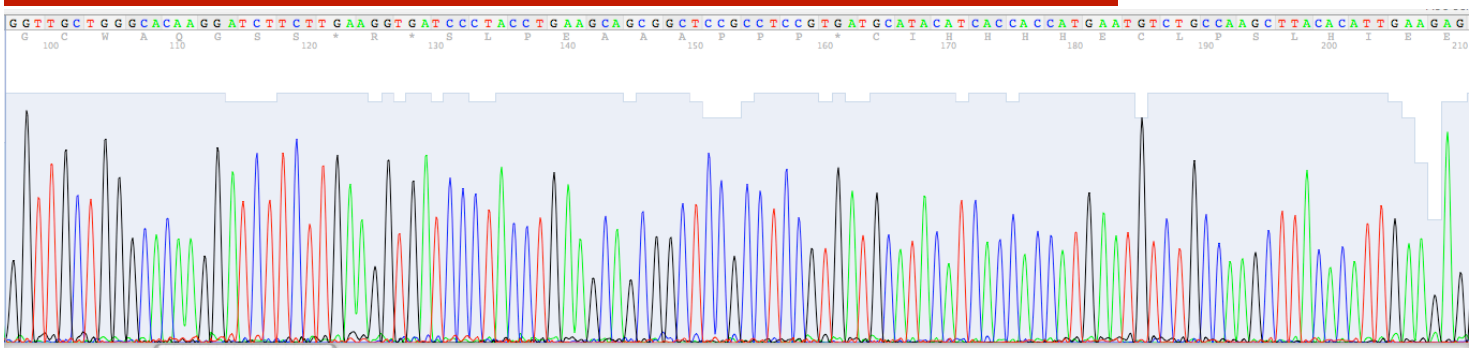
April 2007

Basic EDUT Usage



Getting Started

EDUT is easy to use. Once you have an aligned fasta file, with each sequence listed on two lines, as in the example below, command line options are used to specify the data file and to control input and output. In order for EDUT to be effective for error detection, a data set must include at least 3 SNPs and 4 chromosomes; most data sets will be much larger.

```
>sequence01
ACGTACGTACGTACGT
>sequence02
GCGTACCTACGTACGA
```

Before starting, make sure you have made EDUT executable (see the README). The simplest usage of EDUT, using the example data set provided with the program, is as follows (don't actually type the '$', that is the UNIX command prompt).

$ ./EDUT_1.1.pl -i example_fasta.txt -s

This would result in output that looks like this.

```
# Number of pattern a triplets is 37
# The proportion of pattern a triplets is 0.0160869565217391
#

raw                corrected   site        sequence      sequence
priority           priority    number      name          number
score              score

_______            _________   _________   __________    _______
4                  0.5         563         Cbf3_28        18
4                  0.6         503         Cbf3_28        18
3                  0.5         417         Cbf3_04        3
```

```
7                1.1          1320         Cbf3_30              19
7                1.1          1320         Cbf3_28              18
4                0.6          505          Cbf3_28              18
4                0.8          176          Cbf3_28              18
6                1.1          1474         Cbf3_28              18
9                1.2          926          Cbf3_28              18
10               1.5          1276         Cbf3_28              18
1                0.1          568          Cbf3_04              3
4                0.7          449          Cbf3_30              19
4                0.7          449          Cbf3_28              18
8                1.1          799          Cbf3_28              18
2                0.2          799          Cbf3_04              3
11               1.5          914          Cbf3_30              19
61               8.5          914          Cbf3_28              18
9                1.6          1426         Cbf3_28              18
4                0.5          513          Cbf3_28              18
```

However, to make the output easier to use, we may want to sort it by site number or corrected priority score. To redirect the EDUT output to a file, we would use the following command line input.

$ ./EDUT_1.1.pl -i example_fasta.txt -s > example_out.txt

Then to sort the output based on site number, we can use the UNIX sort command. Site number is the third column in the output, so we would specify column 3. To sort by corrected priority score, we would sort by column 2.

$ sort -n -k 3 example_out.txt

EDUT Options

To see all the usage options for EDUT, execute the program with no options specified. The following usage statement will print.

```
An input file is required.
Input may be a fasta file with option -i
or an ms formatted file from a simulated data set with option -a
The expected usage of this program:

usage: ./EDUT_1.1.pl -b [-d delimiter] -v -m -s -i input_filename

 -i  followed by the filename for the input fasta format - input is
     required.The fasta file should be formatted so that there are no
     line breaks within the sequence data. A line break should appear
     after the sequence name, after each sequence and after the last
```

```
     sequence. This version also requires all capital letters for the
     nucleotides.

 -a followed by the filename for the ms formatted simulation output
    the simulation input may substitute for the -i fasta file input.

 -b indicates print binary data
 -d followed by the delimiter for the binary data output.
    tab (the default)
    space
    none
 -v indicates print verbose output
 -m indicates printing a moderate/reasonable amount of output
 -s indicates that the output should be only a short summary.
    This is the recommended output option.

example: ./EDUT_1.1.pl -i a_fasta_file.txt -b -d none -s

    The following options are new to EDUT_1.1:
 -g use 1 to exclude the gapped sites
    use 0 include the gapped sites (the default)
 -n use 1 to exclude the sites where sequence is missing indicated
by 'N'
    use 0 include the sites where sequence is missing (the default)
```

Two of the options indicated above will dramatically reduce processing time in data sets that include large alignment gaps or samples with missing data. With the -g option set to 1, as in '-g 1', sites within alignment gaps are excluded. With -n set to 1, sites with missing data are excluded. When an alignment includes a large indel or one or more sequences with large amounts of missing data, these options should be applied to the a first run of EDUT. It is important to note, however, that errors could be present within the sequenced regions that are excluded, and thus users should always plan to run EDUT a second time with all sites included.

Using EDUT Output

The primary purpose of EDUT is to identify individual base calls within haplotype data that are most likely to have contributed to errors in haplotype inference. In the example above, the highest corrected priority score is from nucleotide site 914 in sample 'Cbf3_28'. By inspecting the original sequence trace files for 'Cbf3_28', at position 914 and other flagged sites (e.g., 1426), it was

determined that the sample was heterozygous at a number of nucleotide sites and that those heterozygous sites had gone undetected during sequence assembly. This resulted in the inference of an incorrect haplotype. 'Cbf3_04' is also flagged several times in the EDUT output above. Experimental resolution of the haplotypes at both 'Cbf3_04' and 'Cbf3_28' through allele-specific PCR eliminated all flagged sites at *Cbf3*.

Other EDUT Output

If we'd like to see our locus as binary data, we could specify binary output on the command line, using option the option '-d'. Binary format can be useful for further analysis with R (http://www.r-project.org/) statistical environment or similar tools. In this example we have printed the data with no space between characters by using 'd none' to specify the delimiter as none. In the example below, we print the first three lines of the example data file (there are 23 more lines of data, but you get the idea). There is an insertion/deletion polymorphism on the 5' end of the sequenced region, thus the sequences shown begin with "NNN".

```
$ ./EDUT_1.1.pl -i example_fasta.txt -b -d none

NNN00000000000000000000000000000000000000010000000000000000000
NNN00000000000000000000000000000000000000010000000000000000000
NNNN00100000000000000000000000000000010000010000000000000000000
```

Effective Usage of EDUT

When applied to haplotype data, EDUT can help identify likely errors in haplotype inference. EDUT is most effective when source data is available, so that base calls can be verified. EDUT can be used in an iterative data verification process where errors in data sets are identified and experimentally resolved, and then the data set is tested again.

There are at least two considerations when preparing data files for EDUT that are important for getting good results. First, if your sequence alignment includes an outgroup, the outgroup sequence should be excluded from your

EDUT input file. Second, when a sample is homozygous for a single haplotype, include the haplotype only once in EDUT input. Including two copies of an incorrect haplotype will prevent EDUT from detected phasing errors.

To see a complete list of EDUT usage options, including the input of simulation data, execute EDUT without any command line arguments.

EDUT Version History

• **EDUT_1.01** - This was the first public release of EDUT. This version was fully functional, but the only supported input was aligned fasta files.
• **EDUT_1.02** - This version added support for coalescent simulations from Richard Hudson's ms (mksample) program. (http://home.uchicago.edu/~rhudson1/source/mksamples.html).

- **EDUT_1.1** - This version adds two new options that permit the exclusion of sites with alignment gaps (indicated by a '-' in the alignment) or missing data (indicated by an 'N'). When large alignments gaps or missing data are present, using these options will dramatically reduce processing time. However, excluding sites could mean that some errors in the data set are not detected.
- **EDUT_1.2** - This version places a comment character "#" in front of warning messages that would otherwise infer with reading binary data sets into R or similar environments.
- **EDUT_1.2.1 -** Fixes a problem with error messages printed when coalescent simulations are used as input.