File S1

PERL script to calculate the frequency of all SNPs in two matched pools

```
# Usage: cerevisiae_VCF_diff <vcf-file> <min-counts> <min-fraction>
# Input file must be a VCF file created from two samples (e.g., WT and mutant).
# When calling the function, the user must specify the minimum reads needed in
each sample
# (this study used 10) and the minimum fraction of an alternative allele
observed in either
# of the two samples (this study used 0.5).
# This program will ask which column is the WT and which column is the mutant.
# Output file is a tab-delimited file with same name as input file, plus the
date.
# Output file contains the allele descriptions and their counts.
# "FRACTION_DIFF_REF" shows the fraction difference of the reference allele: WT
minus mutant.
# The closer this value is to 1 (or -1), the greater the difference between WT
and mutant.

#!/usr/bin/perl
use warnings;
use strict;
use List::Util qw( min max );

my $input = shift;
my $date = ((localtime)[5]+1900)."_".((localtime)[4]+1)."_".((localtime)[3]);
my $output = $input.$date.".txt";

open IN, $input or die $!;
open OUT, ">$output" or die $!;
select OUT;

my $min_num = shift;
my $min_fraction = shift;

my
%chr=(chrI=>1,chrII=>2,chrIII=>3,chrIV=>4,chrV=>5,chrVI=>6,chrVII=>7,chrVIII=>8,

      chrIX=>9,chrX=>10,chrXI=>11,chrXII=>12,chrXIII=>13,chrXIV=>14,chrXV=>15,ch
rXVI=>16,chrMito=>17,"2-micron"=>18);
my %column;

$/="\n";

my $labels = join "\t", qw/CHROM POS REF_ALLELE ALT_ALLELE1 ALT_ALLELE2
ALT_ALLELE3 NUM_REF_WT NUM_ALT1_WT NUM_ALT2_WT NUM_ALT3_WT NUM_REF_MUT
NUM_ALT1_MUT NUM_ALT2_MUT NUM_ALT3_MUT FRACTION_DIFF_REF/;
print $labels,"\n";

my $WT_column = "-1";
my $mutant_column = "-1";

while (<IN>) {
      my @row = split /\t/, $_;
```

```
my @new_row;

#  Meta-information lines
if ($row[0] =~ /^##/) {next;}

# header line:  determine order of columns
elsif ($row[0] =~ /^#[A-Za-z0-9]/) {
      my $count=0;

      for (@row) {
            print STDOUT "$count: $_\n";
            $column{$_} = $count;
            $count++;
      }
      print STDOUT "Which column is the WT sample? ";
      chomp ($WT_column = <STDIN>);
      print STDOUT "Which column is the mutant sample? ";
      chomp ($mutant_column = <STDIN>);
      next;
}

# data line (i.e., with mutation)
else {
      $new_row[0] = $chr{$row[$column{"#CHROM"}]};
      $new_row[1] = $row[$column{POS}];
      $new_row[2] = $row[$column{REF}];

      my @MUTATIONS = split /,/, $row[$column{ALT}];
      if (!defined$MUTATIONS[0] || ($MUTATIONS[0] eq "."))
{$new_row[3] = "";}
      else {$new_row[3]=$MUTATIONS[0];}
      if (!defined$MUTATIONS[1] || ($MUTATIONS[1] eq "."))
{$new_row[4] = "";}
      else {$new_row[4]=$MUTATIONS[1];}
      if (!defined$MUTATIONS[2] || ($MUTATIONS[2] eq "."))
{$new_row[5] = "";}
      else {$new_row[5]=$MUTATIONS[2];}

      my %format_column;
      my @format = split /:/, $row[$column{FORMAT}];
      my $count=0;
      my $RA="RA"; my $AA="AA";
      for (@format) {
            $format_column{$_} = $count;
            if ($_ eq "RO")        { $RA = "RO"; }
            if ($_ eq "AO")        { $AA = "AO"; }
            $count++;
      }

      if ($row[$WT_column] =~ /:/) {
            my @WT = split /:/, $row[$WT_column];
            chomp (@WT);
            $new_row[6] = $WT[$format_column{$RA}];
            my @AA1 = split /,/, $WT[$format_column{$AA}];
```

```
                  if (!defined$AA1[0] || ($AA1[0] eq "."))      {$new_row[7] =
0;}
                  else {$new_row[7]=$AA1[0];}
                  if (!defined$AA1[1] || ($AA1[1] eq "."))      {$new_row[8] =
0;}
                  else {$new_row[8]=$AA1[1];}
                  if (!defined$AA1[2] || ($AA1[2] eq "."))      {$new_row[9] =
0;}
                  else {$new_row[9]=$AA1[2];}
            }
            else {
                  @new_row[6..9] = ("?","?","?","?");
            }

            if ($row[$mutant_column] =~ /:/) {
                  my @MUT = split /:/, $row[$mutant_column];
                  chomp (@MUT);
                  $new_row[10] = $MUT[$format_column{$RA}];
                  my @AA2 = split /,/, $MUT[$format_column{$AA}];
                  if (!defined$AA2[0] || ($AA2[0] eq "."))      {$new_row[11] =
0;}
                  else {$new_row[11]=$AA2[0];}
                  if (!defined$AA2[1] || ($AA2[1] eq "."))      {$new_row[12] =
0;}
                  else {$new_row[12]=$AA2[1];}
                  if (!defined$AA2[2] || ($AA2[2] eq "."))      {$new_row[13] =
0;}
                  else {$new_row[13]=$AA2[2];}
            }
            else {
                  @new_row[10..13] = ("?","?","?","?");
            }

      }

      my $WT_sum = 0;
      my $mut_sum = 0;

      for (@new_row[6..9]) {
            if ($_ ne "?") {$WT_sum=$WT_sum+$_;}
      }
      for (@new_row[10..13]) {
            if ($_ ne "?") {$mut_sum=$mut_sum+$_;}
      }

      if ( $WT_sum >= $min_num && $mut_sum >= $min_num ) {
            if ( ( ( max(@new_row[7..9]) / $WT_sum) >= $min_fraction) ||
            ( ( max(@new_row[11..13]) / $mut_sum) >= $min_fraction) ) {
                  $new_row[14] = ($new_row[6]/$WT_sum)-($new_row[10]/$mut_sum);
                  my $line = join "\t", @new_row;
                  print $line, "\n";
            }
      }

}
```