Inferring admixture in wild populations of yeast using admixture graphs

Anna Fijarczyk
13/03/2018
Why admixture graphs?

• Inference of admixture events in populations of wild yeast *Saccharomyces paradoxus*

• Utility of admixture graphs for inferring history of admixture in genomic data
Inference of admixture history

Exploratory analysis

Modeling demographic parameters

vonHoldt et al. 2016

Paukopoulou et al. 2018
Inference of admixture history

Admixture graphs

1) branch lengths – measures of drift
2) admixture proportions

Paukopoulou et al. 2018
qpGraph/AdmixTools
Constructing admixture graphs

- qpGraph in AdmixTools (Patterson et al. 2012)
- MixMapper (Lipson et al. 2013)
- TreeMix (Pickrell & Pritchard 2012)
- admixturegraph R package (Leppälä et al. 2017)
The case of *S. paradoxus*

Distribution of 5 lineages in North America

Eberlein et al. in prep.
SpC* is a hybrid between SpC and SpB

Leducq et al. 2016
Is SpD a hybrid?

Eberlein et al. in prep.
Possible scenarios of emergence of SpD

- SpD is a hybrid between SpB & SpC*
- SpD is a hybrid between SpB & SpC
Possible scenarios of emergence of SpD

Model 1

Model 2
admixturegraph R package

1. Calculate f statistics
2. Specify and visualize admixture graphs
3. Fit a graph to data summarized as f statistics and test the goodness of fit
4. Compare model likelihoods
1. Calculate f4 statistics
f4 statistic

\[ (y - z) \times (w - x) \]

w, x, y, z: frequencies of allele a1 in each population

Patterson 2012
f4 statistic

\[ (y - z) \times (w - x) \]

\( w, x, y, z: \) frequencies of allele a1 in each population

<table>
<thead>
<tr>
<th></th>
<th>Population 1</th>
<th>Population 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>0.19</td>
<td>0.36</td>
</tr>
<tr>
<td>x</td>
<td>0.13</td>
<td>0.42</td>
</tr>
</tbody>
</table>

\[ f4 = 0 \]
w, x, y, z: frequencies of allele a1 in each population

(y - z) * (w - x)

0.42  0.19  0.13  0.36  f4 < 0

Patterson 2012
f4 statistic

\[ (y - z) \times (w - x) \]

\( w, x, y, z: \) frequencies of allele a1 in each population

0.19 0.43 0.13 0.36 \( f4 > 0 \)

Sum and divide by N loci or normalize to get D
f statistics and D statistics

Admixture and Ancestry Inference from Ancient and Modern Samples through Measures of Population Genetic Drift

Alexandre M. Harris¹ and Michael DeGiorgio¹ *

1. Calculate f statistics

- AdmixTools - qpDstat (eigenstrat, ped)
- scikit-allel python package (vcf)
- popstats (ped)
### f4 statistics – qpDstat output

<table>
<thead>
<tr>
<th>Pop1</th>
<th>Pop2</th>
<th>Pop3</th>
<th>Pop4</th>
<th>f4-stat</th>
<th>Z</th>
<th>#BABA</th>
<th>#ABBA</th>
<th>#N SNPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>0.108051</td>
<td>15.724</td>
<td>23105</td>
<td></td>
<td>1293 201872</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>Ci</td>
<td>D</td>
<td>0.103952</td>
<td>15.211</td>
<td>22334</td>
<td></td>
<td>1349 201872</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
<td>Ci</td>
<td>0.129948</td>
<td>43.133</td>
<td>28009</td>
<td></td>
<td>1776 201872</td>
</tr>
<tr>
<td>A</td>
<td>Ci</td>
<td>B</td>
<td>C</td>
<td>0.125849</td>
<td>30.693</td>
<td>28009</td>
<td></td>
<td>2604 201872</td>
</tr>
<tr>
<td>A</td>
<td>Ci</td>
<td>B</td>
<td>D</td>
<td>0.063648</td>
<td>10.407</td>
<td>14198</td>
<td></td>
<td>1349 201872</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
<td>D</td>
<td>0.057948</td>
<td>10.050</td>
<td>12991</td>
<td></td>
<td>1293 201872</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>C</td>
<td>Ci</td>
<td>0.009798</td>
<td>5.048</td>
<td>3884</td>
<td></td>
<td>1906 201872</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>C</td>
<td>Ci</td>
<td>0.005700</td>
<td>5.164</td>
<td>2370</td>
<td></td>
<td>1220 201872</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>Ci</td>
<td>0.004099</td>
<td>3.301</td>
<td>2604</td>
<td></td>
<td>1776 201872</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>Ci</td>
<td>B</td>
<td>0.040304</td>
<td>3.192</td>
<td>22334</td>
<td></td>
<td>14198 201872</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>0.050102</td>
<td>4.020</td>
<td>23105</td>
<td></td>
<td>12991 201872</td>
</tr>
<tr>
<td>A</td>
<td>Ci</td>
<td>D</td>
<td>C</td>
<td>0.062201</td>
<td>10.668</td>
<td>16440</td>
<td></td>
<td>3884 201872</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>D</td>
<td>Ci</td>
<td>0.072000</td>
<td>14.651</td>
<td>16440</td>
<td></td>
<td>1906 201872</td>
</tr>
<tr>
<td>B</td>
<td>Ci</td>
<td>D</td>
<td>C</td>
<td>0.170252</td>
<td>14.016</td>
<td>36740</td>
<td></td>
<td>2370 201872</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>D</td>
<td>Ci</td>
<td>0.175952</td>
<td>15.048</td>
<td>36740</td>
<td></td>
<td>1220 201872</td>
</tr>
</tbody>
</table>

### Populations:
- A – SpA
- B – SpB
- C – SpC
- Ci – SpC*
- D – SpD
### f4 statistic

```r
> data
   W X Y Z    D  Z.value
 1 A B C D 0.108051 15.724
 2 A B C D 0.103952 15.211
 3 A C B C 0.129948 43.133
 4 A C B C 0.125849  6.693
 5 A C B D 0.063648 10.407
 6 A C B D 0.057948 10.050
 7 A D C C 0.009798  5.048
 8 B D C C 0.005700  5.164
 9 A B C C 0.004099  3.301
10 A D C B 0.040304  3.192
11 A D C B 0.050102  4.020
12 A C D C 0.062201 10.668
13 A C D C 0.072000 14.651
14 B C D C 0.170252 14.016
15 B C D C 0.175952 15.048
> plot(f4stats(data))
> |
```
2. Specify and visualize admixture graphs

```r
> leaves <- c("B", "D", "C1", "C", "A")
> inner_nodes <- c("b1", "d", "c1", "b2", "c3", "c2", "bc", "abc")
> edges <- parent_edges(c(edge("B", "b1"), edge("b1", "b2"), edge("b2", "bc"), edge("bc", "abc"),
+                      edge("D", "d"), edge("C1", "c1"), edge("c1", "c3"), edge("C", "c2"), edge("c2", "bc"),
+                      edge("A", "abc"),
+                      admixture_edge("d", "b1", "c1", "alpha"),
+                      admixture_edge("c3", "b2", "c2", "alpha2")))
> graph1 <- agraph(leaves, inner_nodes, edges)
> plot(graph1, show_inner_node_labels = TRUE, show_admixture_labels = TRUE)
```
2. Specify and visualize admixture graphs

**Model 1**

**Model 2**

3. Fit a graph to data

```r
> f2 <- fit_graph(data, graph2)
> plot(f2)
> summary(f2)

Call: inner_fit_graph(data, graph, point, Z.value, concentration, optimisation_options,
  parameters, iteration_multiplier, qr_tol)

After fixing \( \{\alpha_2\} \) none of the remaining proportions \( \{\alpha_1\} \) affect the quality of the fit!

Optimal admixture proportions:
  alpha1  alpha2
  0.000011  0.528040

Optimal edge lengths:
  edge_b2.b1  edge_b2.c*  edge_d.D  edge_c2.c*  edge_c2.c1  edge_b1.B  edge_b1.d  edge_c*.Ci  edge_c1.C  edge_c1.d  edge_bc.b2
  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.2064575
  edge_bc.c2  edge_abc.A  edge_abc.bc
  0.1285932  0.000000  0.000000
```
3. Fit a graph to data

```r
> f2 <- fit_graph(data, graph2)
> plot(f2)
```
3. Fit a graph to data

- **Model 1**: admixture proportion: $a = 0.55$

- **Model 2**:
4. Compare model likelihoods

Extracting data to build MCMC

```r
> mcmc1 <- make_mcmc_model(graph1, data)
> mcmc2 <- make_mcmc_model(graph2, data)
> mcmc1$parameter_names
[1] "alpha"    "alpha2"  "edge_b2_b1"  "edge_b2_c3"  "edge_c3_c1"  "edge_c2_c3"  "edge_bc_b2"  "edge_bc_c2"
> mcmc2$parameter_names
[1] "alpha1"    "alpha2"  "edge_b2_b1"  "edge_c2_c1"  "edge_bc_b2"  "edge_bc_c2"
```

Running MCMC

```r
> initial1 <- rep(0.5, length(mcmc1$parameter_names))
> initial1
[1] 0.5 0.5 0.5 0.5 0.5 0.5 0.5
> chain1 <- run_metropolis_hasting(mcmc1, initial1, iterations = 100000, verbose = FALSE)
> initial2 <- rep(0.5, length(mcmc2$parameter_names))
> chain2 <- run_metropolis_hasting(mcmc2, initial2, iterations = 100000, verbose = FALSE)
```

Comparing model likelihoods

```r
> thinned_1 <- thinning(burn_in(chain1_1, 10000), 50)
> thinned_2 <- thining(burn_in(chain2_1, 10000), 50)
>
> model_likelihood_r(thinned_1[, "likelihood"], 100)

   mean     sd
[1,] -5.185961 0.6846644

> model_likelihood_r(thinned_2[, "likelihood"], 100)

   mean     sd
[1,] -46.82939 0.728294

> model_bayes_factor_r(thinned_1[, "likelihood"], thinned_2[, "likelihood"], 100)

   mean     sd
[1,] 41.76783 0.9816662
```
Possible scenarios of emergence of SpD

SpD is a hybrid between SpB & SpC*

SpD is a hybrid between SpB & SpC

SpC* is a hybrid between SpC and SpD

SpC* is a hybrid between SpB and SpD
Possible scenarios of emergence of SpD

Model 1

Model 2

Model 3

Model 4
Fit a graph to data
B-like regions are much longer in SpD than in SpC*

Distribution of B-like fragments in SpC* and SpD
Conclusions

+ Flexibility in testing multiple hypotheses
+ Estimations of admixture proportions
+ Good documentation

- time of admixture not taken into account
- any prior guess about admixture not taken into account
Thank you

Landry Lab

S. paradoxus genomics

Chris Eberlein  Mathieu Hénault

Matteo Bouvier  Gil Charron

Landry Lab

Université Laval

IBIS