hybriddetective Example

This example of the use and utility of hybriddetective is a workthrough of the development of a collectively diagnostic panel of SNP markers for the detection of hybridization between northern and southern ecotypes of European green crab (*Carcinus maenus*) as originially described in Jeffery et al. 2017. Unless otherwise noted, all functions will be run using their default parameter settings. Descriptions of the functions and their parameters can be found in the README file.

Before beginning, the R packages genepopedit genepopedit and parallelnewhybrid have been installed to R from GitHub.

Install genepopedit

```
devtools::install_github("rystanley/genepopedit")

## Skipping install of 'genepopedit' from a github remote, the SHA1 (af
54eb71) has not changed since last install.
##   Use `force = TRUE` to force installation
```

Install parallelnewhybrid

```
devtools::install_github("bwringe/parallelnewhybrid")

## Skipping install of 'parallelnewhybrid' from a github remote, the SH
A1 (70d3ab18) has not changed since last install.
##   Use `force = TRUE` to force installation
```

Load these packages into the R environment

```
library(hybriddetective)

## Warning: replacing previous import 'plyr::summarise' by 'dplyr::summ
arise'
## when loading 'hybriddetective'

library(genepopedit)
```

In addition to these packages, the programs PGDSpider and PLINK must also be installed on the user's computer.

Step 1 - use the function *getTopLoc* to produce a panel of the 200 most informative loci

Use the function *getTopLoc* by providing an input genotype file (genepop format), the number of SNPs to select for the panel and the paths to PLINK and PGDSpider. The genepop file must contain two groups (populations) of individuals that are genetically distinct, where hybridization between the two groups is of interest for the specific project.

Note: *where.plink.path* and *where.PGDspider.path* are computer specific paths to the folders in which PLINK and PGDSpider respectively are installed.

```
getTopLoc(GPD = "ExampleSNPdata.txt", panel.size = 200,
  where.PLINK = where.plink.path, where.PGDspider = where.PGDspider.pat
h)

## Reading Data
## Creating training and working datasets
## Calculating Fst
## Calculating Linkage

  |=================================================================| 1
00%

Writing output
## Process Completed.
```

The function has selected the 200 most informative SNPs, and created a validation data set called "ExampleSNPdata_200_Loci_Panel.txt". These are saved to the directory specified in "GPD".

To get an idea of what this file looks like, we can use the function *genepop_flatten* from the *genepopedit package*

```
genepop_flatten(genepop = "ExampleSNPdata_200_Loci_Panel.txt")[1:3, 1:5
]

##   SampleID Population SampleNum   3395   4476
## 1   NTH_01        NTH        01 004004 003003
## 2   NTH_04        NTH        04 002002 004003
## 3   NTH_07        NTH        07 002002 003003
```

Step 2 - Simulate multigenerational hybrids based on the training dataset using the function *freqbasedsim_AlleleSample*

Using the genotype file created above (the validation data set) containing the panel of informative SNPs, the function freqbasedsim_AlleleSample will simulate multigenerational hybrid datasets.

To check for inter-simulation variation in efficacy, three independent simulations will be specified by setting *NumSims* to 3.

To check for intra-simulation variation in MCMC chain convergence, three identical replicates of each of the independent simulations will be specified by setting *NumReps* to 3.

NOTE: Warnings that "some row.names duplicated" are expected and do not affect the output of the function.

```r
freqbasedsim_AlleleSample(GPD = "ExampleSNPdata_200_Loci_Panel.txt", NumSims = 3, NumReps = 3)
```

The simulation function has made 10 files, three replicates of each independently simulated dataset and an indvidual file that specifies the names of the individuals in the datafiles. The file names are "ExampleSNPdata_200_Loci_Panel_S1R1_NH.txt", "ExampleSNPdata_200_Loci_Panel_S1R2_NH.txt", "ExampleSNPdata_200_Loci_Panel_S1R3_NH.txt", "ExampleSNPdata_200_Loci_Panel_S2R1_NH.txt", "ExampleSNPdata_200_Loci_Panel_S2R2_NH.txt", "ExampleSNPdata_200_Loci_Panel_S2R3_NH.txt", "ExampleSNPdata_200_Loci_Panel_S3R1_NH.txt", "ExampleSNPdata_200_Loci_Panel_S3R2_NH.txt", "ExampleSNPdata_200_Loci_Panel_S3R3_NH.txt", and "ExampleSNPdata_200_Loci_Panel_individuals.txt".

Note: in the file names "S"" refers to the independent simulations, and "R"" refers the replicated datasets.

Note: alternatively the function *freqbasedsim_GTFreq*, which samples alleles in a slightly differ manner could have been used. The differences between *freqbasedsim_GTFreq* and *freqbasedsim_AlleleSample* are described in the README file.

Step 3 - Conduct NewHybrids analysis of the simulated data sets.

The function *parallelnh_OSX* from the R package **parallelnewhybrid** was used to run the simulated datasets through **NewHybrids**. **parallelnewhybrid** is described in detail in Wringe et al. 2017

For the purposes of this exmple, this step is omitted and users should use the provided pre-analyzed data. The data provided are from Jeffery et al. 2017, and consists of two independent simulations of 29 pure "population 1", 29 pure "population 2", 58 F1, 58 F2, 29 backcross to "population 1" and 29 backcross to "population 2" individuals. Each of these independently simulated datasets was then replicated three times. The data are in process of being archived.

Step 4 - Check results for convergence

Occasionally, the MCMC chains in NewHybrids will fail to converge. In such cases, nearly all individuals will have the highest posterior probability of membership in the F2 hybrid class. The function *preCheckR* quickly checks all results within a folder for convergence.

```
nh_preCheckR(PreDir = "Example_NewHybrids_Results/")

## PrecheckR Progress:
##
  |=================================================================| 1
00%
## [1] "Looks good bud, giv'er"
```

The function has reported that all MCMC chains have successfully converged, and evaluation of the efficacy of the panel can continue. If failure to converge is detected, the user should delete the non-converged results and re-analyze the simulated data.

Step 4 - Visualize individual cumulative probability of assignments for each analyzed dataset

This step is optional, but does provide a good way of visualizing the results.

```
nh_multiplotR(NHResults = "Example_NewHybrids_Results/")
  |
  |=================================================================| 1
00%
```

This function will produce a plot for each analysis provided. They will each look similar to this.

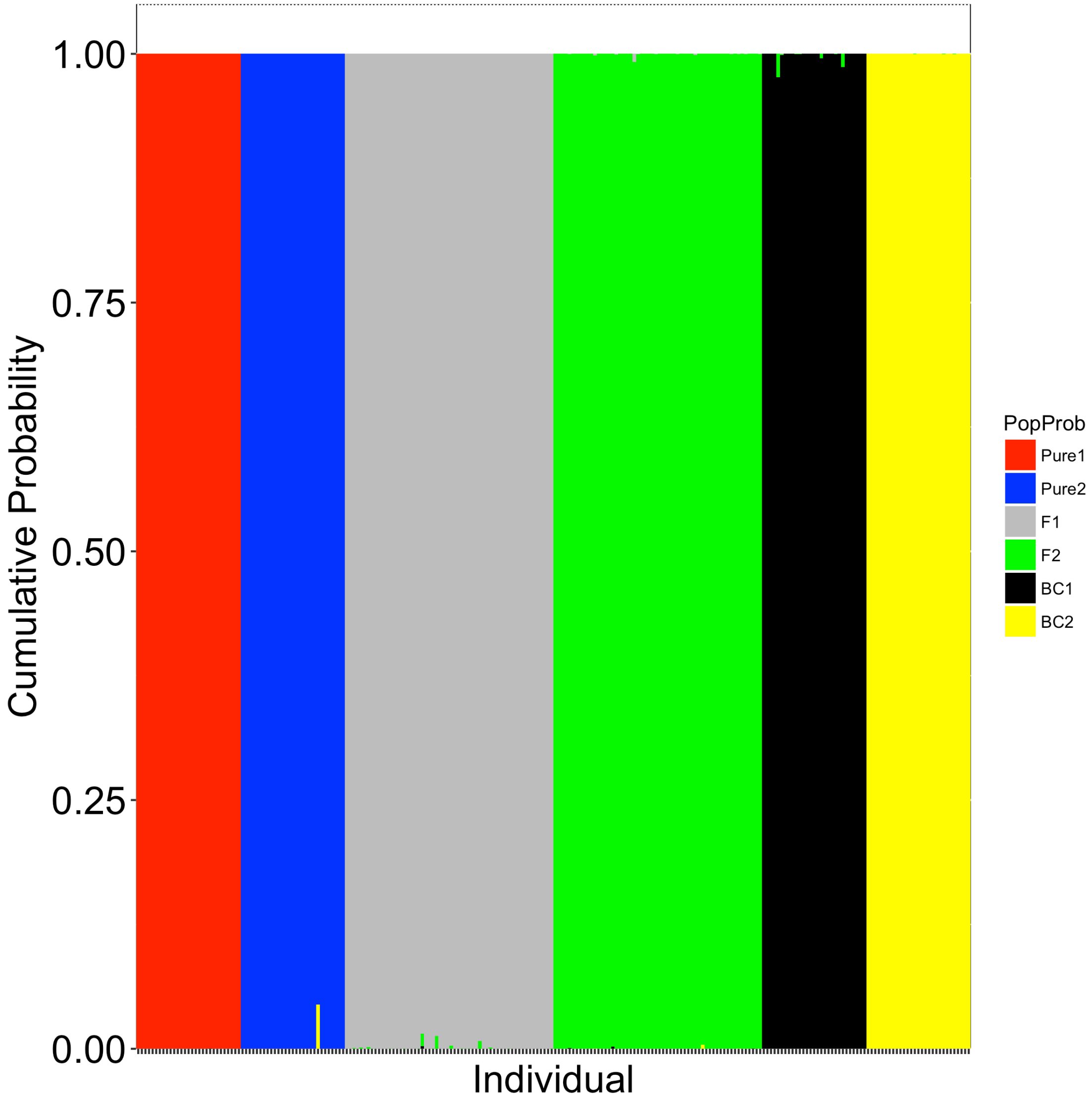

Figure 1. Example plot produced by nh_multiplotR from the evaluation of panel efficacy in Jeffery et al. 2017. Indivdiuals are along the x axis, and the y axis is the individual specific cumulative posterior probability of assignment to each of the six possible hybrid classes. The hybrid classes are denoted by colour.

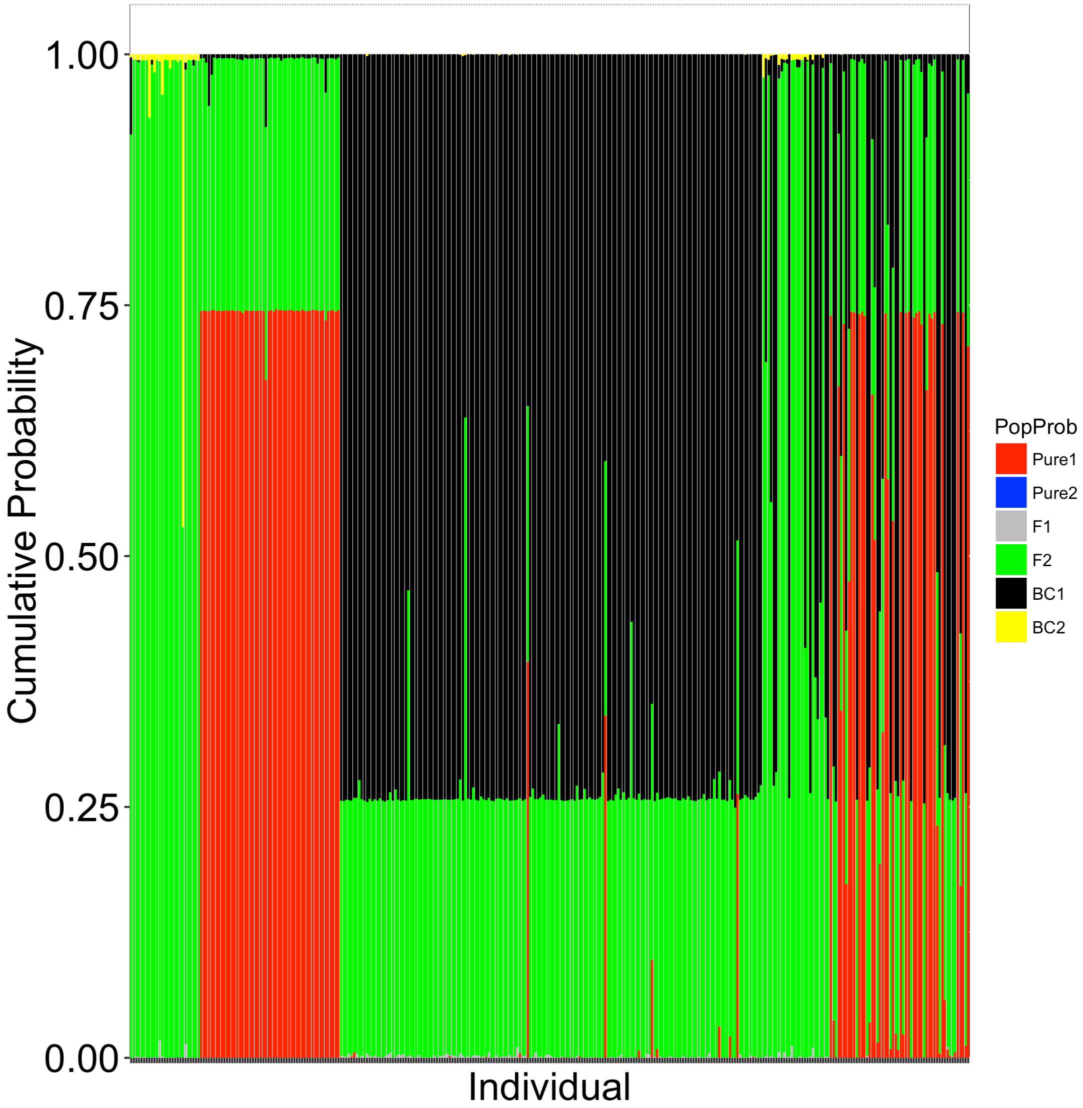

Figure 2. Example plot produced by nh_multiplotR where the MCMC chains have failed to converge. Note that most individuals have erroniously been assigned a relatively high posterior probability of assignment to the F2 class (green). Indivdiuals are along the x axis, and the y axis is the individual specific cumulative posterior probability of assignment to each of the six possible hybrid classes. The hybrid classes are denoted by colour.

Step 5 - Evaluate the hybrid-class specific efficacy of the panel across a range of posterior probability of assignment thresholds

NOTE: If *hybridPowerComp* is used to test one panel size, "geom_path" aesthetic warnings from ggplot2 will occur. This has to do with missing factor levels (i.e. multiple panel sizes) in the faceting of some of the plots (refer to Table 1). These errors do not affect the output of the function.

```
hybridPowerComp(dir = "Example_NewHybrids_Results/")

## Calculating Accuracy
##
##
##             Calculating Efficiency
##
##
##             Calculating Power!!!!
##
##
##             Calculating Mean Posterior Probabilities
##
##
##             Calculating Type I Error
##
##
##             Calculating Type II Error
##
##
##             Makin' you some plots
##
##
##             I'm saving your plots for you over here
##
##
##             I'm savin' the data too
##
```

This will create 31 plots that illustrate the efficacy of the panel tested. The function automatically makes a table describing each plot produced, and saves it as a .csv file.

```
Plot.Legend <- read.csv("Plot_Legends.csv", header = TRUE, stringsAsFac
tors = FALSE)

knitr::kable(Plot.Legend)
```

Plot	Description
Plot_1	Accuracy Boxplot by hybrid class and critical posterior probability

Plot_2	Efficiency Boxplot by hybrid class and critical posterior probability
Plot_3	Power Boxplot by hybrid class and critical posterior probability
Plot_4	Accuracy Boxplot by pure classes and all hybrids and critical posterior probability
Plot_5	Efficiency Boxplot by pure classes and all hybrids and critical posterior probability
Plot_6	Power Boxplot by pure classes and all hybrids and critical posterior probability
Plot_7	Accuracy Lineplot by hybrid class faceted by panel size
Plot_8	Efficiency Lineplot by hybrid class faceted by panel size
Plot_9	Power Lineplot by hybrid class faceted by panel size
Plot_10	Accuracy Lineplot by pure classes and all hybrids faceted by panel size
Plot_11	Efficiency Lineplot by pure classes and all hybrids faceted by panel size
Plot_12	Power Lineplot by pure classes and all hybrids faceted by panel size
Plot_13	Accuracy Lineplot by hybrid class faceted by panel size and hybrid grouping
Plot_14	Efficiency Lineplot by hybrid class faceted by panel size and hybrid grouping
Plot_15	Power Lineplot by hybrid class faceted by panel size and hybrid grouping
Plot_16	Accuracy Lineplot by panel size faceted by hybrid class
Plot_17	Efficiency Lineplot by panel size faceted by hybrid class
Plot_18	Power Lineplot by panel size faceted by hybrid class
Plot_19	Accuracy Lineplot by panel size faceted by pure classes and all hybrids
Plot_20	Efficiency Lineplot by panel size faceted by pure classes and all hybrids
Plot_21	Power Lineplot by panel size faceted by pure classes and all hybrids

Plot_22	Accuracy Dotplot by hybrid class faceted by critical posterior probability %in% Thresholds
Plot_23	Efficiency Dotplot by hybrid class faceted by critical posterior probability %in% Thresholds
Plot_24	Power Dotplot by hybrid class faceted by critical posterior probability %in% Thresholds
Plot_25	Accuracy Dotplot by pure classes and all hybrids faceted by critical posterior probability %in% Thresholds
Plot_26	Efficiency Dotplot by pure classes and all hybrids faceted by critical posterior probability %in% Thresholds
Plot_27	Power Dotplot by pure classes and all hybrids
faceted by critical posterior probability %in% Thresholds	
Plot_28	Type I Error Boxplot by hybrid class and panel size
Plot_29	Type I Error Boxplot by pure classes and all hybrids and panel size
Plot_30	Mean Posterior Probability of Assignment per simulation faceted by hybrid class
Plot_31	Mean Posterior Probability of Assignment per simulation faceted by pure classes and all hybrids

Among the files created are the overall accuracy, efficiency and power of the panel

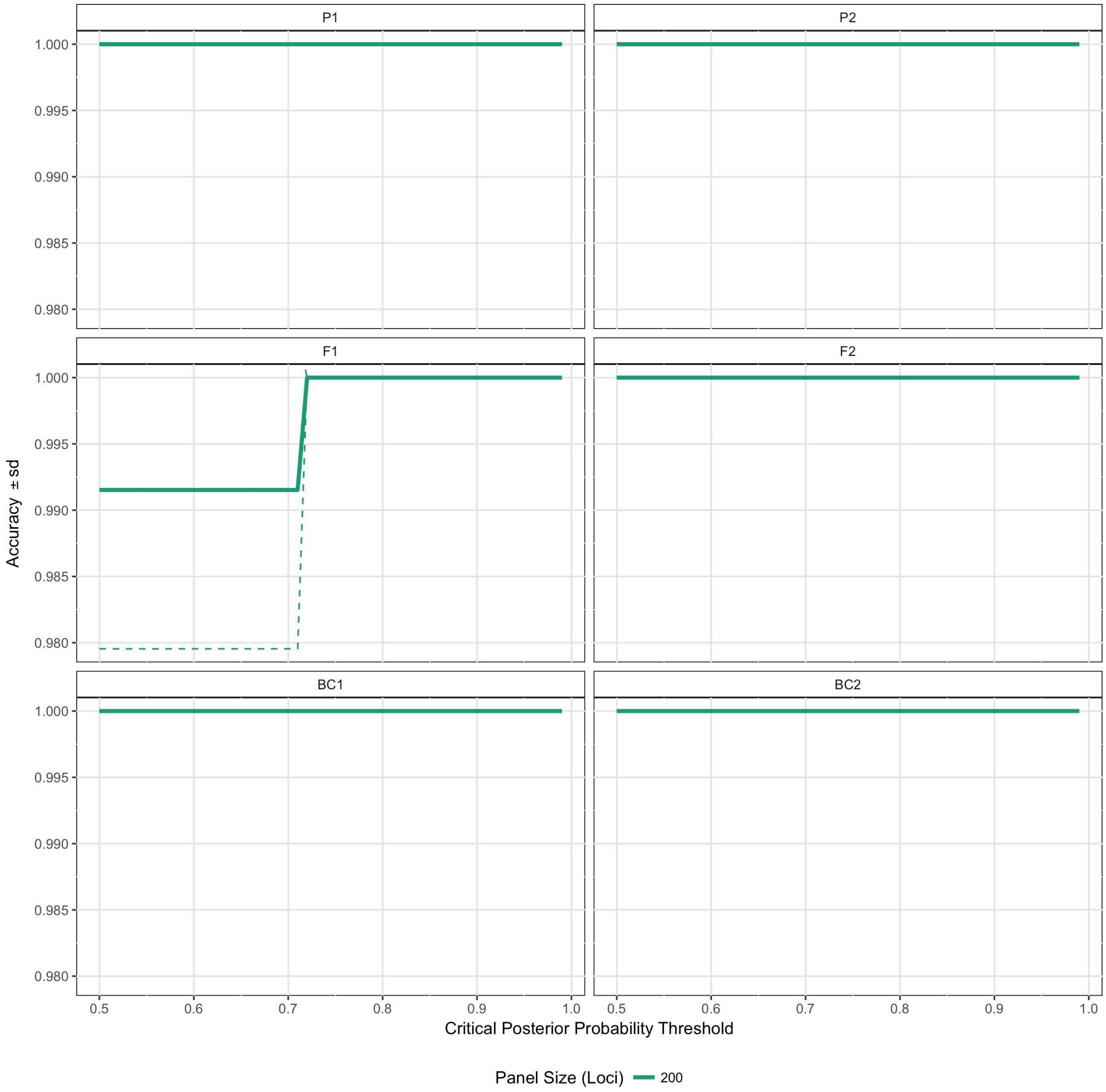

Figure 3. Example one of the accuracy plots produced by hybridPowerComp. The critical posterior probability for assignment is along the x axis, and the y axis is the panel accuracy +/- sd. Each of the six hybrid classes is plotted in a separate facet.

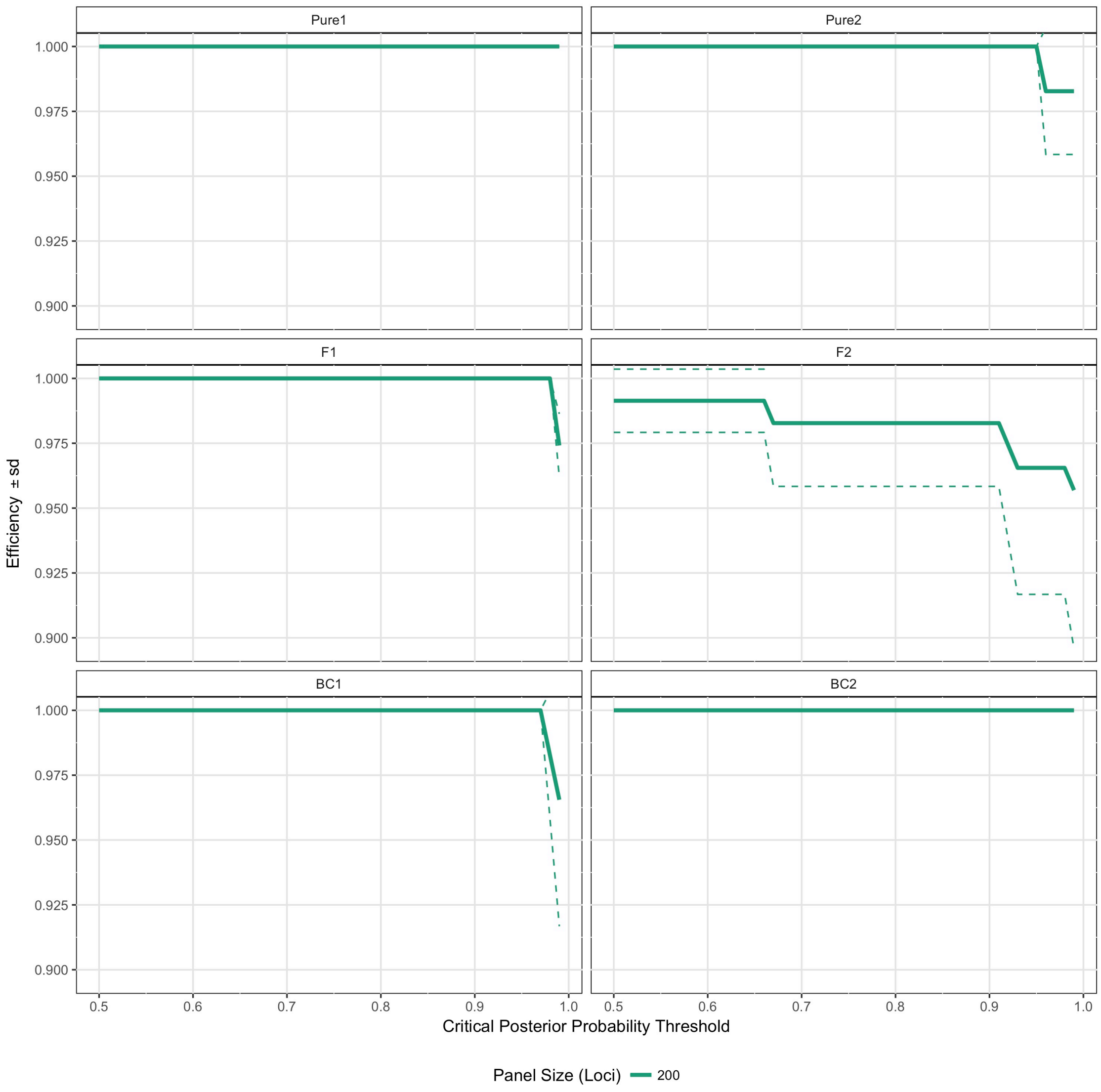

Figure 4. Example one of the efficiency plots produced by hybridPowerComp. The critical posterior probability for assignment is along the x axis, and the y axis is the panel accuracy +/- sd. Each of the six hybrid classes is plotted in a separate facet.

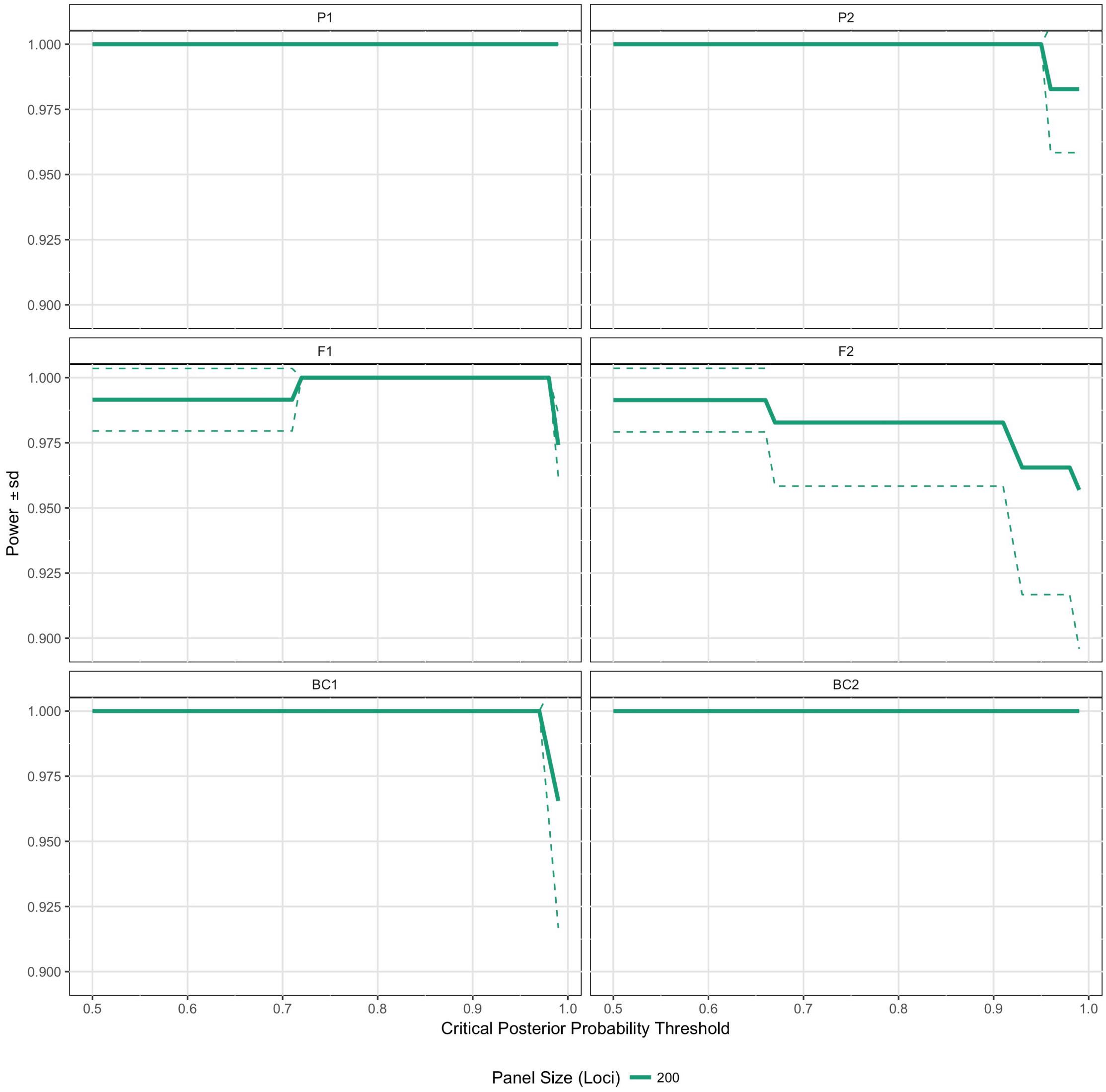

Figure 5. Example one of the power plots produced by hybridPowerComp. The critical posterior probability for assignment is along the x axis, and the y axis is the panel accuracy +/- sd. Each of the six hybrid classes is plotted in a separate facet.

Figures 3-5 show that the panel that was developed is has very good accuracy, efficiency and power. Figure 3 shows that at critical posterior probability thresholds between 0.5 and 1.0, of the individuals assigned to a given class, > 98% of them will have been assigned correctly. Similarly, Figure 4 indicates that > 95% of indivdiuals in each class will be identified at critical posterior probability thresholds between 0.5 and 0.9. Which results in power > 0.95 (unitless) at critical posterior probability thresholds between 0.5 and 0.9.

Step 6 - Combine the experimental data with simulated pure individuals

While NewHybrids does not require that known individuals or genotype frequencies of the populations in question be provided *a priori*, the inclusion of simulated pure individuals with experimental data does improve the NewHybrid's efficacy.

```
nh_analysis_generateR(ReferencePopsData = "~/Dropbox/DFO Aquaculture In
teraction/Word Documents/hybriddetective/hybriddetective_example/Exampl
eSNPdata_200_Loci_Panel_S1R1_NH_EX.txt", UnknownIndivs = "~/Dropbox/DFO
Aquaculture Interaction/Word Documents/hybriddetective/hybriddetective_
example/Top200CrabGenepop2.txt", output.name = "Hybriddetective_example
.txt")
```

Step 7 - Optional - Assign known hybrid class designations to simulated invidivals.

NewHybrids allows the user to assign indivudals to known hybrid classes. This option allows the program to more accurately model the expected genotype frequencies of the two (potentially) hybridizing populations, and thus increase its power for detecting hybridization. This process can easily be accomplished using the function *nh_Zcore*. In this case, we will use the options "z" and "s", along with numeric population designators (0 for pure population 1, 1 for pure population 2). Where "z" indicates that it is known beforehand that an individual belongs to a hybrid class, and "s" indicates that the indivdiual is to be used for the calculation of allele frequencies only. For more information, users are directed to the User's Guide to the Program NewHybrids Version 1.1. beta

```
example_zeds <- read.csv("Hybriddetective_Example_Zeds.csv", header = T
RUE, stringsAsFactors = FALSE)

knitr::kable(example_zeds[c(1:3, 61:64), ])
```

	Individual	Zscore
1	1	z0s
2	2	z0s
3	3	z0s
61	61	z1s
62	62	z1s
63	63	z1s
64	64	z1s

```
nh_Zcore(GetstheZdir = "~/Dropbox/DFO Aquaculture Interaction/Word Docu
ments/hybriddetective/hybriddetective_example/Hybriddetective_Example_G
iveZeds/", multiapplyZvec = "~/Dropbox/DFO Aquaculture Interaction/Word
Documents/hybriddetective/hybriddetective_example/Hybriddetective_Examp
le_Zeds.csv")
```

Step 8 - Visualize the output of NewHybrids

```
#nh_plotR("Example_NH_Results/")
```

This will produce a plot of the results

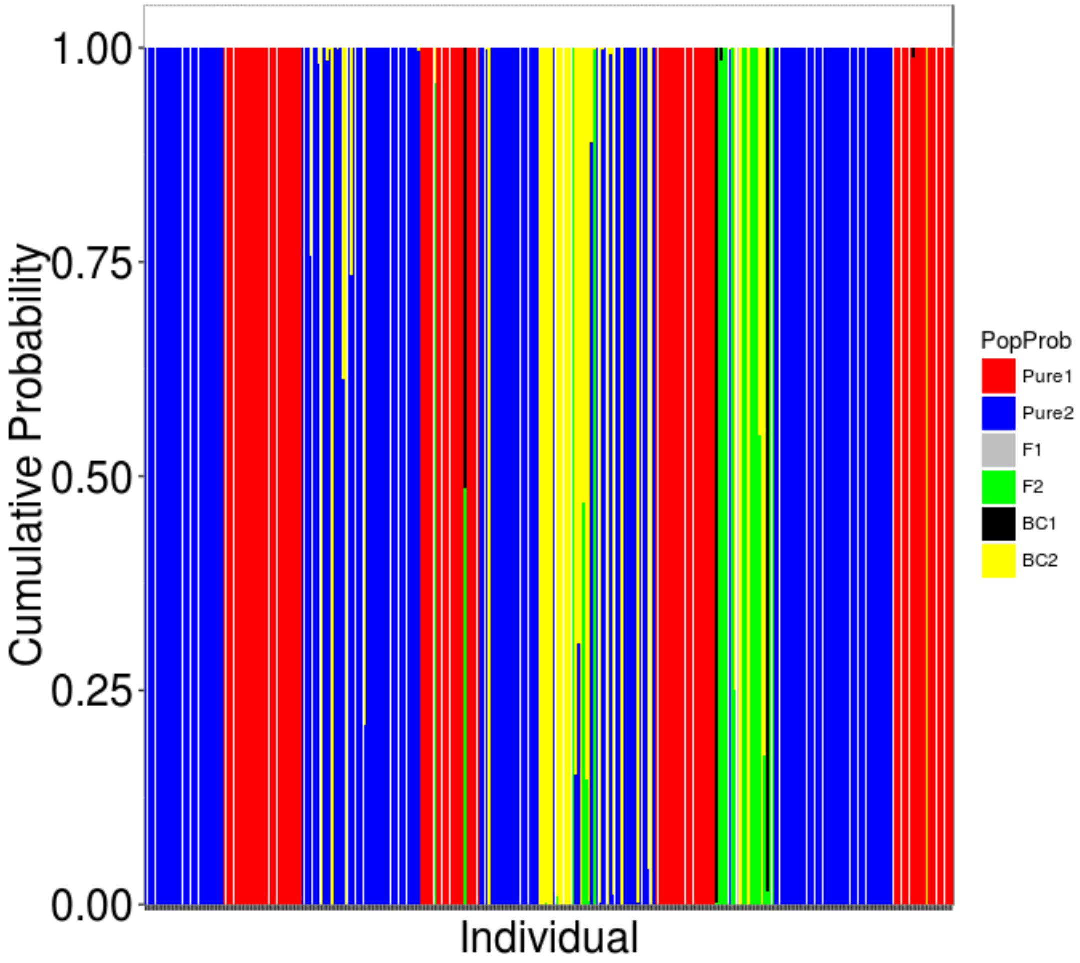

Figure 6. Example plot produced by nh_multiplotR from results of the NewHybrids anaysis conducted in Jeffery et al. 2017. Indivdiuals are along the x axis, and the y axis is the individual specific cumulative posterior probability of assignment to each of the six possible hybrid classes. The first solid blue and red groups are the simulated pure North and South individuals, and the following individuals are experimental samples. The hybrid classes are denoted by colour.

From here, the **NewHybrids** output can be analyzed however the user wishes.