```
#!/usr/bin/perl -w
# calcualte methylation status using Lister et al. 2009 method
# do not separate different position types (CG, CHG, CHH)
use strict;
use Math::CDF qw(:all);
my $FDR = 0.01;
my $print_values = 0;
my $usage = "$0 <meth file> <error rate> <max_pos_id>";
die $usage unless(@ARGV >= 2);
my ($meth_file, $rate) = @ARGV[0..1];
my $meth_max = 0;
my @accu_meth;
my @total_meth;
my @meth_level;
my $max_pos_id = 42859753; #don't want to include chrC, chrM, pUPC19
if(@ARGV > 2){
        $max_pos_id = $ARGV[2];
}
open(MF, $meth_file) or die "Can't open $meth_file: $!";
while(<MF>){
        next if (/SampleID/);
        chomp;
        my ($sample_id, $pos_id, $depth, $num_C, $percent, $type) = split /\t/;
        next if($pos_id > $max_pos_id);
                #$meth_level{$type}->[$depth]->[$num_C]++;
                $meth_level[$depth]->[$num_C]++;
                #$total_meth{$type}->[$depth]++;
                $total_meth[$depth]++;
                #if(!defined $meth_max{$type}){
                #       $meth_max{$type} = $depth;
                #}elsif($meth_max{$type} < $depth){
                if($meth_max < $depth){
                        $meth_max = $depth;
                }
                #}
}
close MF;

# calculate accumulated number of positions with <= k unconverted Cs
#foreach my $t(keys %meth_max){
        foreach my $i(0..$meth_max){
                if(!defined $meth_level[$i]->[0]){
                        $meth_level[$i]->[0] = 0;
                }
                $accu_meth[$i]->[0] = $meth_level[$i]->[0];
                next if($i == 0);
                foreach my $j(1..$i){
                        if(!defined $meth_level[$i]->[$j]){
                                $meth_level[$i]->[$j] = 0;
                        }
                        $accu_meth[$i]->[$j] = $accu_meth[$i]->[$j-1] +
                             $meth_level[$i]->[$j];
                }
        }

print STDERR "File: $meth_file, Conversion Error: $rate\n";
print STDERR "Max depth: $meth_max\n";
```

```
#foreach my $t(sort keys %meth_max){
#       print STDERR $t, " ", $meth_max{$t}, " ";
#}
#print STDERR "\n";

#my $max_depth = 18;
# calculate at each depth, how many positions have k unconverted C
if($print_values){
    print "Number of positions with k unconverted Cs at each depth\n";
}
#foreach my $t(sort keys %meth_max){
#       my $max_depth = $meth_max{$t};
        if($print_values){
#       print $t;
    foreach my $i(0..$meth_max){
                print "\t", $i;
        }
        print "\n";
        }
        foreach my $i(0..$meth_max){
                if($print_values){
                    print $i;
                }
                foreach my $j(0..$i){
                #       if(!defined $meth_level{$t}->[$i]->[$j]){
                #               $meth_level{$t}->[$i]->[$j] = 0;
                #       }
                        if($print_values){
                           print "\t", $meth_level[$i]->[$j], "|",
                           $accu_meth[$i]->[$j];
                        }
                }
                if($print_values){
                    print "\n";
                }
        }


undef @meth_level;

# find cutoff at each depth level
my @cutoff;
if($print_values){
    print "Lister equation values\n";
}
#foreach my $t(sort keys %meth_max){
        #my $max_depth = $meth_max{$t};
        if($print_values){
         #   print $t;
    foreach my $i(0..$meth_max){
                print "\t", $i;
        }
        print "\n";
        }

        foreach my $i(0..$meth_max){
                next if(!defined $total_meth[$i]);
                if($print_values){
                    print $i;
                }
                if($i < 2){
                        if($print_values){
                            print "\t0";
                        }
                        $cutoff[$i] = $i+1;
                        #next;
                }else{
                        $cutoff[$i] = 4;
                foreach my $j(1..$i){

                        my $prob = pbinom($j, $i, $rate);
                        #if($j > 0){
                                $prob = $prob - pbinom($j-1, $i, $rate);
                        #}
                        my ($num_unC, $num_mC) = (0,0);
                        $num_unC = $accu_meth[$i]->[$j-1];
                        if(!defined $num_unC){
                                $num_unC = 0;
                        }
                        $num_mC = $total_meth[$i] - $num_unC;
                        my $left = $prob * $num_unC;
                        my $right = $FDR * $num_mC;
                        if($left < $right){
                                $cutoff[$i] = $j;
                                last;
                        }
                        if($print_values){
                            print "\t", $left, "|", $right;
                        }
                }
                if($print_values){
                    print "\n";
                }
        }
        }


undef @total_meth;
if($print_values){
    print "Cutoff values at each depth\n";
}
#my $max_depth = 100;

#foreach my $t(sort keys %meth_max){
#       my $max_depth = $meth_max{$t};
        #if($print_values){
            #print STDERR $t, "\n";
        #}
    foreach my $i(0..$meth_max){
                if($print_values){
                print $i;
                if($i != $meth_max){
                        print "\t";
                }
                }
        }
        if($print_values){
            print "\n";
        }
        if($print_values){
        foreach my $i(0..$meth_max){

                if(defined $cutoff[$i]){
                        print $cutoff[$i];
                }else{
                        print "0";
                }
                if($i != $meth_max){
                        print "\t";
                }
        }
        print "\n";
        }

my %num_pos;
open(MF, $meth_file) or die "Can't open $meth_file: $!";
#seek(MF, 0, 0) or die "Can't seek to beginning of file: $!";
while(<MF>){
        chomp;
        if (/SampleID/){
                print;
                print "\tisMeth\n";
                next;
        }

        my ($sample_id, $pos_id, $depth, $num_C, $percent, $type) = split /\t/;
        next if($pos_id > $max_pos_id);
        my $isMeth = 0;
                if($depth >= 2 && $num_C >= 1 && $num_C >= $cutoff[$depth]){
                        $isMeth = 1;
                        $num_pos{$type}++;
                }
                print join("\t", ($sample_id, $pos_id, $depth, $num_C, $percent, $type,
                    $isMeth)), "\n";

}
close MF;

print STDERR "Total mC position in ", $meth_file, "\n";
my $total = 0;
foreach my $t(sort keys %num_pos){
    print STDERR "$t: ", $num_pos{$t}, "\t";
        $total += $num_pos{$t};
}
print STDERR "Total: $total\n";
```