is rather interesting since betweenness can be used to classify relative population importance. Presently, it is common to use genetic diversity as a surrogate to identify populations of high conservation importance, but betweenness relates to the connectivity of the gene flow topology on the landscape and is not necessarily correlated with genetic diversity.

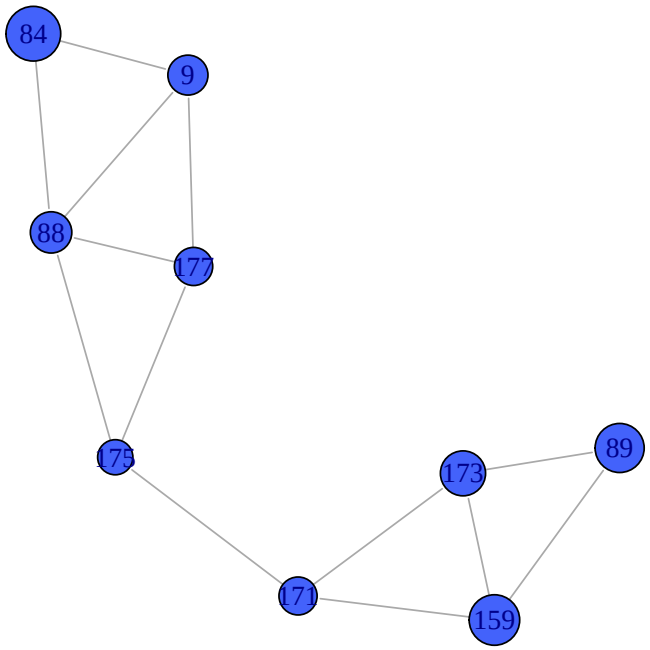
```
> cor.test(V(graph.cladeC)$size,pop.betweenness,method="spearman")

        Spearman's rank correlation rho

data:  V(graph.cladeC)$size and pop.betweenness
S = 3259.634, p-value = 0.5779
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.1144049
```

Conditional Genetic Distance

In Dyer *et al.* (2010) we showed that graph distance (e.g., the shortest path connecting points in the topology) was more powerful than pair-wise structure and distance approaches. We denoted the among population distance as *cGD* for conditional graph distance.

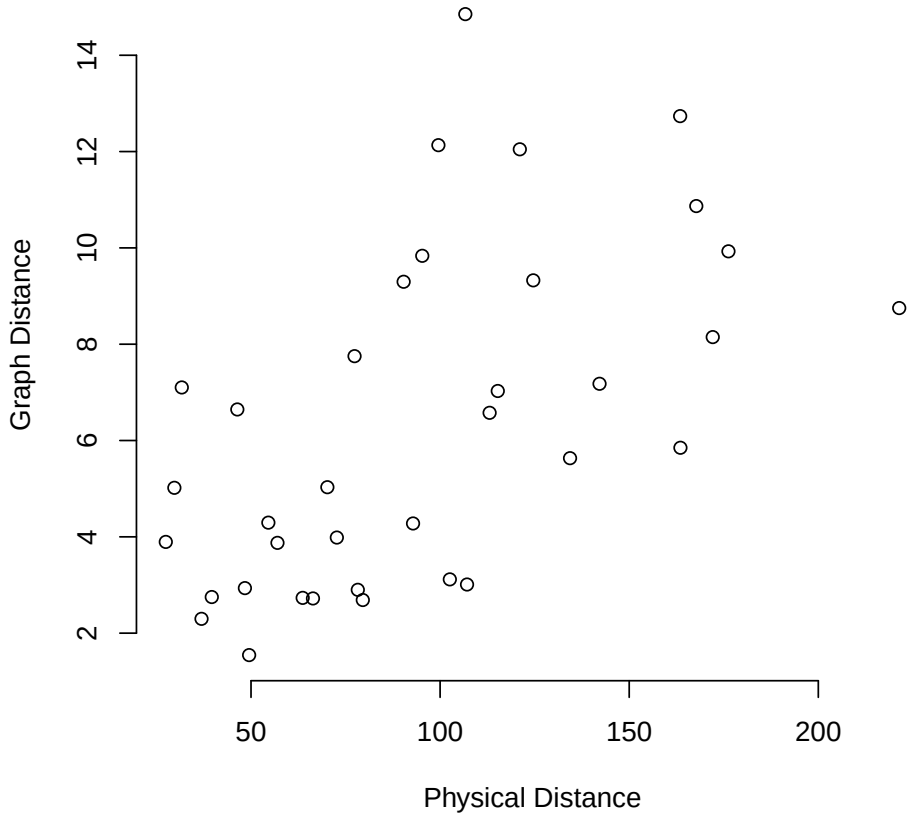
Since this topology is disconnected, we'll just focus on the medium sized component, the one with `84` in it.

```
> connected.to.84 <- subcomponent(graph.cladeC,v="84")
> med.graph <- subgraph(graph.cladeC,v=connected.to.84)
> med.layout <- layout.fruchterman.reingold(med.graph)
> plot(med.graph,layout=med.layout,vertex.label=V(med.graph)$name)
> D <- shortest.paths(med.graph)
```



As discussed previously, we can also get the pair-wise physical distance and then examine "Isolation by Graph Distance" (IBGD), which has some nice properties that make it perhaps more precise than IBD based upon pair-wise structure estimates.

```
> pops <- V(med.graph)$name
> P <- stratum.distance(baja.cladeC,"Pop",lat="Lat",lon="Long",subset=pops)
> plot(D[lower.tri(D)] ~ P[lower.tri(P)], bty="n",xlab="Physical Distance",ylab="Graph Distance")
```



We can use a Mantel test to see if there is a correlation between graph and physical distance for this subcomponent.

```
> require(ecodist,quietly=T)
> mantel(as.dist(D)~as.dist(P)) ##pval3 is Ho: Mantel-R=0
```

```
   mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
 0.5687480  0.0070000  0.9940000  0.0070000  0.4855738  0.7003350
```

The `pval3` is the probability of $H_O : Mantel\rho = 0$.

Graph Partitions

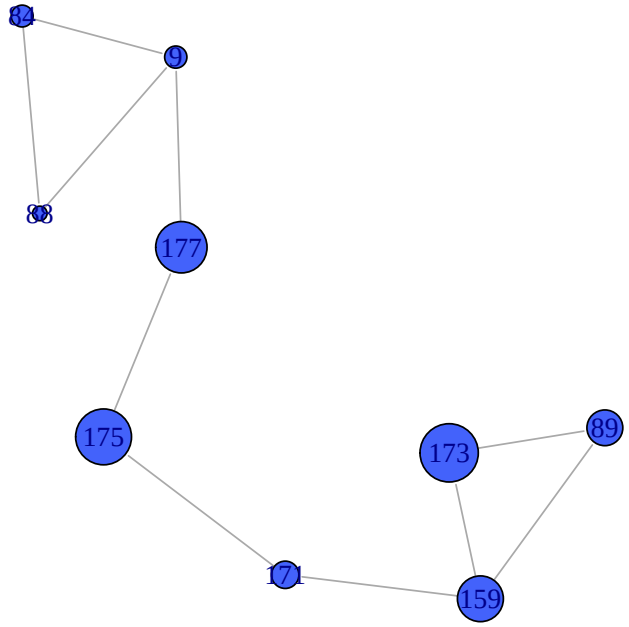
A very important point needs to be made here regarding subgraphs and partitions of the whole data set. The disconnected subgraph in the previous section is not necessarily the same graph you would get if you partitioned the genotypes into only those populations and then make the graph. Compare the previous network topology to this one.

```
> tmp.pop <- baja[baja$Pop %in% c("9","84","88","89","159","171","173","175","177")]
> tmp.graph <- population.graph(tmp.pop,"Pop")
```

```
tranforming data... done
Rotating mv genos and partitioning...  done
```

```
Estimating conditional genetic covariance... done
Making graph... done
```

```
> plot(tmp.graph,layout=med.layout,vertex.label=V(tmp.graph)$name)
```



This is because Population Graphs are constructed using *Conditional Genetic Covariance*. The genetic covariance between populations `173` & `171` is conditional on the their covariance with all the other data in the data set. In the first graph this includes the populations in this subgraph as well as the populations outside the subgraph.