POLY(1) **POLY(1)**

NAME

poly – DNA sequence polymorphism simulator

SYNOPSIS

poly [-s seed] [-F] specfile

DESCRIPTION

Poly reads from *specfile* the name of a commented FASTA format file containing a DNA sequence followed by a specification of how to mutate the DNA sequence. The resulting mutated sequence is written to the standard output in commented FASTA format. Normally, comments are output that detail the seeding of the random number generator and the exact manner in which the sequence was mutated. With the *–F* option set these comments are not output and a pure FASTA file is written.

The *–s* option permits one to specify a positive integer with which to seed the random number generator so that one can consistently generate the same data set if desired. Otherwise the process id of the particular invocation is used to see the random number generator, resulting in a distinct data set with each invocation.

Commented FASTA format is FASTA format in which one may additionally place lines that begin with a '#' between entries. These lines are all interpreted as comments and ignored. *Poly* reads such files and, of course, produces them as well.

1. Specification File Syntax and Semantics:

A *poly* input file must contain the name (including path if necessary) of a file from which to read a DNA sequence on its first line. The file name must constitute the entire line with no white-space on either side permitted. Subsequent lines contain mutation specifications, one per line. Formally the specification grammar is:

<Spec> ← <DNA: file name> <Mutation Operator> *

Each mutation operator line has a symbol specifying the operation in the first column of the line followed by the necessary arguments to the given operator. The syntax is as follows:

<Mutation Operator> ← 'S' <fraction>
'D' <minlen> '-' <maxlen> <fraction>
'X' <minlen> '-' <maxlen> <fraction>

The S-operator requests that the given *fraction* of the DNA sequence by subject to point mutations (SNPs if you will). The fraction must be a real value between 0 and 1, e.g., .01 requests that 1% of the sequence be permuted. The locations for the mutations are chosen with uniform probability across the target sequence. Generally, there need only be one such request in a specification.

The D-operator requests that the given *fraction* of the DNA sequence be deleted in blocks whose sizes are chosen uniformly from the interval *[minlen,maxlen]*, and from non-overlapping locations chosen uniformly across the target sequence. The X-operation requests that the given *fraction* of the DNA sequence be translocated in blocks of size uniformly chosen from *[minlen,maxlen]*. Both the source and destination coordinates for a translocation are chosen uniformly across the sequence.

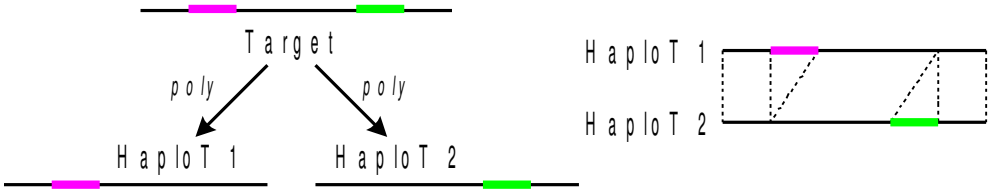
As an example, one might specify:

```
MyDNATarget
S     .0008
D 1-1 .00012
D 2-2 .00006
D 3-3 .00002
D  500- 1000 .00005
```

```
X 1000-20000 .00005
```

which will have the effect of inducing .1% modification in SNPs, 80% of which are substitution, 12% of which are 1-base deletions, 6% of which are 2-base deletions, and 2% of which are 3-base deletions. In addition, .05% of the genome will be deleted in blocks in the 500-1000 base-pair range, and another .05% will be translocated in blocks of size 1kbp to 20kbp.

Note that *poly* only deletes and does not insert sequence. The rationale behind this is that generally the target sequence has a rich repeat structure. Inserting random sequence does not preserve this characteristic. Therefore, we imagine that *poly* should be applied in the following way: one takes a given DNA sequence, and then applies *poly* to it repeatedly to generate a set of "haplotypes". Note that each haplotype then looks like it has inserted sequence wherever another had a segment deleted. That is we use an evolutionary model where the target sequence should be though of as the ancestral sequence.

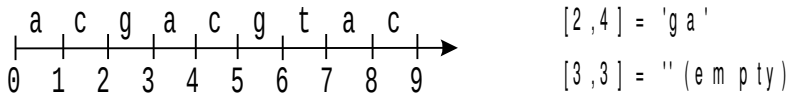


A final subtle point, is that *poly* first performs all deletion and translocation operations, and thereafter performs point mutations on the resulting, potentially, shorter sequence, at the specified rate. Thus, while deletion and translocation blocks are guaranteed not to overlap, substitutions do occur within translocated blocks.

2. Sample Execution:

We conclude with an example of the output of the *poly* program. We ran the specification above on a target DNA sequence that was 2 million bases long. *Poly* was run with neither option flag set, so that it generated a random seed (reported in the first comment) and prefaced its FASTA result with comments detailing the transformation that it performed. In order to shorten the listing, segments were deleted and such deletions are denoted by ellipses, '`...`'. If the *-F* option had been set then the comments would not appear, one would get just the initial '>'-line followed by the sequence.

The comments contain the seed for the run, followed by a rehash of the contents of the specification file. Then comes a sorted listing of the deleted and inserted blocks of the sequence, where a translocation has been mapped to a deletion and ensuing insertion. Note carefully that we use the convention that an index specifies a location between two characters as illustrated immediately below, thus [3280,3281] is the block containing the 3281'st symbol of the sequence. All coordinates are expressed relative to the target sequence.



After the deletion and insertion directives, there follows a sorted list of the characters that were point mutated as part of the SNP simulation. One has to be very careful here, to observe that the positions for SNPs, unlike those for deletions and translocations, are given with respect to the result sequence. So, for example, '`SNP at 251`', states that the 251'st character of the result was mutated from the orginal.

```
#
# Seed = 551
#
# Sequence from file: MyDNATarget
#
# Delete 0.012% of the genome in blocks of 1-1 bp
# Delete 0.006% of the genome in blocks of 2-2 bp
# Delete 0.002% of the genome in blocks of 3-3 bp
# Delete 0.05% of the genome in blocks of 500-1000 bp
# Translocate 0.05% of the genome in blocks of 1000-20000 bp
# SNP rate = 0.08%
#
# Did Structural Polymorphisms:
#   Delete [3280,3281]
#   Delete [10816,10817]
#   Delete [26306,26308]
#   Delete [33598,33599]
    ...
#   Delete [312295,312296]
#   Delete [313623,313625]
#   Delete [315414,315415]
#   Delete [326019,332370]
#   Delete [332520,332521]
#   Delete [335137,335138]
    ...
#   Delete [1966769,1966770]
#   Delete [1971571,1971572]
#   Insert [326019,332370] at 1986352
#   Delete [1987166,1987167]
#   Delete [1991981,1991982]
#   Delete [1993590,1993591]
#   Delete [1996702,1996703]
#
# Followed by SNPs:
#   SNP at 251
#   SNP at 3278
#   SNP at 4013
#   SNP at 6271
    ...
#   SNP at 1995678
#   SNP at 1996325
#   SNP at 1997269
#   SNP at 1998213
#
>
aacccacggtcatccccccggaggagcaattagggcacgcttacgagtat
ttgtcgacaccgatctaaccttcagctatggagagtactaaagaagtaga
gagcttacctccggctgctaagtctattatagctgacataagcaagtgcg
....
```

DIAGNOSTICS

Exit status is 0 if everything works normally, 1 if there is an error in the input, and 2, if there is not enough memory.

AUTHORS

Gene Myers:

Created October 12, ‘98

Last Revised November 8, ‘98