EMBO Population Genomics Practical 2

Andrea Manica

For this practical, we will use the f2 we pre-calculated in the previous practical. We can simply reload them with

```
library(admixtools)
library(tidypopgen)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: tibble
```

```
f2_blocks = f2_from_precomp("./data/f2_tidypopgen", verbose = FALSE)
```

qpWave

qpWave allows us to estimate the lower bound of the number of waves that have gone from the right to the left populations. Let us start by looking at the possibility of hybridisation between archaic hominins and European modern populations:

```
neand_euras_wave <- qpwave(data = f2_blocks,
      left = c("French","Spanish","Tuscan"),
      right = c("AltaiNea","Mota", "Yoruba", "Denisova", "Mbuti")
)
```

```
## i Computing f4 stats...
## i Computing number of admixture waves...
```

```
neand_euras_wave
```

```
## $f4
## # A tibble: 8 x 8
##   pop1   pop2    pop3     pop4           est        se      z      p
##   <chr>  <chr>   <chr>    <chr>        <dbl>     <dbl>  <dbl>  <dbl>
## 1 French Spanish AltaiNea Mota     0.000206  0.000101  2.05   0.0403
## 2 French Spanish AltaiNea Yoruba   0.000200  0.0000826 2.42   0.0155
## 3 French Spanish AltaiNea Denisova 0.000129  0.0000622 2.07   0.0387
## 4 French Spanish AltaiNea Mbuti    0.000136  0.0000793 1.71   0.0873
## 5 French Tuscan  AltaiNea Mota     0.000212  0.000178  1.19   0.235
## 6 French Tuscan  AltaiNea Yoruba   0.000123  0.000140  0.883  0.377
## 7 French Tuscan  AltaiNea Denisova 0.0000901 0.000102  0.884  0.377
## 8 French Tuscan  AltaiNea Mbuti    0.0000110 0.000140  0.0784 0.937
##
## $rankdrop
## # A tibble: 2 x 7
##   f4rank   dof chisq     p dofdiff chisqdiff p_nested
##    <int> <int> <dbl> <dbl>   <int>     <dbl>    <dbl>
## 1      1     3  2.00 0.573       5      9.27   0.0986
## 2      0     8 11.3  0.187      NA     NA     NA
```

So, we can see that we can't reject the rank 0 matrix, so there is just one wave maximum into Europe (consistent with the idea that we see similar Neanderthal admixture in all populations).

QUESTION: Now, add the 'Han' and 'Onge' to the set of left populations and repeat the test. Does the result change? What does it show?

apAdm

qpAdm allows to estimate the proportion of admixture from multiple populations required to create a target population. It has been argued that modern European populations are the result of mixing of up to three ancestral streams: the local Mesolithic Hunter-Gatherers, the Levant Neolithic farmers, and the Yamnaya (who arrived during the Bronze age). In our dataset, we have examples of these populations named "Loschbour", "LBK" and "Yamnaya".

Let us start modelling the modern French:

```
french_adm <- qpadm(data = f2_blocks,
      left = c("Loschbour", "LBK", "Yamnaya"),
      right = c("Mbuti", "Mota", "Dinka", "Yoruba", "Han"),
      target= "French")
```

```
## i Computing f4 stats...
## i Computing admixture weights...
## i Computing standard errors...
## i Computing number of admixture waves...
```

The object produced by `qpadm` includes a number of elements; the most useful is

```
french_adm$popdrop
```

```
## # A tibble: 7 x 14
##   pat      wt   dof chisq        p f4rank Loschbour    LBK Yamnaya feasible best
##   <chr> <dbl> <dbl> <dbl>    <dbl>  <dbl>     <dbl>  <dbl>   <dbl> <lgl>    <lgl>
## 1 000       0     2  5.64 5.96e-2      2    -0.386  0.161    1.22  FALSE    NA
## 2 001       1     3  7.48 5.81e-2      1     0.536  0.464   NA     TRUE     TRUE
## 3 010       1     3  5.83 1.20e-1      1    -0.828 NA        1.83  FALSE    TRUE
## 4 100       1     3  6.21 1.02e-1      1    NA      0.271    0.729 TRUE     TRUE
## 5 011       2     4 15.0  4.64e-3      0     1     NA       NA     TRUE     NA
## 6 101       2     4 18.7  9.12e-4      0    NA      1       NA     TRUE     NA
## 7 110       2     4 10.5  3.25e-2      0    NA     NA        1     TRUE     NA
## # i 3 more variables: dofdiff <dbl>, chisqdiff <dbl>, p_nested <dbl>
```

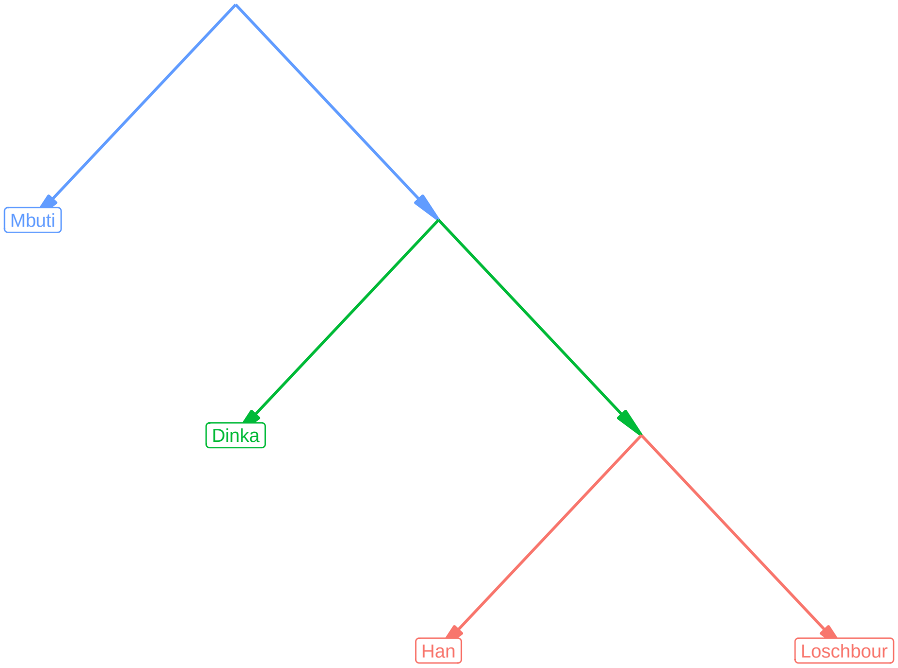
This table compares models in which different populations have been dropped. The `pat` column shows the left populations that are present in the model: 0 means present, 1 absent. So, 000 is the saturated model (with all left populations), 100 is a model without the first population (Loschbur), 010 without the second (without LBK), etc. For each model, we are given a test of whether the model can be rejected given the data (`chisq` and `p`; obviously we want a model that is compatible with the data, so NOT significant). However, note that some models might fit with the data, but do so by having negative contributions (which makes mathematical sense, but are not biological feasible); this is highlighted in the column `feasible`. Furthermore, qpAdm can estimate the most likely number of sources. Models with the correct number of sources are given by `best`.

In this case, we can see that both 000 and 100 can not be rejected (p>0.05). However, the best models column shows that the most supported scenario is one with only two sources (one of '001', '010' and '100'); of those, the two that are rejected are also not feasible (as they have negative contributions), leaving us with only '100'. So the French are best modelled as a mix of LBK and Yamnaya. Looking at the proportions, it is 1/4 LBK vs 3/4 Yamnaya.

QUESTION: Now, let's look at the Basque. Can they be modelled in the same way? How large is the Yamnaya component compared to the French?

qpGraph

We now want to create a very simple graph that recapitulates what we know of the most important events of the demography history of humans. Africa has some deep structure, and from a group related to the Dinka, humans came out of Africa, and split into Europeans and Asians:



To create such a graph, we need to create an edge matrix. Note that the edges can connect either a population in the dataset or an internal node. We are free to call those nodes anything we want. In the following example, we will use: "R" for root, "eAfr" for the east African last common ancestors between Dinka and non Africans, and "outAfrica" as the most recent common ancestor between Asians (represented by the Han) and Europeans (represented by Loschbur).

```
base_edges <- matrix(
  c("R",    "Mbuti",
    "R", "eAfr",
    "eAfr", "Dinka",
    "eAfr", "outAfrica",
    "outAfrica",    "Han",
    "outAfrica",    "Loschbour"),
  ncol=2,
  byrow = TRUE,
  dimnames=list(NULL, c("from","to")))

base_edges
```

```
##      from        to
## [1,] "R"         "Mbuti"
## [2,] "R"         "eAfr"
## [3,] "eAfr"      "Dinka"
## [4,] "eAfr"      "outAfrica"
## [5,] "outAfrica" "Han"
## [6,] "outAfrica" "Loschbour"
```

You can create the matrix in multiple ways, but the approach above allows to neatly write each edge as a line. We can now convert the edge list into an `igraph` object:

```
base_igraph <- base_edges %>% edges_to_igraph()
```

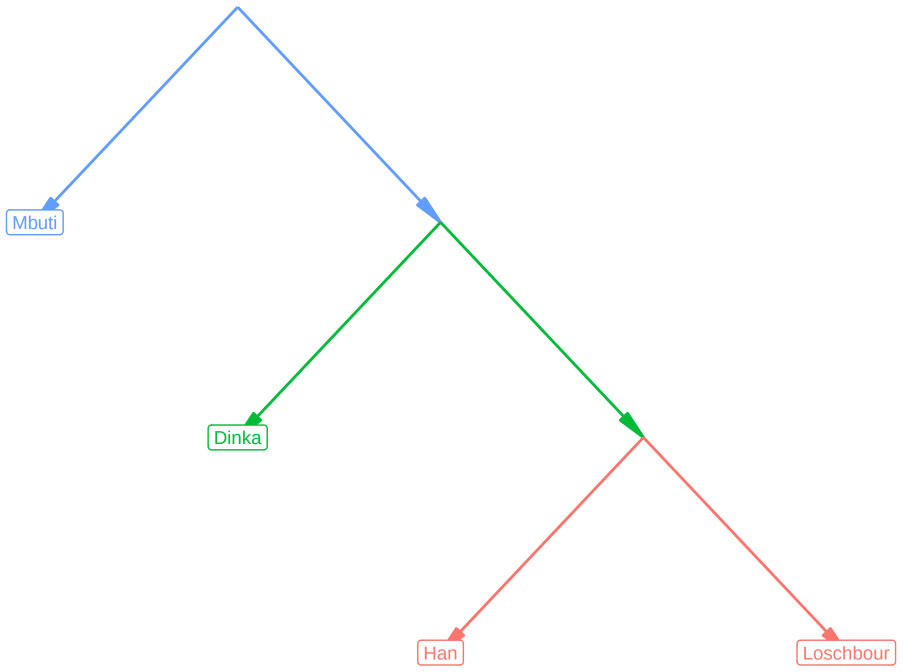
`igraph` is a commoly used library to represent graphs, and it can represent graphs which are not suitable admixture graphs (e.g. graphs that are cyclic, or with multiple simultaneous splits). We can verify that our graph is valid with:

```
is_valid(base_igraph)
```

```
## [1] TRUE
```

Now we plot it to confirm that we have obtained the graph that we wanted:

```
base_igraph %>% plot_graph()
```



This looks good. But for inspecting and interacting with admixture graphs, it is often useful to be able to get the labels of internal nodes. We can do that easily with the interactive library `plotly`:

```
base_igraph %>% plotly_graph()
```

If you hover over the edges and internal nodes, you will be shown the appropriate labels.

We are now ready to fit our graph to data:

```
base_qpgraph <- qpgraph(data = f2_blocks, graph = base_igraph)
```

Remember from the lecture that we assess qpgraph based on f3. We can request to see the comparison of the predicted vs observed f3 simply with:

```
base_qpgraph$f3
```

```
## # A tibble: 6 x 9
##   pop1  pop2      pop3         est       se    fit         diff       z     p
##   <chr> <chr>     <chr>      <dbl>    <dbl>  <dbl>        <dbl>   <dbl> <dbl>
## 1 Mbuti Dinka     Dinka     0.0241 0.000295 0.0241  0.00000332  0.0112 0.991
## 2 Mbuti Dinka     Han       0.0218 0.000313 0.0218 -0.0000568  -0.181  0.856
## 3 Mbuti Dinka     Loschbour 0.0220 0.000357 0.0218  0.000156    0.436  0.663
## 4 Mbuti Han       Han       0.0712 0.000537 0.0712 -0.0000705  -0.131  0.895
## 5 Mbuti Han       Loschbour 0.0531 0.000588 0.0530  0.0000598   0.102  0.919
## 6 Mbuti Loschbour Loschbour 0.0912 0.00121  0.0910  0.000248    0.206  0.837
```

We can see that our graph is compatible with the data. All comparisons of the f3 estimated from the data (`est`) versus the fit from the graph (`fit`) have differences that are very small, resulting in *z* values close to zero and consequently very large p-values).

A z value of 2 or 3 is often used as a threshold for a good fit. We can check that there are no combination with such a large value with:

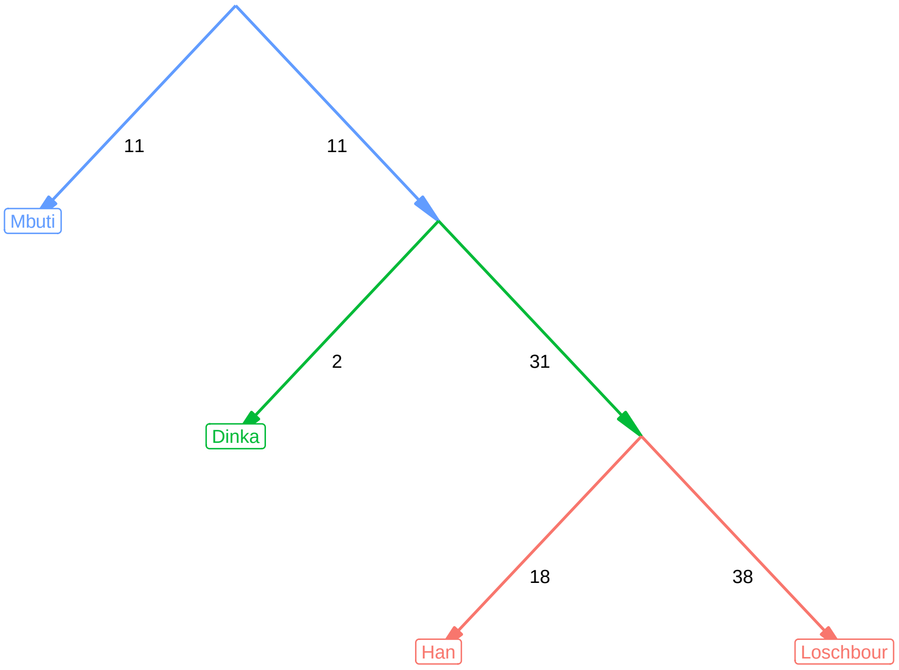
```
base_qpgraph$f3 %>% filter(abs(z)>2)
```

```
## # A tibble: 0 x 9
## # i 9 variables: pop1 <chr>, pop2 <chr>, pop3 <chr>, est <dbl>, se <dbl>,
## #   fit <dbl>, diff <dbl>, z <dbl>, p <dbl>
```

There are no such values, so we can conclude that the graph is compatible with the data (note that this does not mean that the graph is a correct representation of the past, there could be multiple graphs that fit the data).

Let's visualise it:

```
base_qpgraph$edges %>% plot_graph()
```



If you substitute `plot_graph()` with `plotly_graph()` you can get an interactive graph.

Careful around the root

Note that admixture graphs are ill suited at defining the topology around the root; the root and the outgroup have to be defined a priori, and admixtools simply splits the distance between the two first nodes to place the root.

QUESTION: Let's see what happens if we choose the wrong outgroup. Try creating a graph where we swap the role of Mbuti and Dinka (i.e. placing Mbuti as the closest African population to European and Asians). Use `base_swapped_qpgraph` as the name for the graph you create.

How do we demonstrate that this graph is not compatible with the data? We can compare these two models more formally with:

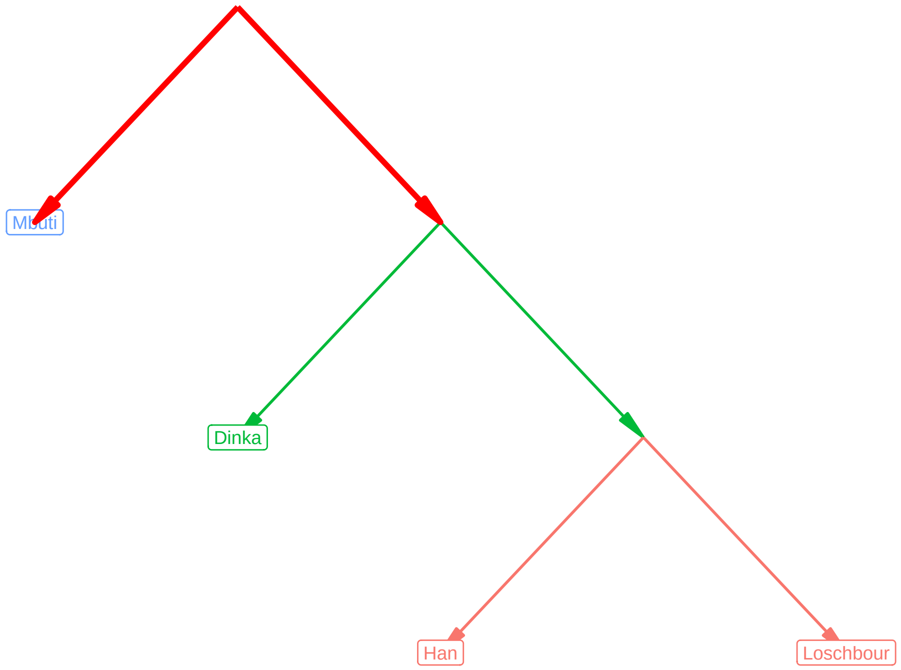
```
fits = qpgraph_resample_multi(f2_blocks,
                              graphlist = list(base_qpgraph[[1]], base_swapped_qpgraph[[1]]),
                              nboot = 100)
compare_fits(fits[[1]]$score_test, fits[[2]]$score_test)
```

```
## $diff
## [1] 0.001528927
##
## $se
##             [,1]
## [1,] 0.07306546
##
## $z
##             [,1]
## [1,] 0.02092545
##
## $p
## [1] 0.9833051
##
## $p_emp
## [1] 0.98
##
## $p_emp_nocorr
## [1] 0.98
##
## $ci_low
## [1] -0.1228281
##
## $ci_high
## [1] 0.1353509
```

In reality, the root does little here, the models are identical, and the root is simply an “aesthetic” element for how we plot the graph. So, be aware that the choice of outgroup is very important, as it will colour your interpretation but can not be easily verified (unless you choose something very wrong).

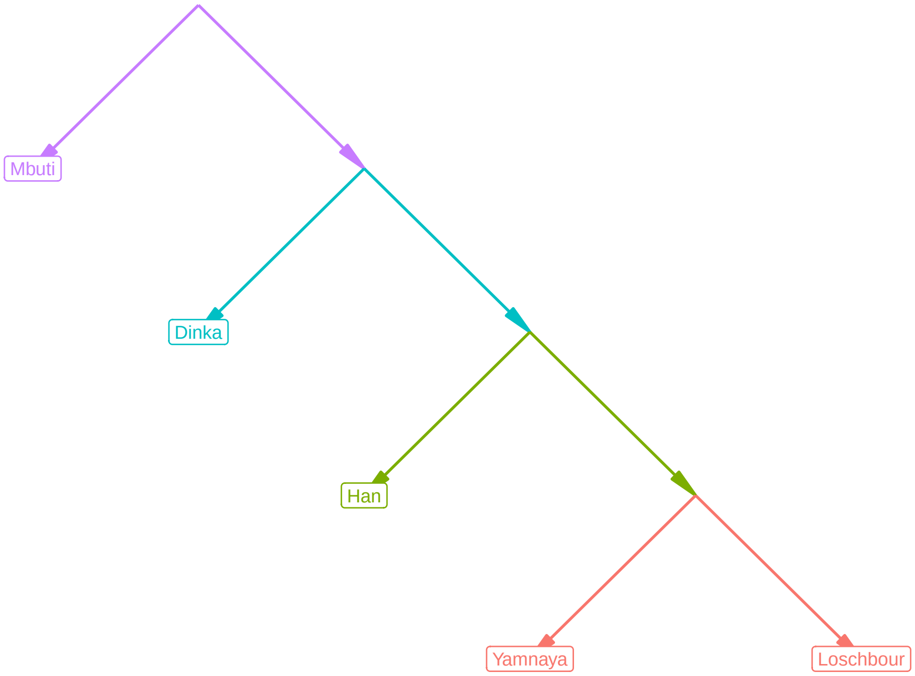
We can see the issues with the root by highlighting unidentifiable edges:

```
base_igraph %>% plot_graph(highlight_unidentifiable = TRUE)
```



Adding the Yamnaya

We can add more populations to obtain the following graph:



Note that, to add the "Yamnaya", you will need to add an internal node before Loschbour. We will call this node "wEurasian". So, the end of the previous set of edges becomes:

```
"outAfrica", "wEurasian",
"wEurasian", "Yamnaya",
"wEurasian", "Loschbour")
```

QUESTION: Add these edges and create an object names `yamnaya_igraph`, and plot it to make sure that it matches our desired topology. After checking that the graph is valid, fit the graph, inspect it, and check whether it is compatible with the data:

And ask whether our graph is compatible with the data

Does any population have a z value larger than 3? What do you conclude (remember to plot and inspect the fitted graph)?

Adding an admixture edge

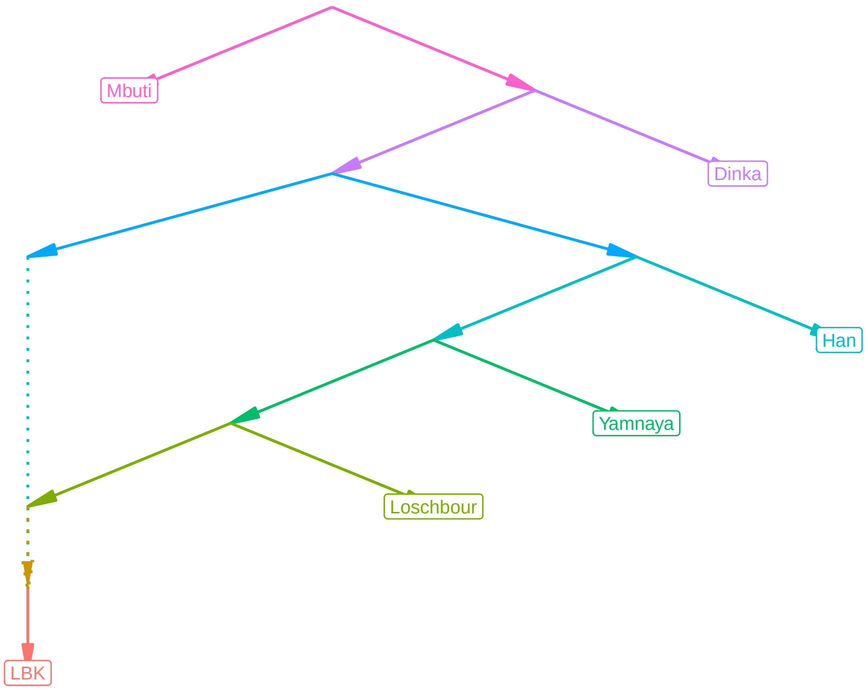
The trees that we considered so far did not have any admixture. To add admixture, we simply need to add edges such that we have two edges going to the same node (which must be an internal node). We add an early Neolithic farmer, LBK, which is modelled as a mixture of an early ghost population that came out of Africa and split before the European/Asian split,

Note that, when a real population is modelled as admixed, the admixture edges have to converge first to an intermediate internal node that is then connected to a real population (if you have two admixture edges going directly to a population, then `is_valid()` will fail).

QUESTION: Create a set of edges `lbk_edges` to match the graph above, and then fit the graph to the data. Is the model compatible with the data?

Note that, in admxiture graphs visualised with the original admixtools software, additional intermediate nodes were often added for aestethic purposes (they were labelled automatically, and so they were plotted so that we could refer to them). It is possible to add such nodes, but they are not identifiable. They are not a problem, they do not affect the fit, so it is a matter of personal preference. Here is an example of such a graph:

```r
lbk_extra_edges <- matrix(
  c(
    "R",    "Mbuti",
    "R",    "eAfr",
    "eAfr", "pBasalEurasian",
    "eAfr", "Dinka",
    "pBasalEurasian", "BasalEurasian",
    "pBasalEurasian","outAfrica",
    "outAfrica", "Han",
    "outAfrica","wEurasian",
    "wEurasian", "Yamnaya",
    "wEurasian", "pLoschbour",
    "pLoschbour", "Loschbour",
    "pLoschbour","WHG",
    "BasalEurasian", "pLBK",
    "WHG", "pLBK",
    "pLBK","LBK"),
  ncol = 2,
  byrow = TRUE,
  dimnames = list(NULL, c("from", "to")))
lbk_extra_igraph <- lbk_extra_edges %>% edges_to_igraph()
lbk_extra_igraph %>% plot_graph()
```



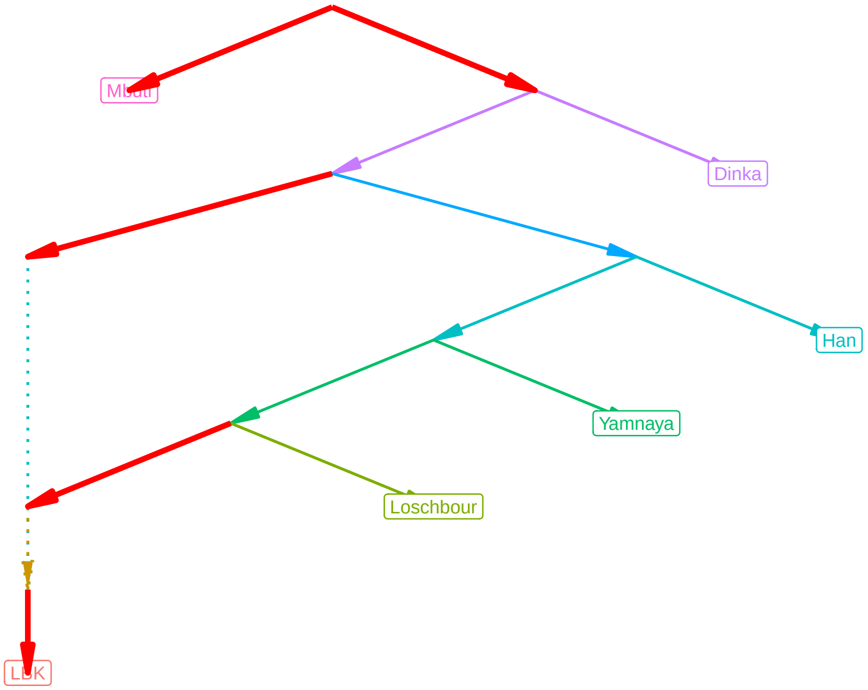
This graph is valid:

```
is_valid(lbk_extra_igraph)
```

```
## [1] TRUE
```

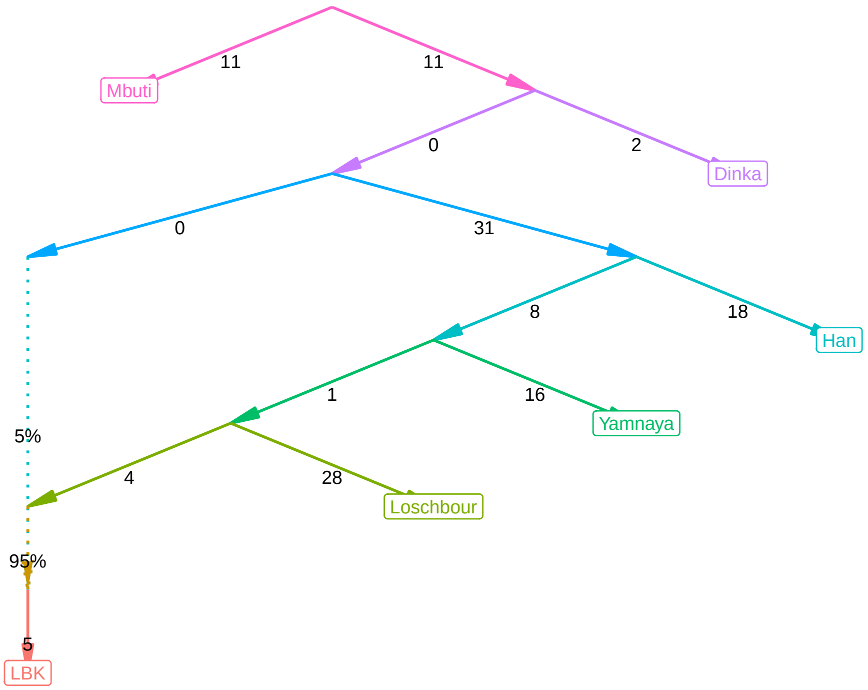
But the edges that we added are not identifiable:

```
lbk_extra_igraph %>% plot_graph(highlight_unidentifiable = TRUE)
```



If we fit the model, we can notice that some of the edges that we added have zero drift (a clear indication they are redundant), but note that the edge going to the generic WHG that is the sister population to Loschbour returns a small drift, even though it is not identifiable:

```r
lbk_extra_qpgraph <- qpgraph(data = f2_blocks, graph = lbk_extra_igraph)
lbk_extra_qpgraph$edges %>% plot_graph()
```



Note that the admixture proportion is identical to the previous estimates even though we added those unidentifiable edges; so, they are purely aesthetic, they do not impact our fit.

Adding another edge

QUESTION: If you have additional time, can you add an addition admixture edge, exploring what happens if you add the Sardinians. Try modelling the Sardinian as a direct descendant of the ancestor of LBK, and then try to add some admixture from the Yamnaya (as we saw earlier that we could detect that signal). Can you get away with modelling just a Neolithic component, or do you need the Yamnaya admixture to make the model work.

If you have clear hypotheses, it is best to build alternative graphs and compare them. On the other hand, if you are trying to place populations without a clear framework, you would be better off automatically exploring many graphs. As that exploration is computatonally intensive, we will not do that during this practical, but you have the building blocks on how to compare models, so you would just compare a lot of different models rather than 2.