

```
use strict;
use warnings;

my $l = $ARGV[0];

# Read in sequences
my @DNA;
open (M, "motifs.txt") || die "Cannot open motifs.txt: $!\n";
while (<M>) {
    chomp;
    my @seq = split //;
    push @DNA, \@seq;
    #last if @DNA == 2; # only for testing
}
close (M);

# Enumerate all combinations

my @n;
foreach (@DNA) {
    push @n, ${$_}-$l;
}

my @s;
foreach (@n) {
    push @s, 0;
}

my $best_score = 0;
my @best_s;

while (1) {

    my $score = score(\@DNA, \@s, $l);
    print "@s: $score\n";
    if ($score > $best_score) {
        $best_score = $score;
        @best_s = @s;
    }

    for (my $i = $#n; $i >= 0; $i--) {
        if ($s[$i] == $n[$i]) {
            $s[$i] = 0;
        } else {
            $s[$i]++;
            last;
        }
    }

    # Back to 0,0,0, ...
    my $back = 1;
    foreach (@s) {
        if ($_ != 0) {
            $back = 0; last;
        }
    }
    last if $back;
}

print "\n\nBEST: @best_s: $best_score\n\n";

for (my $j = 0; $j < @DNA; $j++) {
    for (my $i = 0; $i < $l; $i++) {
        print "$DNA[$j][$best_s[$j]+$i]";
    }
}
```

```
    print "\n";
}

sub score {

    my @DNA = @{$_[0]};
    my @s   = @{$_[1]};
    my $l   = $_[2];

    my $score = 0;
    for (my $i = 0; $i < $l; $i++) {
        my %base;
        for (my $j = 0; $j < @DNA; $j++) {
            $base{$DNA[$j][$s[$j]+$i]} ++;
        }

        my $consensus = 0;
        foreach (keys %base) {
            if ($base{$_} > $consensus) {
                $consensus = $base{$_};
            }
        }

        $score += $consensus;
    }

    return $score;
}
```

```

use strict;
use warnings;

my $l = $ARGV[0];

# Read in sequences
my @DNA;
open (M, "motifs.txt") || die "Cannot open motifs.txt: !\n";
while (<M>) {
    chomp;
    my @seq = split //;
    push @DNA, \@seq;
}
close (M);

my @s;
foreach (@DNA) {
    push @s, 0;
}

my @best_s = @s;
my $best_score = 0;

for ($s[0] = 0; $s[0] < @{$DNA[0]}-$l; $s[0]++) {
    for ($s[1] = 0; $s[1] < @{$DNA[1]}-$l; $s[1]++) {
        my $score = score(\@DNA, \@s, $l, 2);
        if ($score > $best_score) {
            $best_s[0] = $s[0];
            $best_s[1] = $s[1];
            $best_score = $score;
        }
    }
}

$s[0] = $best_s[0];
$s[1] = $best_s[1];

for (my $i = 2; $i < @DNA; $i++) {
    $best_score = 0;
    for ($s[$i] = 0; $s[$i] < @{$DNA[$i]}-$l; $s[$i]++) {
        my $score = score(\@DNA, \@s, $l, $i+1);
        if ($score > $best_score) {
            $best_s[$i] = $s[$i];
            $best_score = $score;
        }
    }
    $s[$i] = $best_s[$i];
}

print "\n\nBEST: @best_s: $best_score\n\n";

for (my $j = 0; $j < @DNA; $j++) {
    for (my $i = 0; $i < $l; $i++) {
        print "$DNA[$j][$best_s[$j]+$i]";
    }
    print "\n";
}

sub score {
    my @DNA    = @{$_[0]};
    my @s      = @{$_[1]};
    my $l      = $_[2];
    my $no_seq = $_[3];

```

```
my $score = 0;
for (my $i = 0; $i < $l; $i++) {
    my %base;
    for (my $j = 0; $j < $no_seq; $j++) {
        $base{$DNA[$j][$s[$j]+$i]} ++;
    }

    my $consensus = 0;
    foreach (keys %base) {
        if ($base{$_} > $consensus) {
            $consensus = $base{$_};
        }
    }

    $score += $consensus;
}

return $score;
}
```

```
use strict;
use warnings;

my @v = ("","T","G","C","A","T","A","C");
my @w = ("","A","T","C","T","G","A","T","C");

my $n = @v;
my $m = @w;

my @s;

$s[0][0] = " ";
for (my $i = 1; $i < $n; $i++) {
    $s[$i][0] = 0;
}

for (my $j = 1; $j < $m; $j++) {
    $s[0][$j] = 0;
}

for (my $i = 1; $i < $n; $i++) {
    for (my $j = 1; $j < $m; $j++) {
        my @max = ($s[$i-1][$j], $s[$i][$j-1]);
        if ($v[$i] eq $w[$j]) {
            push @max, $s[$i-1][$j-1]+1;
        }
        @max = sort {$b <=> $a} @max;
        $s[$i][$j] = $max[0];
    }
}

foreach (@s) {
    print "@{$_}\n";
}

print "Longest common substring: $s[$n-1][$m-1]\n";
```