RUNNING AND PROGRAMMING DNASAM

ANDREW J. ECKERT AND JOHN D. LIECHTY

1. Introduction

The inspiration for creating `dnasam` arose from the need to be able to calculate population genetics statistics and do neutrality testing in an automated fashion for resequencing projects that contain DNA sequence alignments for hundreds or thousands of loci. `dnasam` is designed to utilize Richard Hudson's `ms` program to generate coalescent simulations, where the parameters for the simulations can be based on the observed data, on parameters supplied by the user, or some combination of both. After a statistic is calculated for the observed data, `dnasam` will, when appropriate, calculate the statistic under the user-specified model for the simulations. `dnasam` takes as input one or more multialigned fasta files (which can have no outgroup, a single outgroup sequence or multiple aligned sequences for an outgroup), and generates output that is in either a human-friendly form, or a tab-seperated table that is appropriate for importing into the program `R` (where undefined data values are assigned the `R`-friendly value of 'NA').

The intended audience for this document is both the end-user and the programmer. For the end-user, we'll provide the information you need to get `dnasam` up and running with, we hope, as little pain as possible. For the programmer, we will give a quick overview of how `dnasam` is structured and a tutorial on how to create a new class to incorporate an additional statistical test into `dnasam`. `dnasam` has been designed in a manner that makes it relatively straight-forward for a programmer with a modest amount of experience in object-oriented Perl to add additional statistical tests to the workflow. For both audiences, we have included in this document a list of the classes that perform the statistical calculations in `dnasam`, with references, in an effort to make details about the implementation of the various statistical calculations both accessible and transparent.

We welcome suggestions, bug reports and other feedback on `dnasam`. Please address email correspondences to `jdliechty@ucdavis.edu`.

Date: July 22, 2009.

2. Installation

These notes reflect installing `dnasam` on a UNIX-like operating system, including GNU/Linux and Apple's OS X. We'll add additional notes for installing `dnasam` on a Windows computer in the future. Note that installing some of the software mentioned below may require super-user access to your computer, and hence may require that you contact the IT-team at your facility.

First, if you want to confirm `perl` is installed on your computer, at the terminal prompt, type 'perl -version' (where `smith$` is just the command prompt for our hypothetical user 'smith'):

```
smith$ perl -version
```

For almost all UNIX-like operating systems, Perl will already be installed. If you get some message along the lines of 'command not found', it is probably because either Perl is not installed on your computer, or because the `perl` command could not be found using the directories indicated in your PATH environmental variable.

`dnasam` can be downloaded from:

`http://dendrome.ucdavis.edu/adept2/dnasam.html`

There are three options for downloading `dnasam`:

- `dnasam<version>_osx_x86.tar.gz`: This version includes a pre-compiled copy of `ms` for Apple's OS X running on x86 hardware (compiled in OS X 10.5 - it may not run on older versions of OS X).
- `dnasam<version>_linux_x86.tar.gz`: This version includes a pre-compiled copy of `ms` for GNU/Linux running on x86 hardware (verified to run in a Fedora Core 8 Linux environment).
- `dnasam<version>_source.tar.gz`: This file does not include either source code or a compiled binary for Richard Hudson's `ms` program. If you already have `ms` installed on your computer, you can install this package, and follow the instructions provided below to create a symbolic link to the copy of `ms` on your computer.

As a convenience, we provide a precompiled version of Richard Hudson's `ms` program in the `dnasam<version>_osx_x86.tar.gz` and `dnasam<version>_linux_x86.tar.gz` packages.

Next you'll need to use the `tar` command to unzip and untar the `dnasam` file. Move the `dnasam<version>.tar.gz` to the directory where you wish to install `dnasam` and issue the following command:

```
tar zxvf dnasam<version>.tar.gz
```

If you installed a version `dnasam` with a precompiled binary for `ms`, you should have everything you need to run the program, and you can skip ahead to the 'Running dnasam' section.

If you need to install Richard Hudson's `ms` program, the source code is available from:

`http://home.uchicago.edu/~rhudson1/source/mksamples.html`

Included in the folder with the source code for `ms` is a file titled 'readme' that includes instructions on how to compile the `ms` program using the Gnu `gcc` compiler.

The Gnu `gcc` compiler and Perl are freely available for most GNU/Linux distributions, but may need to be installed on your computer. (Installing `gcc` and Perl in various operating systems is beyond the scope of this document. If you have questions about installing `gcc` or Perl, please refer to the documentation for `gcc` or Perl for your operating system or to your local System Administrator.)

If you use Apple's OS X operating system and you don't have `gcc` intstalled on your computer, you can install Apple's Xcode development environment (which includes `gcc`). Xcode can be downloaded from:

`http://developer.apple.com/technology/Xcode.html`.

(Note you will need administrative privileges to install Xcode.)

Once you have compiled `ms`, or if `ms` is already installed on your computer, you'll need to note the path to `ms` on your computer so you can provide the path to `dnasam`. In this example, we'll use the path:

`/Users/smith/msdir/ms`.

Next, `cd` to the `dnasam/msdir` directory. You'll need to create a symbolic link to your installation of `ms` from this directory with the following command:

```
ln -s /Users/smith/msdir/ms ms
```

You can then verify that the link has been made with the `ls -l` command as follows:

```
smith\$ ls -l /Users/smith/dnasam/msdir
total 8
lrwxr-xr-x  1 smith  smith  23 Feb  9 19:37 ms -> /Users/smith/msdir/ms
```

In house, we have installed and run `dnasam` in OS X version 10.5.6 using the Xcode Development Environment version 3.1.2, and in Fedora Core Linux 8.

3. Running dnasam

You must specify one of three methods for importing FASTA alignments into `dnasam`:

`-input_file`: Specify the multifasta alignment to use as input.

`-input_dir`: Specify the directory containing the multifasta files you wish to run.

`-input_list`: Specify a list that contains the full path to each multifasta file you wish to run. This list also contains the number of outgroup sequences present in each alignment.

Optional command-line switches include:

`-outgroup_count`: The outgroup sequences must appear last in the alignment. Specify how many sequences at the end of the file are outgroup sequence. Default is `0`.

`-print`: Set to `table` for tab-separated table output. Set to `human` for human-readable output. Default is `human`.

`-output_file`: Specify the name to use for your `dnasam` output data.

`-ms`: Any additional options you wish to pass to `ms`.

`-simulate`: Allows you to specify `W` (for $\hat{\theta}_W$), `P` (for $\hat{\theta}_\pi$) or `S` (for S), calculated from the observed data in your alignment, as the θ or S parameter to use as input for `ms` for the simulations. (If you wish to use a specific value for θ or S for your `ms` simulations, instead of a value calculated from the observed data, you can pass that information to `ms` using the `-ms` option.) Default is $\hat{\theta}_W$. See examples below.

`-numsims`: The number of `ms` simulations to run. Default is 10000.

`-ms_path`: The path where the ms executable (or a symbolic link to it) can be found. Default is `'./msdir/ms'`.

`-sim_outdir`: The path to a directory where `dnasam` will store a file for each alignment containing the statistics calculated for each `ms` simulation performed for that alignment. Default is to not create this file.

`-island_pops`: Allows the user to pass information to `ms` that will be used with the `-I` flag for `ms` where the size of each subpopulation is expressed as a fraction of the total population size. `dnasam` then determines integers to use with the `-I` flag for `ms` based on the number of samples in the alignment. For example, if you specify `-island_pops=[.3,.5,.2]`, and you are running a batch of multiple files, for a file with $n = 10$ samples would lead to the parameter `-I 10 3 5 2` getting passed to `ms`. If the next file in the batch had a $n = 20$ samples, the command passed to `ms` would be `-I 20 6 10 4`. The integers values are calculated from the fractional values by calculating the fractional values for all but the largest population fraction, rounding up or down as expected, then the largest subpopulation sample size is generated by subtracting all these calculated integers from n.

`-island_mig_param`: Allows the user to specify the migration parameter that is used with the `-I` flag in `ms`.

3.1. **Some example commands for running `dnasam`.** (NB: The '\' character below indicates where we've had to wrap a command or output around to the next line for purpose of formatting printing. When entering one of these commands at the shell prompt, the entire command should be typed on a single line, omitting the '\' character.)

```
./dnasam.pl -input_file=./example_data/no_outgroup/cpk3.fas -numsims=5000
```

This gives the output:

```
For file ./example_data/no_outgroup/cpk3.fas (ms command:  \
'./msdir/ms 32 5000 -t 1.98646956213771'):
                          S:  8
                     Length:  630
                    Samples:  32
                     ThetaW:  1.986 (0.6050)
                   ThetaW_SD:  0.898
                     ThetaPi:  2.232 (0.6598)
                   ThetaPi_SD:  1.405
                     ThetaNs:  0.969 (0.4406)
                   Tajima's D:  0.370 (0.6794)
                           F*:  0.681 (0.7261)
                           D*:  0.677 (0.7441)
                           R2:  0.137 (0.6745)
                           R3:  -0.138 (0.3144)
                           R4:  0.138 (0.6214)
                           Ch:  1.036 (0.4513)
                          Zns:  0.178 (0.5232)
                    Dns_star:  1.311 (0.4874)
                    Zns_star:  -0.133 (0.4880)
               Zns_star_star:  0.136 (0.4897)
                           Za:  0.216 (0.5982)
                           ZZ:  0.038 (0.7240)
                            B:  0.000 (0.4954)
                            Q:  0.000 (0.4954)
                            h:  7.000 (0.7664)
                           Hd:  0.780 (0.7375)
                        Hd_SD:  0.052
                    ThetaHom:  0.282 (0.2699)
                  ThetaHom_SD:  0.064
           ThetaHomCorrected:  0.210 (0.2699)
        ThetaHomCorrected_SD:  0.064
                            F:  0.244 (0.2694)
                       ThetaK:  2.465 (0.7002)
                     Fu's Fs:  0.007 (0.5361)
       Non-empty simulations:  4990
-------------
Total run time:  1.15846213333333 minutes.
```

This is the default 'human' output. Note the actual command used to generate the `ms` simulations is reported at the top, and note the `-t 1.98646956213771` value for θ corresponds to $\hat{\theta}_W$. Also keep in mind that some of these numbers are based on random simulations, so the output you see might have different values.

Note that next to each summary statistic, where we also calculated the same statistic for the `ms` simulations, we have a p-value in parentheses. This p-value is the frequency of how often the summary statistic calculated for the simulations under the given model was less than or equal to the summary statistic for our observed data.

If we wanted to analyse the same data as above, but using $\hat{\theta}_\pi$ as our θ value to pass to `ms`, we could issue the following command:

```
./dnasam.pl -input_file=./example_data/no_outgroup/cpk3.fas -numsims=20000 \
-simulate=P
```

Which gives the following output:

```
For file ./example_data/no_outgroup/cpk3.fas (ms command:  \
'./msdir/ms 32 20000 -t 2.23185483870968'):
                        S:  8
                   Length:  630
                  Samples:  32
                   ThetaW:  1.986 (0.5057)
                ThetaW_SD:  0.898
                  ThetaPi:  2.232 (0.5897)
               ThetaPi_SD:  1.405
                  ThetaNs:  0.969 (0.3877)
               Tajima's D:  0.370 (0.6852)
                       F*:  0.681 (0.7364)
                       D*:  0.677 (0.7551)
                       R2:  0.137 (0.6868)
                       R3:  -0.138 (0.3057)
                       R4:  0.138 (0.6390)
                       Ch:  1.036 (0.4533)
                      Zns:  0.178 (0.5118)
                 Dns_star:  1.311 (0.4702)
                 Zns_star:  -0.133 (0.5065)
            Zns_star_star:  0.136 (0.5012)
                       Za:  0.216 (0.5816)
                       ZZ:  0.038 (0.7192)
                        B:  0.000 (0.4498)
                        Q:  0.000 (0.4498)
                        h:  7.000 (0.6864)
                       Hd:  0.780 (0.6804)
                   Hd_SD:  0.052
                ThetaHom:  0.282 (0.3258)
```

```
               ThetaHom_SD:  0.064
         ThetaHomCorrected:  0.210 (0.3258)
      ThetaHomCorrected_SD:  0.064
                         F:  0.244 (0.3253)
                    ThetaK:  2.465 (0.6077)
                  Fu's Fs:  0.007 (0.5360)
     Non-empty simulations:  19975
-------------
Total run time:  5.10892298333333 minutes.
```

and we can see that the value of $\hat{\theta}_\pi$ that was determined from the observed data was passed as the parameter for θ to `ms`. Also note how increasing the number of simulations increases the time required to run the program.

Below shows how `dnasam` can be called for an alignment with a single outgroup (where the outgroup appears last in the multifasta file):

```
./dnasam.pl -input_file=./example_data/ferritin_outgroup.fas \
-numsims=20000 -outgroup_count=1
```

And this results in the following output:

```
For file ./example_data/ferritin_outgroup.fas (ms command:  \
'./msdir/ms 32 20000 -t 1.7381608668705', \
with outgroup: './msdir/ms 32 20000 -t 1.7381608668705'):
                          S:  7
                     Length:  613
                    Samples:  32
                     ThetaW:  1.738 (0.6073)
                  ThetaW_SD:  0.816
                    ThetaPi:  0.764 (0.1858)
                 ThetaPi_SD:  0.640
     Outgroup sample count:  1
          S(with outgroup):  7
     ThetaW(with outgroup):  1.738 (0.6150)
  ThetaW_SD(with outgroup):  0.816
    ThetaPi(with outgroup):  0.764 (0.1897)
 ThetaPi_SD(with outgroup):  0.640
                Divergence:  20.406
                   ThetaNs:  4.844 (0.9739)
                Tajima's D:  -1.631 (0.0296)
                        F*:  -2.422 (0.0227)
                        D*:  -2.268 (0.0327)
                   ThetaNe:  5.000 (0.9807)
                    ThetaH:  0.075 (0.0931)
        ThetaH_outgroup_SD:  1.648
                    ThetaL:  0.419 (0.1284)
        ThetaL_outgroup_SD:  1.236
```

```
                          H:  0.690 (0.6992)
                      normH:  0.498 (0.6208)
                      normE:  -1.637 (0.0088)
                F(outgroup):  -2.630 (0.0137)
                D(outgroup):  -2.482 (0.0167)
                         R2:  0.079 (0.1004)
                         R3:  0.086 (0.9267)
                         R4:  0.117 (0.3919)
                         Ch:  25.339 (0.9722)
                        R2e:  0.079 (0.1109)
                        R3e:  0.086 (0.9388)
                        R4e:  0.117 (0.4049)
                        Che:  25.339 (0.9798)
                        Zns:  0.120 (0.3495)
                   Dns_star:  0.226 (0.1750)
                   Zns_star:  0.894 (0.8166)
              Zns_star_star:  0.531 (0.8880)
                         Za:  0.040 (0.2355)
                         ZZ:  -0.080 (0.1613)
                          B:  0.000 (0.5200)
                          Q:  0.000 (0.5200)
                          h:  6.000 (0.6840)
                         Hd:  0.516 (0.2067)
                     Hd_SD:  0.100
                   ThetaHom:  0.938 (0.7954)
                ThetaHom_SD:  0.285
          ThetaHomCorrected:  0.699 (0.7954)
       ThetaHomCorrected_SD:  0.282
                          F:  0.500 (0.7923)
                     ThetaK:  1.905 (0.5317)
                   Fu's Fs:  -2.398 (0.0933)
       FixedDifferenceCount:  20.000
    SharedPolymorphismCount:  NA
      MultiAllelicSiteCount:  0.000
      Non-empty simulations:  19920
-------------
Total run time:  5.45669165 minutes.
```

Note that additional summary statistics are calculated when an outgroup is present. Also note that `ms` is called twice: Once for the ingroup data, ignoring information in the outgroup; and a second time, where we ignore sites in the ingroup where there is no called site for the outgroup. In this second case, we calcutate values of $\hat{\theta}_W$, $\hat{\theta}_\pi$ and S, and use these values as the parameters that get passed to `ms`. This second set of simulations is then used for determining the p-values for the summary statistics that require an outgroup.

If you wish to run `dnasam` on a file with multiple outgroup sequences, the outgroup sequence must appear last in the file, and you must specify how many outgroup sequences there are:

```
./dnasam.pl -input_file=./example_data/multiOutgroup.fasta \
-outgroup_count=17
```

and this gives the output:

```
For file ./example_data/multiOutgroup.fasta (ms command:  \
'./msdir/ms 11 10000 -t 2.04850291288443', \
with outgroup: './msdir/ms 11 10000 -t 2.04850291288443'):
                          S:  6
                     Length:  673
                    Samples:  11
                     ThetaW:  2.049 (0.6189)
                  ThetaW_SD:  1.111
                     ThetaPi:  1.236 (0.3166)
                  ThetaPi_SD:  0.955
     Outgroup sample count:  17
          S(with outgroup):  6
     ThetaW(with outgroup):  2.049 (0.6126)
  ThetaW_SD(with outgroup):  1.111
    ThetaPi(with outgroup):  1.236 (0.3235)
 ThetaPi_SD(with outgroup):  0.955
                Divergence:  14.364
                   ThetaNs:  4.545 (0.9262)
                Tajima's D:  -1.569 (0.0566)
                        F*:  -1.811 (0.0653)
                        D*:  -1.617 (0.0900)
                   ThetaNe:  2.000 (0.6746)
                    ThetaH:  15.782 (0.9996)
        ThetaH_outgroup_SD:  1.550
                    ThetaL:  9.200 (0.9977)
        ThetaL_outgroup_SD:  1.342
                         H:  -14.545 (0.0002)
                     normH:  -11.115 (0.0000)
                     normE:  9.431 (1.0000)
              F(outgroup):  -0.463 (0.3631)
              D(outgroup):  0.034 (0.5366)
                       R2:  0.167 (0.4418)
                       R3:  0.181 (0.8839)
                       R4:  0.226 (0.7912)
                       Ch:  12.598 (0.9390)
                      R2e:  0.120 (0.0669)
                      R3e:  0.049 (0.8169)
                      R4e:  0.138 (0.1152)
```

```
                      Che:  12.598 (0.9683)
                      Zns:  0.421 (0.7043)
                 Dns_star:  0.821 (0.3239)
                 Zns_star:  0.600 (0.7022)
            Zns_star_star:  0.513 (0.8345)
                       Za:  0.780 (0.9316)
                       ZZ:  0.359 (0.9999)
                        B:  0.600 (0.8959)
                        Q:  0.433 (0.9320)
                        h:  3.000 (0.3032)
                       Hd:  0.345 (0.0607)
                    Hd_SD:  0.172
                 ThetaHom:  1.895 (0.9510)
              ThetaHom_SD:  1.167
         ThetaHomCorrected:  1.436 (0.9510)
      ThetaHomCorrected_SD:  1.136
                        F:  0.686 (0.9369)
                   ThetaK:  0.986 (0.2848)
                 Fu's Fs:  1.023 (0.7396)
     FixedDifferenceCount:  3.000
   SharedPolymorphismCount:  0.000
     MultiAllelicSiteCount:  1.000
     Non-empty simulations:  9852
-------------
Total run time:  0.875972316666667 minutes.
```

Note that when the number of simulations to perform is not specified, the default number of simulations is 10000.

You can have `dnasam` run all the files in a given directory by using the `-input_dir` switch, with the caveat that arguments used with `dnasam` must apply to each file (so each file must have the same number of outgroup sequence):

```
./dnasam.pl -input_dir=./example_data/no_outgroup/ -numsims=5000
```

For cases where you have a collection of multifasta files you wish to run, and those files have different numbers of outgroup sequences present, you can create a file that lists each file, then a tab, then the number of outgroup sequence present (and again, outgroup sequence must appear at the end of the file), then run `dnasam` with the `-input_list` switch. For example, the file:

```
./example_data/input_list_test.txt
```

contains the following information:

```
# Files with 0, 1 or multiple outgroups...
./example_data/no_outgroup/ug-2_498.fas 0
./example_data/ferritin_outgroup.fas    1
```

```
./example_data/multiOutgroup.fasta      17
```

And you can then run the following command to process those three files as a batch:

```
./dnasam.pl -input_list=./example_data/input_list_test.txt
```

You can use the `-print` switch to instruct `dnasam` to print the output in a tab-delimited table that can be imported into a spreadsheet program or a statistical analysis program like `R`. In the command below, we also use the `-output_file` switch to instruct `dnasam` to output the data to the file indicated:

```
./dnasam.pl -input_list=./example_data/input_list_test.txt -print=table \
-output_file='test.out'
```

Using the `-ms` switch, you can pass parameters directly to `ms`. For example, in the command below we have `ms` generate the simulations using a value of 5 for θ, rather than $\hat{\theta}_W$ or $\hat{\theta}_\pi$ as determined from our observed data:

```
./dnasam.pl -input_file=./example_data/ferritin_outgroup.fas \
-numsims=10000 -outgroup_count=1 -ms='-t 5'
```

Which gives the output:

```
For file ./example_data/ferritin_outgroup.fas (ms command:  \
'./msdir/ms 32 10000 -t 5', with outgroup: './msdir/ms 32 10000 -t 5'):
                          S:  7
                     Length:  613
                    Samples:  32
                     ThetaW:  1.738 (0.0181)
                  ThetaW_SD:  0.816
                    ThetaPi:  0.764 (0.0027)
                 ThetaPi_SD:  0.640
     Outgroup sample count:  1
          S(with outgroup):  7
     ThetaW(with outgroup):  1.738 (0.0177)
  ThetaW_SD(with outgroup):  0.816
    ThetaPi(with outgroup):  0.764 (0.0030)
 ThetaPi_SD(with outgroup):  0.640
                Divergence:  20.406
                   ThetaNs:  4.844 (0.6260)
                Tajima's D:  -1.631 (0.0314)
                        F*:  -2.422 (0.0280)
                        D*:  -2.268 (0.0316)
                   ThetaNe:  5.000 (0.6302)
                    ThetaH:  0.075 (0.0023)
        ThetaH_outgroup_SD:  1.648
                    ThetaL:  0.419 (0.0023)
        ThetaL_outgroup_SD:  1.236
```

```
                          H:  0.690 (0.4225)
                      normH:  0.498 (0.5923)
                      normE:  -1.637 (0.0240)
                F(outgroup):  -2.630 (0.0160)
                D(outgroup):  -2.482 (0.0161)
                         R2:  0.079 (0.1056)
                         R3:  0.086 (0.9764)
                         R4:  0.117 (0.4888)
                         Ch:  25.339 (0.9586)
                        R2e:  0.079 (0.1211)
                        R3e:  0.086 (0.9920)
                        R4e:  0.117 (0.5080)
                        Che:  25.339 (0.9715)
                        Zns:  0.120 (0.2405)
                   Dns_star:  0.226 (0.0415)
                   Zns_star:  0.894 (0.9668)
              Zns_star_star:  0.531 (0.9764)
                         Za:  0.040 (0.0671)
                         ZZ:  -0.080 (0.0634)
                          B:  0.000 (0.2174)
                          Q:  0.000 (0.2174)
                          h:  6.000 (0.0420)
                         Hd:  0.516 (0.0077)
                     Hd_SD:  0.100
                   ThetaHom:  0.938 (0.9925)
                ThetaHom_SD:  0.285
          ThetaHomCorrected:  0.699 (0.9925)
       ThetaHomCorrected_SD:  0.282
                          F:  0.500 (0.9925)
                     ThetaK:  1.905 (0.0165)
                   Fu's Fs:  -2.398 (0.1762)
       FixedDifferenceCount:  20.000
    SharedPolymorphismCount:  NA
      MultiAllelicSiteCount:  0.000
      Non-empty simulations:  10000
-------------
Total run time:  6.61269868333333 minutes.
```

And we can see from the line with the commands used for `ms` that $\theta = 5$ was used as the parameter for the simulations.

`dnasam` can also be used to generate parameters to pass to `ms` where, for each simulation, a new value is used, where that value is pulled from a uniform distribution over a user-specified range. To do this, in the `-ms` parameter line, in place of the parameter value, you specify `tbs[a,b]`, where `a` and `b` and the minimum and maximum values of the uniform distribution. For example:

```
./dnasam.pl -input_file=./example_data/ferritin_outgroup.fas \
-numsims=10000 -outgroup_count=1 -ms='-t tbs[3,5] -r tbs[6,8.5] 800'
```

will run 10000 `ms` simulations, but for each simulation, a different value for `-t` between 3 and 5 is used. Similarly, for the first value for the `-r` switch for `ms`, we use values between 6 and 8.5, uniformly distributed. The second value for the `-r` switch, 800, is the length of our alignment in the model. This command gives the following output:

```
For file ./example_data/ferritin_outgroup.fas (ms command:  \
'./msdir/ms 32 10000 -t tbs -r tbs 800 <tbs.table', with outgroup: \
'./msdir/ms 32 10000 -t tbs -r tbs 800 <tbs.table'):
                          S:  7
                     Length:  613
                    Samples:  32
                     ThetaW:  1.738 (0.0476)
                  ThetaW_SD:  0.816
                    ThetaPi:  0.764 (0.0069)
                 ThetaPi_SD:  0.640
     Outgroup sample count:  1
          S(with outgroup):  7
     ThetaW(with outgroup):  1.738 (0.0440)
  ThetaW_SD(with outgroup):  0.816
    ThetaPi(with outgroup):  0.764 (0.0071)
 ThetaPi_SD(with outgroup):  0.640
                Divergence:  20.406
                   ThetaNs:  4.844 (0.7531)
                Tajima's D:  -1.631 (0.0147)
                        F*:  -2.422 (0.0109)
                        D*:  -2.268 (0.0123)
                   ThetaNe:  5.000 (0.7684)
                    ThetaH:  0.075 (0.0040)
        ThetaH_outgroup_SD:  1.648
                    ThetaL:  0.419 (0.0048)
        ThetaL_outgroup_SD:  1.236
                         H:  0.690 (0.5592)
                     normH:  0.498 (0.7022)
                     normE:  -1.637 (0.0122)
              F(outgroup):  -2.630 (0.0085)
              D(outgroup):  -2.482 (0.0091)
                       R2:  0.079 (0.0766)
                       R3:  0.086 (0.9927)
                       R4:  0.117 (0.4506)
                       Ch:  25.339 (0.9839)
                      R2e:  0.079 (0.0831)
                      R3e:  0.086 (0.9970)
                      R4e:  0.117 (0.4590)
                      Che:  25.339 (0.9886)
```

```
                      Zns:  0.120 (0.3948)
                 Dns_star:  0.226 (0.0315)
                 Zns_star:  0.894 (0.9767)
            Zns_star_star:  0.531 (0.9864)
                       Za:  0.040 (0.0710)
                       ZZ:  -0.080 (0.0194)
                        B:  0.000 (0.2702)
                        Q:  0.000 (0.2702)
                        h:  6.000 (0.0355)
                       Hd:  0.516 (0.0095)
                   Hd_SD:  0.100
                 ThetaHom:  0.938 (0.9907)
              ThetaHom_SD:  0.285
         ThetaHomCorrected:  0.699 (0.9907)
      ThetaHomCorrected_SD:  0.282
                        F:  0.500 (0.9906)
                   ThetaK:  1.905 (0.0191)
                 Fu's Fs:  -2.398 (0.3712)
      FixedDifferenceCount:  20.000
   SharedPolymorphismCount:  NA
     MultiAllelicSiteCount:  0.000
     Non-empty simulations:  9999
-------------
Total run time:  5.42550493333333 minutes.
```

If instead of 800 we wanted to use the length of an alignment in our observed data as the second parameter to `-r`, we can place the term `SEQLEN` in the position of that second parameter, and `dnasam` will use the length of the alignment for that parameter when `ms` is run. So:

```
./dnasam.pl -input_file=./example_data/ferritin_outgroup.fas \
-numsims=10000 -outgroup_count=1 -ms='-t tbs[3,5] -r tbs[6,8.5] SEQLEN'
```

gives the following output:

```
For file ./example_data/ferritin_outgroup.fas (ms command:  \
'./msdir/ms 32 10000 -t tbs -r tbs 613 <tbs.table', with outgroup: \
'./msdir/ms 32 10000 -t tbs -r tbs 613 <tbs.table'):
                        S:  7
                   Length:  613
                  Samples:  32
                   ThetaW:  1.738 (0.0484)
                ThetaW_SD:  0.816
                  ThetaPi:  0.764 (0.0079)
               ThetaPi_SD:  0.640
    Outgroup sample count:  1
         S(with outgroup):  7
```

```
    ThetaW(with outgroup):  1.738 (0.0477)
 ThetaW_SD(with outgroup):  0.816
   ThetaPi(with outgroup):  0.764 (0.0063)
ThetaPi_SD(with outgroup):  0.640
               Divergence:  20.406
                  ThetaNs:  4.844 (0.7447)
               Tajima's D:  -1.631 (0.0139)
                       F*:  -2.422 (0.0108)
                       D*:  -2.268 (0.0132)
                  ThetaNe:  5.000 (0.7607)
                   ThetaH:  0.075 (0.0049)
       ThetaH_outgroup_SD:  1.648
                   ThetaL:  0.419 (0.0055)
       ThetaL_outgroup_SD:  1.236
                        H:  0.690 (0.5615)
                   normH:  0.498 (0.7020)
                   normE:  -1.637 (0.0129)
             F(outgroup):  -2.630 (0.0067)
             D(outgroup):  -2.482 (0.0077)
                       R2:  0.079 (0.0781)
                       R3:  0.086 (0.9902)
                       R4:  0.117 (0.4527)
                       Ch:  25.339 (0.9828)
                      R2e:  0.079 (0.0846)
                      R3e:  0.086 (0.9957)
                      R4e:  0.117 (0.4624)
                      Che:  25.339 (0.9892)
                      Zns:  0.120 (0.4015)
                 Dns_star:  0.226 (0.0320)
                 Zns_star:  0.894 (0.9750)
           Zns_star_star:  0.531 (0.9849)
                       Za:  0.040 (0.0683)
                       ZZ:  -0.080 (0.0218)
                        B:  0.000 (0.2699)
                        Q:  0.000 (0.2699)
                        h:  6.000 (0.0366)
                       Hd:  0.516 (0.0100)
                    Hd_SD:  0.100
                 ThetaHom:  0.938 (0.9904)
              ThetaHom_SD:  0.285
        ThetaHomCorrected:  0.699 (0.9904)
    ThetaHomCorrected_SD:  0.282
                        F:  0.500 (0.9903)
                   ThetaK:  1.905 (0.0196)
                Fu's Fs:  -2.398 (0.3732)
    FixedDifferenceCount:  20.000
```

```
   SharedPolymorphismCount:  NA
     MultiAllelicSiteCount:  0.000
     Non-empty simulations:  9999
-------------
Total run time:  5.4315848 minutes.
```

Additional examples of how to run `dnasam` on example alignment files that were included with the distribution can be found in the `ExampleCommands.txt` file.

3.2. `ms` **and its options.** The program ms is used to simulate samples under the coalescent. It is written by Richard Hudson and is the standard program for simulating data under user-specified models. Hudson's website includes a well written manual for ms, so we will not cover all aspects of its abilities; however, a brief introduction is warranted. Please note that a full description is beyond the scope of this manual. The ms documentation is 21 pages of clear explanation that should be consulted.

The basic call to ms is structured in the following format:

`ms <number of samples> <number of simulations> <value of` θ *(not per site)*`>`

In `dnasam`, we set the number of samples to that in the original data file(s) and use our own flag to set the number of simulations. So, if you would like to use the estimate of theta from the original data and you want simulations under the standard neutral coalescent, you do not need to specify anything for the `ms` options called by `dnasam`.

If you would like to specify demography, recombination, gene conversion, population structure and/or population structure you will need to use the ms options. The following is a brief summary of the types of ms options for each of these processes. Definitions of these quantities can be found in the `ms` manual. The list is meant as a guide with which to search the `ms` manual for more details (start with pgs. 19 and 20)

Population size changes, population structure and lineage divergence: `-eG, -eg, -eN, -en, -eM, -em, -ema, -es, -ej, -I`

Recombination and gene conversion: `-r, -c`

Other: `-seed`, `-F` (`-F` sets the minimum absolute frequency of the minor allele at polymorphic sites)

The following is an example of a complex demographic scenario modeled for a sample of 624 individuals simulated using ms:

```
-I 2 565 59 -n 2 0.25 -g 1 2122.0320  -g 2 84.3452 \
-ema 0 2 x 50 200 x -eg 0.00113 1 0.0 -eg 0.00113 2 0.0 \
-ema 0.00113 2 x 44 4 x -ej 0.00606 2 1 -eN 0.00606 1.25 \
-seed 123 234 566.
```

Divergence between lineages was set to 97,000 yrs ago and was associated with a bottleneck. The strength of the bottleneck summed across both lineages was 20% of the ancestral effective population size, with the effective size of the western lineage being 10 times the size of the effective size of the eastern lineage. Genes were shared between these lineages at a rate equal to $N_{e0}m = 1$ during the bottleneck. Emergence from bottlenecks was set to 18,000 yrs ago and was associated with exponential growth and increased levels of gene flow until the present. Gene flow was coupled with population growth, so the value of $N_{e0}m$ changed each generation. The values of $N_{e0}m$ were set so that the value at time zero (*i.e.* currently) was 12.5. Values of N_em were adjusted when necessary to take into account differences in population sizes with respect to N_{e0}. The current effective size in each lineage was assumed to be 20,000 and 80,000. Time was measured in units of N_{e0} generations, with N_{e0} assumed to be 80,000. This model is pictured in Figure 1:

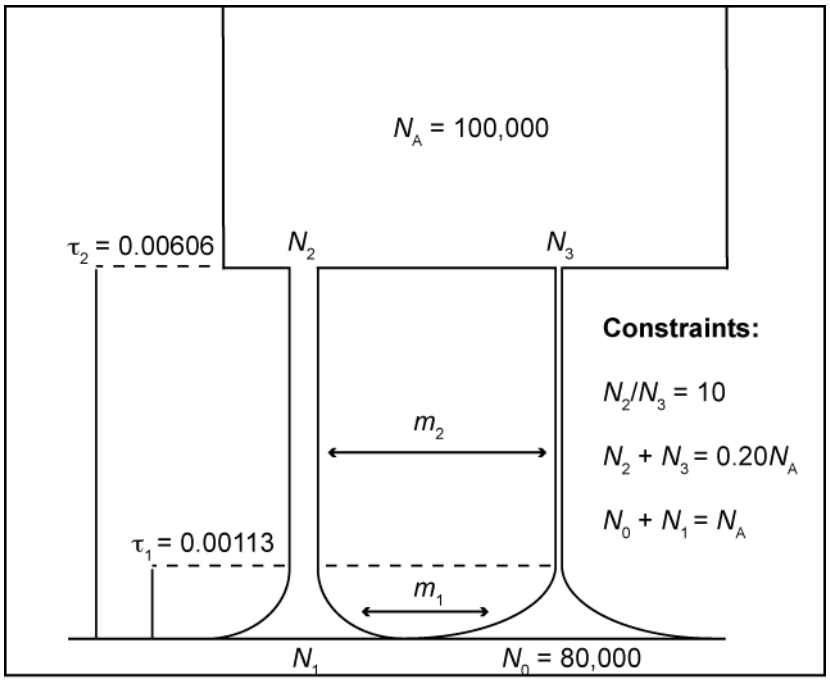

Figure 1. An illustration of the demographic model executed by the ms command given above.

To execute this in `dnasam` you would need a data file with 624 samples, place quotation marks around this command and place it after `-ms` in the command line. Examples given in section 3.1 illustrate some of these options.

4. The dnasam.pl script and core packages

4.1. **dnasam.pl.** `dnasam.pl` is the script the end-user will interact with. The `dnasam.pl` script handles command line switches, determining which input files are to be run, running each input file, and printing the results to either the screen or an output file.

4.2. **DS_BasicAnalysis.pm.** Most of the underlying functionality of `dnasam` is contained in the `DS_BasicAnalysis.pm` package. For each alignment, we instantiate a new `DS_BasicAnalysis.pm` object. `DS_BasicAnalysis.pm` will generate and hold the data structures that will be used by the various analysis packages as well as generate some of the common basic statistics. Then, for each alignment, `DS_BasicAnalysis.pm` will also run `ms`, passing along the `ms` switches that were specified at the command line, parse the `ms` output file, and for each simulation, call each additional analysis package on our list to have each analysis package calculate the statistic for the current simulation.

Calculations for $\hat{\theta}_W$ and $\hat{\theta}_\pi$ are handled by the `DS_BasicAnalysis.pm` package. In general, estimators for θ will be set equal to zero when S, the number of segregating sites, equals zero, whereas other estimators are often set to 'NA' when S is zero. For more information on how each estimator handles the $S = 0$ case, see the documentation for the package that creates that estimator.

For the following terms:

$$A_n = \sum_{i=1}^{n-1} \frac{1}{i}$$

$$B_n = \sum_{i=1}^{n-1} \frac{1}{i^2}$$

note that A_n, B_n are defined when $n \geq 2$.

For sample size n, for calculating values of $\hat{\theta}_W = S/A_n$:

$$(n < 2) \Rightarrow \hat{\theta}_W \text{ is set to 'NA'}$$

$$(n \geq 2) \text{ and } (S = 0) \Rightarrow \hat{\theta}_W \text{ is set to zero.}$$

And similarly for $\hat{\theta}_\pi$:

$$(n < 2) \Rightarrow \hat{\theta}_\pi \text{ is set to 'NA'}$$

$$(n \geq 2) \text{ and } (S = 0) \Rightarrow \hat{\theta}_\pi \text{ is set to zero.}$$

We'll encounter the following terms later in the manual:

Definition 1. *Assume we have n ingroup sequences in our alignment and that it is aligned to a single outgroup sequence. Let S be the set of biallelic segregating sites in the ingroup where one of the two alleles matches the base in the outgroup. For each site that is a member of S, we'll assume that the allele that matches the outgroup is the* ***ancestral allele*** *and the allele that does not match the outgroup is the* ***mutant allele****.*

Let ξ_i be the count of of sites in S that have i mutant alleles. Then the set of ξ_i for $1 \leq i \leq (n-1)$ is defined as the ***unfolded site frequency spectrum****.*

Definition 2. *Assume we have n sequences in our alignment. Then at a biallelic segregating site, a base is a* ***singleton*** *if that allele only occurs once at the site, with the other allele occurring $n-1$ times.*

Definition 3. *Assume we have an alignment with n ingroup sequences and a single outgroup sequence. Let S be the set of biallelic segregating sites in the ingroup where one of the two alleles matches the base in the outgroup. Then a base in the ingroup of our alignment is an* ***external mutant*** *if the following three conditions are met:*

(1) *It occurs in a site that is in S.*

(2) *It is a singleton in the ingroup.*

(3) *The allele of the base does not match the allele of the outgroup (i.e. the other $(n-1)$ bases at that site match the allele for the outgroup).*

Note that the definitions of external mutant and unfolded site frequency spectrum do require an outgroup. Note that for the definition of singleton does not require an outgroup.

4.3. **DS_Functions.pm.** Additional useful functions.

5. Additional included analysis packages

Each of the additional statistical analysis packages will have four properties inherent to that analysis that will be of interest to the end-user:

`_title`: The title that will be displayed as the first entry of the column that contains the statistic generated by the class in the tab-delimited output file that will be generated when the `-output=table` option is selected. This file can then be loaded into the program `R` or your favorite spreadsheet program, where the first row will contain the column titles, and each of the following rows will contain statistics generated for a given alignment/loci.

`_human_title`: The title that will be displayed when the `-output=human` option is selected.

_outgroup_required: When set to 0 (zero), it indicates this calculation is to be performed regardless of whether or not an outgroup is present, when set to 1, it indicates this calculation is performed only in the case where an outgroup is present.

_perform_sims: When set to 0 (zero), it indicates that this statistic will not be calculated for the simulated polymorphisms generated by `ms`, and hence no p-value will be reported for this class. When set to 1 (one), for each set of simulated polymorphisms returned by `ms`, if the sample meets the criteria for the calculation (*i.e.* the minimum number of segregating sites are present that are required to perform the calculation), then the calculation is performed for that set of simulated data and the results are included in the distribution that is used to determine the p-value for the observed statistic.

An advantage to making source code for a program available (as is the case with `dnasam`) is that the details of how each statistic is generated are accessible, whereas closed-source programs will inevitably have some degree of a black-box quality to them. However, this isn't of much comfort to an end-user who has neither the time or the desire to read through the source code. Our goal in the notes that follow on the various additional statistics packages is that, for each class, an end-user will be able to divine all the important mathematical details about how each statistical calculation is implemented.

For the statistics we are dealing with, too often the same name is associated with various formulas in the literature, where the formulas differ possibly due to an introduced improvement or the desire to include additional parameters. Sadly, it is also true that different names appear in the literature for an otherwise identical statistic. When we are aware of these situations, we'll include our observations about it in the notes for the relevant class.

A programmer will, of course, also find this information useful. The information in this section, along information in the following sections, will facilitate the process for a programmer customizing included packages as well as creating new analysis packages to incorporate into `dnasam`.

5.1. **DS_ThetaNs.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaNs`
- `_data_tag = ThetaNs`
- `_outgroup_required = 0`
- `_perform_sims = 1`

Let n denote the number of sequences and η_s denotes the count of all the singletons in the alignment for the locus. Then as an estimator for θ, we calculate:

$$\hat{\theta}_{Ns} = \eta_s \left(\frac{n-1}{n} \right).$$

See:

Fu YX, Li WH. Statistical tests of neutrality of mutations. Genetics. 1993; 133, p. 700, where they note:

$$E(\eta_s) = \frac{n}{n-1}\theta.$$

5.2. **DS_TajimasD.pm.** This class has the following identifying tags as member variables:

- `_human_title = Tajimas D`
- `_data_tag = Tajimas_D`
- `_outgroup_required = 0`
- `_perform_sims = 1`

The formula for Tajima's D is as follows:

$$D = \frac{\theta_\pi - \theta_W}{\sqrt{\frac{1}{A_n}\left(\frac{n+1}{3(n-1)} - \frac{1}{A_n}\right) S + \left(\frac{1}{A_n^2+B_n}\right)\left(\frac{2(n^2+n+3)}{9n(n-1)} - \frac{n+2}{nA_n} + \frac{B_n}{A_n}\right) S(S-1)}}$$

To deal with the possibility that numerical error will give us a non-zero result for the denominator that should be treated as a 'divide by zero' situation, we set the variable `$accuracy = 1.0e-30`, and when the denominator is below this, we'll assign Tajima's D to 'NA' (For a very simple example of when denominator is zero, look at the two sequences AA and AT). Note Tajima's D is not defined if $S < 1$ or $n < 2$ (since A_n and B_n are not defined when $n < 2$). See Wakeley, p. 115. Also note this is different from the formula that appears in Hartl, p. 113, in that we are not incorporating the length of the alignment into the calculation. Also see Gillespie, p.44-45.

5.3. **DS_F_star.pm.** This class has the following identifying tags as member variables:

- `_human_title = F*`
- `_data_tag = F_star`
- `_outgroup_required = 0`
- `_perform_sims = 1`

Calculation for Fu and Li's F* (as corrected by Simonsen *et. al.*).

$$v_{F^*} = \left[\frac{2n^3 + 110n^2 - 255n + 153}{9n^2(n-1)} + \frac{2(n-1)A_n}{n^2} - \frac{8B_n}{n}\right] / (A_n^2 + B_n)$$

$$u_{F^*} = \left(\left(\frac{4n^2 + 19n + 3 - 12(n+1)A_{n+1}}{3n(n-1)}\right) / A_n\right) - v_F^*$$

$$F^* = \frac{\theta_\pi - \theta_{N_S}}{\sqrt{(u_{F^*} \times S) + (v_{F^*} \times S^2)}}$$

See:

Fu YX, Li WH. Statistical tests of neutrality of mutations. Genetics. 1993; 133:693-709.

Simonsen KL, Churchill GA, Aquadro CF. Properties of statistical tests of neutrality for DNA polymorphism data. Genetics. 1995; 141:413-429.

5.4. **DS_D_star.pm.** This class has the following identifying tags as member variables:

- `_human_title = D*`
- `_data_tag = D_star`
- `_outgroup_required = 0`
- `_perform_sims = 1`

$$v_{D^*} = \left[\frac{B_n}{A_n^2} - \frac{2}{n}\left(1 + \frac{1}{A_n} - A_n + \frac{A_n}{n}\right) - \frac{1}{n^2}\right] / \left(A_n^2 + B_n\right)$$

$$u_{D^*} = \left[\left(\frac{(n-1)}{n} - \frac{1}{A_n}\right) / A_n\right] - v_{D^*}$$

$$D^* = \frac{\theta_W - \theta_{N_S}}{\sqrt{(u_{D^*} \times S) + (v_{D^*} \times S^2)}}$$

Calculation for Fu and Li's D* as corrected by Simonsen et al. (1995)

5.5. **DS_ThetaNe.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaNe`
- `_data_tag = ThetaNe`
- `_outgroup_required = 1`
- `_perform_sims = 1`

As part of our analysis, we already generated the unfolded site frequency spectrum $\xi_i, 1 \leq i \leq n-1$. Then

$$\theta_{Ne} = \xi_1,$$

i.e. θ_{Ne} = the number of external mutants in the alignment. Note for processing the simulations, `ms` represents the ancestral allele with a zero and a mutant allele with a one, so we can generate the unfolded site frequency spectrum for each simulation with this information instead of the explicit outgroup information we used for the observed data.

See:

Fu YX, Li WH. Statistical tests of neutrality of mutations. Genetics. 1993; 133, p. 696, equation 18, where they note:

$$E(\eta_e) = \theta.$$

where the variable they denote as η_e corresponds to our ξ_1.

5.6. **DS_thetaH.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaH`
- `_data_tag = ThetaH`
- `_outgroup_required = 1`
- `_perform_sims = 1`

Fay and Wu's $\hat{\theta}_H$ is an unbiased estimator for θ that is weighted on the higher frequency mutant alleles.. It is defined as:

$$\hat{\theta}_H = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} (\xi_i \cdot i^2)$$

See: Hitchhiking Under Positive Darwinian Selection, Justin C. Fay and Chung-I Wu, Genetics 155: 1405-1413 (2000).

5.7. **DS_ThetaH_outgroup_SD.pm.** This class has the following identifying tags as member variables:

- _human_title = ThetaH_outgroup_SD
- _data_tag = ThetaH_outgroup_SD
- _outgroup_required = 1
- _perform_sims = 0

5.8. **DS_thetaL.pm.** This class has the following identifying tags as member variables:

- _human_title = ThetaL
- _data_tag = ThetaL
- _outgroup_required = 1
- _perform_sims = 1

$\hat{\theta}_L$, an estimator for θ, is defined as follows:

$$\hat{\theta}_L = \frac{1}{n-1} \sum_{i=1}^{n-1} (\xi_i \cdot i)$$

See: Statistical Tests for detecting Positive Selection by Utilizing High-Frequency Variants, Kai Zeng, Yun-Xin Fu, Suhua Shi and Chung-I Wu, Genetics 174: 1431-1439 (November 2006), equation 8.

5.9. **DS_ThetaL_outgroup_SD.pm.** This class has the following identifying tags as member variables:

- _human_title = ThetaL_outgroup_SD
- _data_tag = ThetaL_outgroup_SD
- _outgroup_required = 1
- _perform_sims = 0

5.10. **DS_H.pm.** This class has the following identifying tags as member variables:

- _human_title = H
- _data_tag = H
- _outgroup_required = 1
- _perform_sims = 1

Fay and Wu's H statistic is defined as:

$$H = \hat{\theta}_\pi - \hat{\theta}_H$$

See: Hitchhiking Under Positive Darwinian Selection, Justin C. Fay and Chung-I Wu, Genetics 155: 1405-1413 (2000).

5.11. **DS_normH.pm.** This class has the following identifying tags as member variables:

- `_human_title = normH`
- `_data_tag = normH`
- `_perform_sims = 1`
- `_outgroup_required = 1`

From Fay and Wu's H:

$$H = \hat{\theta}_\pi - \hat{\theta}_H = 2(\hat{\theta}_\pi - \hat{\theta}_L),$$

Zeng, Fu, Shi and Wu have created a normalized H-statistic (which we refer to as $normH$ in `dnasam`):

$$normH = H = \frac{\hat{\theta}_\pi - \hat{\theta}_L}{\sqrt{Var(\hat{\theta}_\pi - \hat{\theta}_L)}}$$

with

$$Var(\hat{\theta}_\pi - \hat{\theta}_L) = \frac{n-2}{6(n-1)}\theta + \frac{18n^2(3n+2)B_{n+1} - (88n^3 + 9n^2 - 13n + 6)}{9n(n-1)^2}\theta^2$$

and where, as an estimator for θ, we use θ_W, and as an estimator for θ^2 we use $S(S-1)/(A_n^2 + B_n)$ in our implementation.

See: Kai Zeng, Yun-Xin Fu, Suhua Shi and Chung-I Wu, Statistical Tests for Detecting Positive Selection by Utilizing High-Frequency Variants, Genetics 174: 1431-1439 (November 2006).

5.12. **DS_normE.pm.** This class has the following identifying tags as member variables:

- `_human_title = normE`
- `_data_tag = normE`
- `_outgroup_required = 1`
- `_perform_sims = 1`

Zeng, Fu, Shi and Wu introduce the test statistic E (which we refer to as $normE$ in `dnasam`):

$$normE = E = \frac{\hat{\theta}_L - \hat{\theta}_W}{\sqrt{Var(\hat{\theta}_L - \hat{\theta}_W)}}$$

with

$$Var(\hat{\theta}_L - \hat{\theta}_W) = \left[\frac{n}{2(n-1)} - \frac{1}{A_n}\right]\theta + \left[\frac{B_n}{A_n^2} + 2\left(\frac{n}{n-1}\right)^2 B_n - \frac{2(nB_n - n - 1)}{(n-1)A_n} - \frac{3n+1}{n-1}\right]\theta^2,$$

where we estimate θ with $\hat{\theta}_W$, and we estimate θ^2 with $S(S-1)/(A_n^2 + B_n)$ in our implementation.

See: Kai Zeng, Yun-Xin Fu, Suhua Shi and Chung-I Wu, Statistical Tests for Detecting Positive Selection by Utilizing High-Frequency Variants, Genetics 174: 1431-1439 (November 2006).

5.13. **DS_F_outgroup.pm.** This class has the following identifying tags as member variables:

- `_human_title = F(outgroup)`
- `_data_tag = F_outroup`
- `_outgroup_required = 1`
- `_perform_sims = 1`

5.14. **DS_D_outgroup.pm.** This class has the following identifying tags as member variables:

- `_human_title = D(outgroup)`
- `_data_tag = D_outgroup`
- `_outgroup_required = 1`
- `_perform_sims = 1`

5.15. **DS_R2.pm.** This class has the following identifying tags as member variables:

- `_human_title = R2`
- `_data_tag = R2`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We'll use u_i to denote the number of singletons in the ith sequence, $1 \leq i \leq n$, and u to denote the total number of singletons, so $0 \leq u_i \leq u$. Then we calculate R_2 as follows:

$$R_2 = \frac{((\sum_{i=1}^{n}(u_i - \frac{\theta_\pi}{2})^2)/n)^{\frac{1}{2}}}{S}$$

Note that calculation of R_2 does **not** require an outgroup.

5.16. **DS_R3.pm.** This class has the following identifying tags as member variables:

- `_human_title = R3`
- `_data_tag = R3`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We calculate R_3 in a manner analogous to R_2:

$$R_3 = \frac{((\sum_{i=1}^{n}(u_i - \frac{\theta_\pi}{2})^3)/n)^{\frac{1}{3}}}{S}$$

Note that calculation of R_3 does **not** require an outgroup.

5.17. **DS_R4.pm.** This class has the following identifying tags as member variables:

- `_human_title = R4`
- `_data_tag = R4`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We calculate R_4 in a manner analogous to R_2:

$$R_4 = \frac{((\sum_{i=1}^{n}(u_i - \frac{\theta_\pi}{2})^4)/n)^{\frac{1}{4}}}{S}$$

Note that calculation of R_4 does **not** require an outgroup.

5.18. **DS_Ch.pm.** This class has the following identifying tags as member variables:

- `_human_title = Ch`
- `_data_tag = Ch`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We calculate Ch in a manner analogous to R_2:

$$Ch = \frac{(u - \frac{n\theta_\pi}{n-1})^2 S}{(\frac{n\theta_\pi}{n-1})(S - (\frac{n\theta_\pi}{n-1}))}$$

Note that calculation of Ch does **not** require an outgroup.

5.19. **DS_R2e.pm.** This class has the following identifying tags as member variables:

- `_human_title = R2e`
- `_data_tag = R2e`
- `_outgroup_required = 1`
- `_perform_sims = 1`

We'll use v_i to denote the number of external mutants in the ith sequence, $1 \leq i \leq n$, and v to denote the total number of external mutants present in the alignment, so $0 \leq v_i \leq v$. Then the calculations for R_{2e}, R_{3e}, R_{4e} and Che are analogous to the calculations for R_2, R_3, R_4 and Ch, but we use v_i and v as counts of external mutations rather than u_i and u as counts of singletons. We calculate R_{2e} as follows:

$$R_{2e} = \frac{((\sum_{i=1}^{n}(v_i - \frac{\theta_\pi}{2})^2)/n)^{\frac{1}{2}}}{S}.$$

Note that calculation of R_{2e} **does** require an outgroup.

5.20. **DS_R3e.pm.** This class has the following identifying tags as member variables:

- `_human_title = R3e`
- `_data_tag = R3e`
- `_outgroup_required = 1`
- `_perform_sims = 1`

Analogous to R_{2e}, we calculate R_{3e} as follows:

$$R_{3e} = \frac{((\sum_{i=1}^{n}(v_i - \frac{\theta_\pi}{2})^3)/n)^{\frac{1}{3}}}{S}.$$

Note that calculation of R_{3e} **does** require an outgroup.

5.21. **DS_R4e.pm.** This class has the following identifying tags as member variables:

- `_human_title = R4e`
- `_data_tag = R4e`
- `_outgroup_required = 1`
- `_perform_sims = 1`

Analogous to R_{2e}, we calculate R_{4e} as follows:

$$R_{4e} = \frac{((\sum_{i=1}^{n}(v_i - \frac{\theta_\pi}{2})^4)/n)^{\frac{1}{4}}}{S}.$$

Note that calculation of R_{4e} **does** require an outgroup.

5.22. **DS_Che.pm.** This class has the following identifying tags as member variables:

- `_human_title = Che`
- `_data_tag = Che`
- `_outgroup_required = 1`
- `_perform_sims = 1`

We calculate Che in a manner analogous to Ch, except instead of using u (the total singleton count), we use v (the total external mutant count):

$$Che = \frac{(v - \frac{n\theta_\pi}{n-1})^2 S}{(\frac{n\theta_\pi}{n-1})(S - (\frac{n\theta_\pi}{n-1}))}$$

Note that calculation of Che **does** require an outgroup.

5.23. **DS_Zns.pm.** This class has the following identifying tags as member variables:

- `_human_title = Zns`
- `_data_tag = Zns`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We start with the following measure of linkage disequilibrium:

$$D_{ij} = p_{ij} - p_i p_j$$

where p_i and p_j are the frequencies of the mutant alleles at the at the ith and jth loci, and p_{ij} is the frequency of the sequences that have mutant alleles at both the ith and jth loci.

From an implementation perspective, using `ms`-style binary encoding where 0 represents the ancestral allele and 1 represents the mutant allele, to calculate p_i, we just add the values at the ith locus and divide by the total number of sequences n. To calculate p_{ij}, we count the number of sequences that have a value of 1 at both the ith and jth loci, and divide by n.

After determining p_i, p_j and D_{ij} from our sequence information, we calculate δ_{ij}:

$$\delta_{ij} = \frac{D_{ij}^2}{p_i(1-p_i)p_j(1-p_j)}$$

and note that δ_{ij} gives the same value regardless of which alleles we represent with 0 and 1 at a given biallelic segregating site, so we don't have to know which allele is the ancestral allele and which is the mutant in order to perform the calculation.

Finally, we sum the values of δ_{ij} pairwise over all the S segregating sites, then divide by the total number of pairwise comparisons $\begin{pmatrix} S \\ 2 \end{pmatrix}$:

$$Z_{nS} = \frac{2}{S(S-1)} \sum_{i=1}^{S-1} \sum_{j=i+1}^{S} \delta_{ij}$$

See: A Test of Neutrality Based on Interlocus Associations by John K. Kelly, Genetics, 146: 1197 - 1206 (July, 1997).

5.24. **DS_Dns_star.pm.** This class has the following identifying tags as member variables:

- `_human_title = Dns_star`
- `_data_tag = Dns_star`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We start with the expression for Lewontin's D:

$$D_{ij} = p_{ij} - p_i p_j.$$

calculating p_{ij}, p_i and p_j in the same manner as was done for Z_{nS}. Then for each value of D_{ij} we calculate D_x:

$$D_x = \begin{cases} min\{p_i(1-p_j), p_j(1-p_i)\} & \text{if } D_{ij} \geq 0 \\ min\{p_i p_j, (1-p_i)(1-p_j)\} & \text{if } D_{ij} < 0 \end{cases}$$

and D'_{ij}:

$$D'_{ij} = \frac{D_{ij}}{D_x}$$

and note $D_{ij} = 0$ implies $D'_{ij} = 0$. These values are then used to calculate D^*_{ns}:

$$D^*_{nS} = \frac{2}{S(S-1)} \sum_{i=1}^{S-1} \sum_{j=i+1}^{S} (D'_{ij})^2.$$

(Reference: A Test of Neutrality Based on Interlocus Associations by John K. Kelly, Genetics, 146: 1197 - 1206 (July, 1997) for Z_{nS} information.)

5.25. **DS_Zns_star.pm.** This class has the following identifying tags as member variables:

- `_human_title = Zns_star`
- `_data_tag = Zns_star`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We use our previously calculated value for Z_{nS} and D^*_{nS} to calculate Z^*_{nS}:

$$Z^*_{ns} = Z_{ns} + 1 - D^*_{ns}.$$

See: A Test of Neutrality Based on Interlocus Associations by John K. Kelly, Genetics, 146: 1197 - 1206 (July, 1997)

5.26. **DS_Zns_star_star.pm.** This class has the following identifying tags as member variables:

- `_human_title = Zns_star_star`
- `_data_tag = Zns_star_star`
- `_outgroup_required = 0`
- `_perform_sims = 1`

Using previously calculated statistics, we calculate the Z^{**}_{nS} as follows:

$$Z^{**}_{nS} = \frac{Z_{nS}}{D^*_{nS}}.$$

See: A Test of Neutrality Based on Interlocus Associations by John K. Kelly, Genetics, 146: 1197 - 1206 (July, 1997).

5.27. **DS_Za.pm.** This class has the following identifying tags as member variables:

- `_human_title = Za`
- `_data_tag = Za`
- `_outgroup_required = 0`
- `_perform_sims = 1`

The Z_A statistic is similar to the Z_{nS} statistic, except that instead of doing a complete pairwise comparison between all polymorphic sites, we restrict ourselves to just comparing neighboring polymorphic sites:

$$Z_A = \frac{1}{(S-1)} \sum_{i=1}^{S-1} \delta_{i,i+1}$$

See: DNA Variation at the rp49 Gene Region of Drosophila simulans: Evolutionary Inferences From an Unusual Haplotype Structure, Julio Rozas, Myriam Gullaud, Gaelle Blandin and Monserrat Aguade, Genetics 158: 1147-1155 (July 2001).

5.28. **DS_ZZ.pm.** This class has the following identifying tags as member variables:

- `_human_title = ZZ`
- `_data_tag = ZZ`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We calculate the ZZ statistic using two previously calculated values as follows:

$$ZZ = Z_A - Z_{nS}$$

See: DNA Variation at the rp49 Gene Region of Drosophila simulans: Evolutionary Inferences From an Unusual Haplotype Structure, Julio Rozas, Myriam Gullaud, Gaelle Blandin and Monserrat Aguade, Genetics 158: 1147-1155 (July 2001).

5.29. **DS_B.pm.** This class has the following identifying tags as member variables:

- `_human_title = B`
- `_data_tag = B`
- `_outgroup_required = 0`
- `_perform_sims = 1`

As per Wall, we note the following useful definition:

Definition 1. *A pair of segregating sites is said to be* ***congruent*** *when the two sites exhibit exactly two haptotypes.*

We then define $B^{'}$ as follows:

$$B^{'} = \text{Count of adjacent, congruent segregating sites in our alignment.}$$

So we perform a total of $S - 1$ comparisons of neighboring sites, count the number of adjacent, congruent segregating sites, and then calculate:

$$B = \frac{B^{'}}{(S-1)}.$$

(Reference: Recombination and the Power of Statistical Tests of Neutrality, Jeffrey D. Wall, Genet. Res. Camb. (1999), 74, pp. 65-79)

5.30. **DS_Q.pm.** This class has the following identifying tags as member variables:

- `_human_title = Q`
- `_data_tag = Q`
- `_outgroup_required = 0`
- `_perform_sims = 1`

We use B from a previous calculation. For Q however, we are also interested in the patterns formed by the adjacent congruent segregating sites. To determine these partitions in our implementation, at each neighboring pair of segregating sites, we determine the haplotype of the two bases in the first row. We then go through all the rows, keeping track of both which positions match the haplotype in our first row, as well as all the haplotype patterns that exist. Examining the list of haplotypes that exist for our neighboring segregating sites, if there are exactly two haplotypes (and hence our pair of sites is congruent), we note the pattern that was found. Let A denote the set of all these unique patterns, and we'll use $|A|$ to denote the cardinality of A, that is, the total number of individual haplotypic patterns found for congruent adjacent segregating sites. We then calculate Q, which is defined as follows:

$$Q = \frac{(B + |A|)}{S}.$$

(Reference: Recombination and the Power of Statistical Tests of Neutrality, Jeffrey D. Wall, Genet. Res., Camb. (1999), 74, pp. 65-79.)

5.31. **DS_h_hap_count.pm.** This class has the following identifying tags as member variables:

- `_human_title = h`
- `_data_tag = h`
- `_outgroup_required = 0`
- `_perform_sims = 1`

This package just counts the number of different haplotypes found in our data set, which we label as h.

5.32. **DS_Hd.pm.** This class has the following identifying tags as member variables:

- `_human_title = Hd`
- `_data_tag = Hd`
- `_outgroup_required = 0`
- `_perform_sims = 1`

To calculate $\hat{H}_d$, we first make a call to `DS_BasicAnalysis.pm::get_data_hap_dist_hash()`, which returns a reference to a hash where the keys are an `ms`-style binary representation of each haplotype found in the alignment, and it hashes to a count of how many times that particular haplotype occurs in our alignment. Assume we have k distinct haplotypes in an alignment of n samples. We use these counts to calculate p_i, the frequency of each haplotype, $1 \leq i \leq k$:

$$p_i = \frac{\text{Count of } i\textit{th} \text{ haplotype occurances in the alignment}}{n}.$$

We then calculate $\hat{H}_d$ as a measure of haplotypic diversity as follows:

$$\hat{H}_d = \frac{n}{n-1}\left[1 - \sum_{i=1}^{k} p_i^2\right]$$

(References: Molecular Evolutionary Genetics, Masatoshi Nei, Columbia University Press, 1987, pp. 177. Note that our formulation includes an additional $\frac{n}{n-1}$ as a correction factor for when n is small.

5.33. **DS_Hd_SD.pm.** This class has the following identifying tags as member variables:

- `_human_title = Hd_SD`
- `_data_tag = Hd_SD`
- `_outgroup_required = 0`
- `_perform_sims = 0`

We calculate the the variance of $\hat{H}_d$ according to the following formula:

$$Var(\hat{H}_d) = \frac{2}{n(n-1)}\left[2(n-2)\left[\sum_{i=1}^{k} p_i^3 - (\sum_{i=1}^{k} p_i^2)^2\right] + \sum_{i=1}^{k} p_i^2 - (\sum_{i=1}^{k} p_i^2)^2\right]$$

and then the standard deviation of $\hat{H}_d$:

$$sd(\hat{H}_d) = \sqrt{Var(\hat{H}_d)}$$

where terms are derived in the manner described in our calculation for $\hat{H}_d$. (Reference: Molecular Evolutionary Genetics, Masatoshi Nei, Columbia University Press, 1987, pp. 180.

5.34. **DS_ThetaHom.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaHom`
- `_data_tag = ThetaHom`
- `_outgroup_required = 0`
- `_perform_sims = 1`

Using the previously calculated value for $\hat{H}_d$, calculate $\hat{\theta}_{Hom}$ as follows:

$$\hat{\theta}_{Hom} = \frac{\hat{H}_d}{1 - \hat{H}_d}.$$

5.35. **DS_ThetaHom_SD.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaHom_SD`
- `_data_tag = ThetaHom_SD`
- `_outgroup_required = 0`
- `_perform_sims = 0`

Using previously calculated values for $\hat{H}_d$, $sd(\hat{H}_d)$ and $\hat{\theta}_{Hom}$, we calculate the standard deviation of $\hat{\theta}_{Hom}$ as follows:

$$sd(\hat{\theta}_{Hom}) = \frac{(2 + \hat{\theta}_{Hom})^2(3 + \hat{\theta}_{Hom})^2 sd(\hat{H}_d)}{(\hat{H}_d)^2(1 + \hat{\theta}_{Hom})[(2 + \hat{\theta}_{Hom})(3 + \hat{\theta}_{Hom})(4 + \hat{\theta}_{Hom}) + 10(2 + \hat{\theta}_{Hom}) + 4]}.$$

5.36. **DS_ThetaHomCorrected.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaHomCorrected`
- `_data_tag = ThetaHomCorrected`
- `_outgroup_required = 0`
- `_perform_sims = 1`

From Zouros, for the actual value of $\theta = 4N\mu$ in our population, we have the following (based on the first terms of a Taylor's expansion):

$$E(\hat{\theta}_{Hom}) \approx \theta \left[1 + \frac{2(1+\theta)}{(2+\theta)(3+\theta)}\right]$$

This approximation is then used to form the following equation:

$$f(\theta) = \theta \left[1 + \frac{2(1+\theta)}{(2+\theta)(3+\theta)}\right] - \hat{\theta}_{Hom} = \theta \left[1 + \frac{2(1+\theta)}{(2+\theta)(3+\theta)}\right] - \frac{1-\hat{H}_d}{\hat{H}_d}.$$

Treating $\hat{\theta}_{Hom}$ as a constant and differentiating with respect to θ we get:

$$f^{'}(\theta) = 1 + \frac{[(2+4\theta)(\theta^2+5\theta+6)] - [(2\theta+2\theta^2)(2\theta+5)]}{(\theta^2+5\theta+6)^2}.$$

We then use a Newton-Raphson procedure to numerically solve $f(\theta) = 0$, which will give us a 'corrected' value of $\hat{\theta}_{Hom}$ that we will call $\hat{\theta}_{Hom,cor}$. The first few terms in our iteration are as follows:

$$\begin{aligned}
\theta_0 &= \hat{\theta}_{Hom} \\
\theta_1 &= \theta_0 - \tfrac{f(\theta_0)}{f'(\theta_0)} \\
\theta_2 &= \theta_1 - \tfrac{f(\theta_1)}{f'(\theta_1)} \\
&\dots
\end{aligned}$$

For some specified value of `$accuracy` in our class we perform the iteration until

$$\left|\frac{f(\theta_k)}{f'(\theta_k)}\right| < \texttt{\$accuracy},$$

where we return the value of the numerically determined root as $\hat{\theta}_{Hom,cor}$, or until we exceed `$max_loop` iterations, at which point we assume the Newton-Raphson procedure is not converging, and we return the value 'NA'. We use the following default values: `$accuracy` $= 1.0 \times 10^{-15}$ and `$max_loop` $= 20$.

See: Zouros, E. (1979). Mutation rates, population sizes, and amounts of electrophoretic variation of enzyme loci in natural populations. Genetics 92:623

5.37. **DS_ThetaHomCorrected_SD.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaHomCorrected_SD`
- `_data_tag = ThetaHomCorrected_SD`
- `_outgroup_required = 0`
- `_perform_sims = 0`

Similar to the calculation for `DS_ThetaHom_SD.pm`, using previously calculated values for $\hat{H}_d$, $\hat{\theta}_{Hom,cor}$ and $sd(\hat{H}_d)$, we calculate the standard deviation of $\hat{\theta}_{Hom,cor}$ as follows:

$sd(\hat{\theta}_{Hom,cor}) =$

$$\frac{(2+\hat{\theta}_{Hom,cor})^2(3+\hat{\theta}_{Hom,cor})^2 sd(\hat{H}_d)}{(\hat{H}_d)^2(1+\hat{\theta}_{Hom,cor})[(2+\hat{\theta}_{Hom,cor})(3+\hat{\theta}_{Hom,cor})(4+\hat{\theta}_{Hom,cor})+10(2+\hat{\theta}_{Hom,cor})+4]}.$$

5.38. **DS_WattersonsF.pm.** This class has the following identifying tags as member variables:

- `_human_title = F`
- `_data_tag = F`
- `_outgroup_required = 0`
- `_perform_sims = 1`

For k haplotypes in an alignment, $1 \leq k \leq n$, and p_i indicating the frequency of the ith haplotype, Watterson's F is defined as:

$$F = \sum_{i=1}^{k} p_i^2.$$

5.39. **DS_ThetaK.pm.** This class has the following identifying tags as member variables:

- `_human_title = ThetaK`
- `_data_tag = ThetaK`
- `_outgroup_required = 0`
- `_perform_sims = 1`

For h the haplotype count of our sample, we have an expected value of

$$E(h) = \theta \sum_{j=0}^{n-1} \frac{1}{\theta + j}.$$

Now substitute our observed haplotype count h as an estimator for $E(h)$, for the equation:

$$f(\theta) = \left(\sum_{j=0}^{n-1} \frac{\theta}{\theta + j} \right) - h = (1-h) + \left(\sum_{j=1}^{n-1} \frac{\theta}{\theta + j} \right),$$

and solve $f(\theta) = 0$ numerically (using Newton-Raphson). This root is $\hat{\theta}_k$, an estimator for θ based on our sample size n and our observed haplotype count h. We differentiate $f(\theta)$ with respect to θ to get:

$$f'(\theta) = \sum_{j=1}^{n-1} \frac{j}{(\theta + j)^2}.$$

We'll use our previously calculated value of $\hat{\theta}_\pi$ as an estimate for $\hat{\theta}_k$ in our in initial iteration of our Newton-Raphson method. Then the first few terms of our iteration will be as follows:

$$\begin{aligned}
\theta_0 &= \hat{\theta}_\pi \\
\theta_1 &= \theta_0 - \tfrac{f(\theta_0)}{f'(\theta_0)} \\
\theta_2 &= \theta_1 - \tfrac{f(\theta_1)}{f'(\theta_1)} \\
&\dots
\end{aligned}$$

And as was done in our calculation for $\hat{\theta}_{Hom,cor}$, for some specified value of `$accuracy` in our class we perform the iteration until

$$\left| \frac{f(\theta_j)}{f'(\theta_j)} \right| < \texttt{\$accuracy},$$

where we return the value of the numerically determined root as $\hat{\theta}_k$, or until we exceed `$max_loop` iterations, at which point we assume the Newton-Raphson procedure is not converging, and we return the value 'NA'. We use the following default values: `$accuracy` $= 1.0 \times 10^{-6}$ and `$max_loop` $= 500$.

Note that when $h = n$, a solution fails to exist in the Newton-Raphson procedure, and we return $\hat{\theta}_k = NA$.

See: Ewens, W.J., 1972 The Sampliing Theory of Neutral Alleles, Theor. Popul. Biol., 3:87-112

5.40. **DS_Fs.pm.** This class has the following identifying tags as member variables:

- `_human_title = Fus Fs`
- `_data_tag = Fs`
- `_outgroup_required = 0`
- `_perform_sims = 1`

For observed haplotype count $h_{obs}, 1 \leq h_{obs} \leq n$, initially we want to calculate $S^{'}$ the probability that the actual haplotype count h is greater than or equal to the value of h_{obs} of our sample, conditioned on $\theta = \theta_\pi$. Here we follow Knuth and use the notation:

$$\left[\begin{array}{c} n \\ h_{obs} \end{array} \right]$$

to represent a "n cycle h_{obs}", a (signed) Stirling number of the first kind, and $S_n(\theta)$ is used to represent the following product:

$$S_n(\theta) = \theta(\theta+1)(\theta+2)\cdots(\theta+n-1).$$

Then note the following result:

$$Prob(h = h_{obs} \mid \theta = \theta_\pi) = \frac{\left|\left[\begin{array}{c} n \\ h_{obs} \end{array} \right]\right| \theta^{h_{obs}}}{S_n(\theta)},$$

and it follows that:

$S^{'} = Prob(h \geq h_{obs} \mid \theta = \theta_\pi) =$

$Prob(h = h_{obs} \mid \theta = \theta_\pi) + Prob(h = h_{obs}+1 \mid \theta = \theta_\pi) + \ldots + Prob(h = n \mid \theta = \theta_\pi) \Rightarrow$

$$S^{'} = \frac{1}{S_n(\theta)} \sum_{i=h_{obs}}^{n} \left|\left[\begin{array}{c} n \\ i \end{array} \right]\right| \theta^i$$

Then the formula for Fu's F_s is given by:

$$F_s = \ln\left(\frac{S^{'}}{1-S^{'}}\right).$$

5.41. **DS_FixedDifferenceCount.pm.** This class has the following identifying tags as member variables:

- `_human_title = FixedDifferenceCount`
- `_data_tag = FixedDifferenceCount`
- `_outgroup_required = 1`
- `_perform_sims = 0`

Count the number of sites where the ingroup is fixed, the outgroup is fixed, and the bases for the two differ. When there is a single outgroup sequence, sites are treated as fixed.

5.42. **DS_SharedPolymorphismCount.pm.** This class has the following identifying tags as member variables:

- `_human_title = SharedPolymorphismCount`
- `_data_tag = SharedPolymorphismCount`
- `_outgroup_required = 1`
- `_perform_sims = 0`

When we have multiple sequences in the outgroup, count the number of sites where both the ingroup and the outgroup are biallelic and have the same alleles.

5.43. **DS_MultiAllelicSiteCount.pm.** This class has the following identifying tags as member variables:

- `_human_title = MultiAllelicSiteCount`
- `_data_tag = MultiAllelicSiteCount`
- `_outgroup_required = 1`
- `_perform_sims = 0`

Count the number of sites in the ingroup that are tri- or tetra-allelic and the outgroup has a base that matches A, C, G or T.

5.44. **multiDimenTest.pl.** Different summary statistics include information about different parts of the site frequency spectrum. This attribute derives largely from the fact that different estimators of theta are sensitive to different portions of this spectrum. For example, $\hat{\theta}_W$ is sensitive to changes in S, whereas $\hat{\theta}_\pi$ is sensitive to changes in the number and/or frequency of intermediate frequency variants. This means that rare polymorphisms contribute more to $\hat{\theta}_W$ than to $\hat{\theta}_\pi$. Try an example for yourself or see Figure 4.7 in Wakeley (2009). Thus, Tajima's D can be thought of as a summary statistic that compares low and intermediate frequency variants.

Tests of neutrality that rely on summary statistics already lose information by compressing data patterns into a single statistic. If those statistics, moreover, do not capture large portions of the variation in the site frequency spectrum (which is already a summary of the data) then the situation is even worse. To address this problem, a series of multidimensional tests were proposed by Kai Zeng and colleagues. Conceptually, the idea is to capture more of the variation in the original data without resorting to computationally intensive likelihood or Bayesian techniques by pairing summary statistics and performing tests on the joint values of these statistics. Each summary statistic captures a different aspect of the original data. The tests implemented in `dnasam` are the DH, HEW and DHEW tests: DH is a compound test of Tajima's D and Fay and Wu's H. HEW is a compound test with

Fay and Wu's H with Watterson's F. DHEW is a compound test with Tajima's D, Fay and Wu's H and Watterson's F.

These tests require that we define a rejection region or a tail in the joint distribution, which is a multidimensional space. Values falling in this region are deemed too extreme under the null model. For the two statistic tests this space has 2 dimensions, and, as you might have guessed, for the three statistic test there is 3 dimensions. If we just used a probability threshold of $p = 0.05$ for each single statistic and counted as significant data patterns where $p < 0.05$ for each then we would not be using a nominal threshold of $p = 0.05$ for the multidimensional case. The reason can be understood by looking at the definition of a rejection region. A rejection region is defined as the portion of the distribution where a certain percent of the data are located. When we specify a 5% threshold for the P-value in a one-tailed test, we are specifying a rejection region equal to the 5% tail of the distribution of the test statistic. When the number of dimensions increases from one, the definition of the rejection region stays the same, but the n-dimensional space defined by $p = 0.05$ for each statistic does not capture 5% of the joint distribution. It captures something much less. We must, therefore, find the optimal value of each statistic such that the user-specified rejection contains the user specified value of the joint distribution. This typically gives much larger P-values (i.e. greater than the single test threshold) for each single test that define data patterns as significant under the multidimensional test.

The `multiDimenTest.pl` script included with `dnasam` is run as a separate script. When run, you specify the parameters for the model you wish to run in `ms`. The `multiDimenTest.pl` script runs `ms` with these parameters, parses the output, and determines the multidimentional P-values for these three compound statistics.

For example, the command:

```
multiDimenTest.pl -ms_opts='20 10000 -t 4.4'
```

runs `ms`, generating 10000 simulations according to a model with 20 samples and a value of $\theta = 4.4$, and a default P-value of 0.05. The output:

```
Running ms command: './msdir/ms 20 10000 -t 4.4'
-----
For user-specified p-value = 0.05000, DH-p* = 0.13551, \
HEW-p* = 0.16200, DHEW-p* = 0.26050
Total run time:  0.353 minutes.
```

We could specify a P-value; say, 0.01. Specifying a model that has 20 samples and $\theta = 2.2$ where you wish to run 50000 samples, you would use the command:

```
multiDimenTest.pl -ms_opts='20 50000 -t 2.2' -pval=0.01
```

which would have the following output:

```
Running ms command: './msdir/ms 20 50000 -t 2.2'
-----
```

```
For user-specified p-value = 0.01000, DH-p* = 0.04288, \
HEW-p* = 0.05718, DHEW-p* = 0.09434
Total run time:  1.405 minutes.
```

For more information, see: "Compound Tests for the Detection of Hitchhiking Under Positive Selection, Zeng et. al., Mol. Biol. Evol 24(8):1898-1908. 2007

6. A brief tutorial on how to add an additional analysis package to DNASAM

6.1. **DS_Stat.pm.** This package acts as a parent class from which our other analysis classes will inherit. It has the member data that are set to `undef` in the parent class, but which will be set in the subclasses as indicated below:

`_title`**:** Holds the column title that will be displayed in the 'table' output format. Note that this includes the title for the p-value column when appropriate, separated by a tab.

`_human_title`**:** The title used for the 'human' output format.

`_data_tag`**:** This tag acts as a key to a hash in DS_BasicAnalysis.pm that holds previously calculated values, so we can access that data if a previously calculated value is useful in a later summary statistic calculation.

`_outgroup_required`**:** Set to 0 (zero) in the derived class to indicate this calculation is to be performed regardless of whether or not an outgroup is present, set to 1 to indicate an outgroup is required to perform the calculation for this class.

`_perform_sims`**:** Set to 0 (zero) in the derived class to indicate simulations will not be performed for this class, set to 1 to indicate simulations will be performed.

`_basic_analysis`**:** This is a reference that points back to the `DS_Basic_Analysis` object for the locus.

`_sim_count`**:** Holds the count of the simulations performed.

`_stat`**:** Holds the value of the summary statistic for our observed data.

`_stat_pval_count`**:** Holds the count of how many simulations had a summary statistic that was $\leq$ to the summary statistic for our observed data.

There are also the following member functions:

`get_title()`**:** Returns the value of `_title`.

`get_human_title()`**:** Returns the value of `_human_title`.

`outgroup_required()`**:** Returns the value of `_outgroup_required`.

`print_stats()`: Prints out the appropriate summary statistic infomation in the proper format, depending on whether the user has selected the 'human' or the 'table' print option.

If need be, a derived class can override these superclass member functions. (For example, if one wishes to change the labels that are generated when printing the data in either 'human' or 'table' form.)

Adding an additional analysis package to the `dnasam` workflow requires two steps:

Implement the class : We provide a class package, `DS_Template.pm`, that you can copy, as a foundation for your new class. Say you want a to calculate a fictitious statistic called Smith's Z. In the `lib` directory, you would copy the `DS_Template.pm` file to `DS_SmithsZ.pm`, then open `DS_SmithsZ.pm` in your favorite text editor. You will need to change the package name as follows:

```
package DS_SmithsZ;
```

Then in the constructor you will need to set the following member variables inherited from `DS_Stat.pm`:

- `_title` These will become the column headers in the table output format. So we'll set:
  `_title => "SmithsZ\tSmithsZ_pval"`
- `_human_title` This will be the title for human output. We'll set it to something like:
  `_human_title => "Smiths"`
- `_data_tag` The tag that will get assigned to the statistic, so it can be accessed by later analysis modules. We'll set it to:
  `_data_tag => "SmithsZ"`
- `_outgroup_required` If set to 1, this indicates that we are required to have an outgroup to calculate this statistic. If no outgroup is present, the calculation is skipped. If we set this to zero, this indicates an outgroup is not required, so the calculation is performed on all alignments. We'll set this to:
  `_outgroup_required => 0`
- `_perform_sims` This indicates we want to calculate this statistic for the simulations generated by `ms` to determine a p-value. So we'll set this:
  `_perform_sims => 1`

Then we need to implement the `calculateStat()` method, and if we are calculating the statistic on the simulations from `ms`, we'll also need to implement the

`processSimulation()` method. There are many examples of how to do this in packages that can be found `lib` directory.

Finally we'll need to add exactly two lines of code to the `dnasam.pl` program : We'll need to provide instructions for the new package in the `dnasam.pl` program - the first in the form of a `use` statement, and the second line where we use `new()` to instantiate a new object from the class and push it onto the end of the `@analysis_objects`, our array of analysis objects. For our example for a package that calculates Smith's Z, these lines would be as follows:
`use DS_SmithsZ;` and
`push(@analysis_objects, new DS_R2($basic_analysis));`.

Note that objects on the `@analysis_objects` are used to calculate statistics starting at index 0, and then traversing the array in increasing order of the indices. Values for each statistic are stored in a hash that is pointed to by the `_data_vals` member variable in `DS_Basic_Analysis`, so objects that are calculated later in the `@analysis_objects` array can access previously calculated statistics for the alignment using the `DS_Basic_Analysis -> get_data_val()` method.

6.2. **Additional references.** -

Andrew J. Eckert, John D. Liechty, Brandon R. Tearse, Barnaly Pande and David B. Neale, DnaSAM: Software to perform neutrality testing for large data sets with complex null models. (in review)

Hudson, R. R. (2002) Generating samples under a Wright-Fisher neutral model. Bioinformatics 18:337-8

Coalescent Theory: An Introduction, John Wakeley, Roberts and Company Publishers, 2009.

A Primer in Population Genetics, Daniel L. Hartl, Sinauer, 2000

Principles of Population Genetics, Daniel L. Hartl and Andrew G. Clark, Fourth Edition, Sinauer Associates, Inc. Publishers, Sunderland, Mass., 2007

Population Genetics, A Concise Guide, John H. Gillespie, John Hopkins University Press, 2004

Simonsen, Churchill and Aquadro, Properties of Statistical Tests of Neutrality for DNA Polymorphism Data, Genetics 141: 413-429 (September, 1995)

John K. Kelly, A Test of Neutrality Based on Interlocus Associations, Genetics, 146: 1197 - 1206, July, 1997

Concrete Mathematics: A Foundation for Computer Science, Second edition, Graham, Knuth and Patashnik, Addison-Wesley, 1994