

# **msdup** (given p): readme file

Kosuke M. Teshima and Hideki Innan

kmteshima@kyudai.jp & innan\_hideki@soken.ac.jp

The Graduate University for Advanced Studies, Hayama, Kanagawa 240-0193, Japan.

Kyushu University, Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan

last update, March 31, 2015

## **1 msdup**

**msdup** is a simulation program to generate patterns of single nucleotide polymorphism (SNP) data in a region that involves duplicates and their flanking regions (Teshima and Innan, 2012). The software was developed by modifying the commonly used Hudson's **ms** simulator (Hudson, 1990). Please note that our program was modified from the previous version of **ms**.

Under the assumption that single-copy chromosomes and duplicate chromosomes are segregating in a population, **msdup** (given  $p$ ) generates polymorphism data conditional on the current frequency of the duplicate chromosomes. **msdup** can incorporate the effect of selection acting on the duplicate chromosomes, as well as mutation, recombination and interlocus gene conversion.

### **1.1 Download and compilation**

All files are included in the `source_msdup_given_p.tar.gz` file, which can be downloaded at <http://www.sendou.soken.ac.jp/esb/innan/InnanLab>. The source code of the program is written in C and this program is intended to be run on UNIX, or a UNIX-like operating systems, such as Linux or Mac OS X.

Download the tar file onto your machine and extract it with:

```
tar xzvf source_msdup_given_p.tar.gz
```

After extraction, change the directory by typing:

```
cd source_msdup_given_p
```

Then, compile the program by typing:

```
gcc -o msdup ms_dup_give_p.c streec.c backward_trajectory.c  
rand_mt32.c -lm
```

## 1.2 The basic command line

The following command line shows the usage of `msdup`.

```
./msdup nsam p_dup howmany
-t 4N_0u
-r 4N_0r nsites
-g 4N_0g l_tract
-s 4N_0s h
-l l_left l_dup l_mid l_dup l_right
```

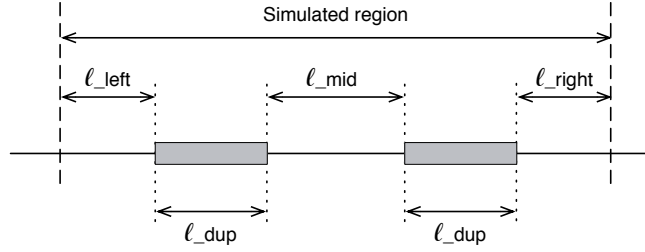


Figure 1: The length of each region. Note that (the length of the simulated region) =  $nsites = l_{left} + l_{dup} + l_{mid} + l_{dup} + l_{right}$ .

The following table is a summary of arguments that must be specified to run `msdup`.

switch	argument	
	<i>nsam</i>	number of chromosomes in the sample.
	<i>p_dup</i>	current frequency of the duplicated chromosome.
	<i>howmany</i>	number replication.
-t	$4N_0u$	population mutation parameter per simulated region.
-r	$4N_0r$	population recombination parameter per simulated region.
	<i>nsites</i>	the length of the simulated region (bp).
-g	$4N_0g$	$4N_0g$ is the initiation rate of gene conversion per $l_{dup}$ .
	<i>l_tract</i>	average length of a conversion tract (bp).
-l	<i>l_left</i>	the length of the 5' flanking region.
	<i>l_dup</i>	the length of the duplicated region (original copy).
	<i>l_mid</i>	the length between the original and duplicated regions.
	<i>l_dup</i>	the length of the duplicated region (duplicate copy, this should be equal to the length of the original copy).
	<i>l_right</i>	the length of the 3' flanking region.
-s	$4N_s$	selection coefficient for duplicated chromosomes.
	<i>h</i>	coefficient of dominance.

### 1.2.1 example command line

The following is an example command line:

```
./msdup 20 0.9 100 -t 13 -r 13 13000 -g 1 100  
-l 1000 1000 9000 1000 1000 -s 4 0.5
```

In this setting, the program will output polymorphism data in a 13,000 bp region. The lengths of regions are: (left, original copy, middle, duplicated copy, right) = (1000, 1000, 9000, 1000, 1000). The sample size is  $nsam = 20$ , and  $4N_0u = 4N_0r = 13$  (per simulated region). 100 replications are performed.  $4Ns$  for the duplicated chromosome is 4 and the dominant coefficient is 0.5.

Let  $c$  be the per site interlocus conversion rate and  $C = 4Nc$ .  $C$  is given by  $C = G \times \ell$ , where  $G = 4N_0g$  is the initiation rate of gene conversion and  $\ell$  is the average tract length. In the above command line, the ratio of the conversion rate to mutation rate can be calculated as

$$c/\mu = \frac{1 \times 100}{13 \times (1000/13000)} = 100$$

### 1.3 The output

An output of the example command line in Section 1.2.1,

```
./msdup 20 0.9 100 -t 13 -r 13 13000 -g 1 100  
-l 1000 1000 9000 1000 1000 -s 4 0.5
```

is shown in Figure 2.

The first line shows the command line you have just typed. The output of each replication starts with a line with '//', followed by "replication:" and the replication number. In the second line, the number of segregating sites are shown. The first number is the total number of segregating sites across the simulated region. The following five numbers mean the numbers of segregating sites in 5', original copy, middle, duplicated copy and 3' regions, respectively. The third and the following lines show the relative positions of segregating sites. The state of SNP at each segregating site is shown with 0 and 1. 0 means ancestral and 1 means derived allele. Spaces are inserted to show boundaries between regions. For single-copy chromosomes, the region of the duplicated copy is filled with "-", which means there is no duplicated copy on the chromosome. Note that at least one site is printed out even when there is no segregating sites in the region. In such cases, all samples have the same allele 0 at the site (eg. 3' region of Figure 2).

```

$ ./msdup 20 .9 2 -t 13 -r 13 13000 -g 1 100 -l 1000 1000 9000 1000 1000 -s 4 0.5
//msdup 20 .9 2 -t 13 -r 13 13000 -g 1 100 -l 1000 1000 9000 1000 1000 -s 4 0.5
Total number of segregating sites
First replication
//replication: 1
//segsites: 50 : 2 {4} 44 {4} 0
//position:
// 0.04178 0.07612 0.09107 0.09232 0.10377 0.15372 0.16589 0.16609 0.16715 0.19735
// 0.21948 0.25291 0.26914 0.27457 0.29138 0.29973 0.31664 0.35117 0.36556 0.37012
// 0.38430 0.40251 0.41136 0.41327 0.42707 0.42740 0.43717 0.48038 0.49421 0.49667
// 0.51862 0.52769 0.52930 0.54278 0.54640 0.61658 0.62133 0.64699 0.66090 0.68954
// 0.69482 0.70009 0.75390 0.78244 0.78763 0.78766 0.80620 0.80707 0.81995 0.82443
// 0.86030 0.86155 0.87300 0.92295
00 0000 0001010100001010010100000110000000001000000 --- 0
00 0000 0001010100001010010100000110000000001000000 --- 0
10 0000 0100001010011100100001011100000010000000000 0000 0
10 0000 010111000000101001010000011100000000010101 0000 0
00 0000 000101000000001000010100000100000000000000 0000 0
00 0000 1000001010011100000001011000000100000000000 0000 0
00 0000 000000101001110000000101010000010010000000 0010 0
10 0000 010111000000101001010000011100000000010101 0000 0
10 0000 000000101001110000000101010000010100000000 0100 0
00 0000 000101000000001000010100000100000000000000 0000 0
00 0000 000101010000101001010000011001000000010101 0000 0
00 0001 00000010110111000010000000000001100010000010 0000 0
00 1000 000000101011101001010000010000000000000000 0000 0
10 0000 000000101001110000000101010000010100000000 0100 0
10 0000 010111000000101001010000011100000000010101 0000 0
01 0000 0000001010011100000001010100000100000100010 0000 0
10 0000 010111000000101001010000011100000000010101 0000 0
00 0000 00100010100111000000010101000010110000001000 0000 0
SNPs in the 5' region
SNPs in the middle region
SNPs in the 3' region
Single-copy chromosome
Duplicate chromosome
SNPs in the original copy
SNPs in the duplicated copy
//replication: 2 Second replication
//segsites: 32 : 1 {6} 21 {6} 4
//position:
// 0.01012 0.07942 0.08218 0.09699 0.10440 0.12689 0.13818 0.20026 0.21646 0.28563

```

Figure 2: An example output of msdup.

## 1.4 Some notes

### 1.4.1 Sample configuration

The number of single-copy chromosomes and duplicate chromosomes are determined based on the frequency of the duplicate chromosomes. This means the sample configuration is given by

$$\text{The number of single-copy chromosome} = nsam - \text{int}(nsam \times p_{dup})$$

$$\text{The number of duplicate chromosome} = \text{int}(nsam \times p_{dup})$$

This default configuration is assigned in `gerpars()` function. You can change the assignment by modifying the code (1.170–188) in `segtre_mig_given_p()` function of `strec.c`. There are some example code there.

### 1.4.2 Trajectory of the duplicate chromosome

The trajectory of the frequency of duplicate chromosomes is generated backward in time for each replication. To simulate trajectory, we used pseudosampling method developed by Kimura and Takahata (1983).

### 1.4.3 Random number generator

Mersenne Twister (32bit version) is used to generate random numbers (Matsumoto and Nishimura, 1998).

## References

- Hudson, R. (1990). Gene genealogies and the coalescent process. In D. Futuyma and J. Antonovics, editors, *Oxford Surveys in Evolutionary Biology*, volume 7, pages 1–44. Oxford University Press.
- Kimura, M. and Takahata, N. (1983). Selective constraint in protein polymorphism: study of the effectively neutral mutation model by using an improved pseudosampling method. *Proc Natl Acad Sci USA*, **80**, 1048–52.
- Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **8**(1), 3–30.
- Teshima, K. M. and Innan, H. (2012). The coalescent with selection on copy number variants. *Genetics*, **190**(3), 1077–1086.

## Release note

- March, 2015:
  - Readme file updated.
- Jan, 2012:
  - The first release.