Tutorial

December 8, 2020

1 KwARG Tutorial

This tutorial gives an overview of the usage of KwARG and the types of output that it can produce.

To recreate the results on the command line, open Terminal and navigate to the directory where you have saved the binaries. Remove the '!' from the commands below.

1.1 Available options

A list of the available options can be viewed by running KwARG with the '-h', '-H' or '-?' option:

[1]:
```
!./kwarg -h
```

```
Usage: ./kwarg [options] < [input]
The program reads data from the input file specified and greedily
constructs a history with a low number of recombinations and
recurrent mutations (homoplasies). The history is constructed by
stepping backwards in time using coalescence, mutation and
recombination events. At each point in this process, all possible
next events (strictly speaking, only a useful subset of all possible
next events) are considered, and the resulting ancestral states are
scored. The scores are used to choose an event either at random or
to proceed to an ancestral state with minimum score (see option -T).
This process is NOT guaranteed to lead to a history with a minimum
number of recombinations or the minimum number of homoplasies.
Legal options are:
 -L[x] Provide an optional label x (should be an integer) to print at
       the start of each line.
 -S[x] Specify cost of a sequencing error (default: x = 0.5).
 -M[x] Specify cost of a recurrent mutation (default: x = 0.9).
 -R[x] Specify cost of a single recombination (default: x = 1.0).
 -C[x] Specify cost of two recombinations immediately following each
       other (default: x = 2.0).
 -T[F] Set annealing temperature 0 < F < 700 (default: F = 30).
       Scores are normalised to lie between 0 and 1, and the next
       step selected with probability proportional to exp(F *
       normalised_score). Setting F = 0 corresponds to random
       selection among possible moves (not recommended). Setting F
```

```
      = -1 corresponds to setting F = Infinity, and will force
      selection of the move with the minimum score at each step.
-V[x] If running a single iteration with given cost parameters, this
      controls the level of verbosity.
      x = 0: no extra output
      x = 1: during each neighbourhood search, output the number of
      neighbours explored, the move selected and its cost
      x = 2: during each neighbourhood search, output the number of
      neighbours explored, the resulting configuration and cost of
      each neighbour, the move selected and its cost.
-b[name] Output a minimum recombination history to file name.
-d[name] Output ancestral recombination graph of minimum
         recombination history in dot format to file name.
-g[name] Output ancestral recombination graph of minimum
         recombination history in GDL format to file name.
-j[name] Output ancestral recombination graph of minimum
         recombination history in GML format to file name.
-t[name] Output list of marginal phylogenies for each site in
         Newick's 8:45 format to file name.
-D[name] Output list of marginal phylogenies for each site in dot
         format to file name.
-G[name] Output list of marginal phylogenies for each site in GDL
         format to file name.
-J[name] Output list of marginal phylogenies for each site in GML
         format to file name.
-I Marginal trees are only output one for each of the intervals
   between two recombination points, instead of one for each site.
-v[nodelabel] Use nodelabel convention for labelling nodes in
              ancestral recombination graphs. The possible
              conventions are: nodelabel =
              'none': do not label nodes
              'id': only label nodes representing sampled
              sequences, using their sequence ids from the data
              file, and nodes representing recombinations,
              indicating the recombination point
              'sequence': label nodes with the inferred sequences;
              these sequences will be in binary format, with 0
              representing wild type and 1 representing mutant
              type, even if the original data is not in binary
              format
              'both': label nodes with both id and inferred
              sequence
              'one': use only one label for a node, sequence id or

              inferred sequence
              Default convention is id. The colour coding scheme
              used for the nodes is red for sequences in the input
              data, blue for recombination nodes, green for
              standard coalescent nodes, and yellow for the final
              coalescence into the most recent common ancestor.
-i Sequences not having a sequence id in the data file are assigned
   their index in the data file as id, e.g. the first sequence in
   the data file would be assigned '1' as id.
-e Label edges in ancestral recombination graphs with the sites
   undergoing mutation along the edge.
-o Assume input data is in own format. Default is to first try to
   parse data in own format, and if that fails to try to parse it
   in fasta format. Specifying this option, no attempt will be made
   to try to parse the data in fasta format.
-f Assume input data is in fasta format. No attempt will be made to
   try to parse the data in own format. Note that the -o and
   the -f options override each other, so only the last one
   occurring in the command line will have an effect.
-a Assume input consists of amino acid (protein) sequences using the
   one letter amino acid alphabet; anything not in the amino acid
   one letter alphabet is treated as an unresolved site (default is
   to assume sequences in binary, i.e. 0/1, format where anything
   but a 0 or a 1 is considered an unresolved site). If the most
   recent common ancestor is assumed known (see option -k), the
   first sequence in the input data is considered to specify the
   wild type of each site and is not included as part of the sample
   set.
-n Assume input consists of nucleic sequences; anything but
   a/c/g/t/u is considered an unresolved site (default is to assume
   binary, i.e. 0/1, format where anything but a 0 or a 1 is
   considered an unresolved site). If the most recent common
   ancestor is assumed known (see option -k), the first sequence in
   the input data is considered to specify the wild type of each
   site and is not included as part of the sample set.
-k Assume that the common ancestral sequence is known, i.e. that we
   know which is the wild type and which is the mutant in each
   site. If the data is in binary format, the all-0 sequence is
   assumed to be the common ancestral sequence (this does not need
   to be present in the data). If the data is in amino acid or
   nucleotide format, the common ancestral sequence has to be
   specified directly and is taken to be the first sequence in the
   data file (see options -a and -n)
-Q[x] Sets the number of runs.
-Z[x] Sets the random seed z (only one run is made in this case).
-X[x] Provide an upper bound x on the number of recombinations
      needed for the input dataset (solutions with more than x
      recombinations will be abandoned).
-s Turns off the header row of the results table.
-h, -H -? Print this information and stop.
```

1.2 Running KwARG

We will first run KwARG on the Kreitman dataset, with 3 iterations for each of the default cost parameters:

[2]:
```
!./kwarg -Q3 < kreitman_snp.txt
```

```
          Seed    Temp   SE_cost   RM_cost    R_cost   RR_cost  SE  RM   R   N_states
Time
    2601625704    30.0     -1.00     -1.00      1.00      2.00   0   0   7        153
0.01203100
    2833715205    30.0     -1.00     -1.00      1.00      2.00  NA  NA  NA        175
0.01934900
    2088249328    30.0     -1.00     -1.00      1.00      2.00   0   0   7        145
0.01337600
    3052951246    30.0      1.00      1.01      1.00      2.00   6   1   6       1444
0.57676100
    3353384182    30.0      1.00      1.01      1.00      2.00   4   1   4       1255
0.20606800
    3410089108    30.0      1.00      1.01      1.00      2.00   1   0   5        760
0.11520900
    1828302667    30.0      0.90      0.91      1.00      2.00   5   0   2        702
0.14626900
    3225485139    30.0      0.90      0.91      1.00      2.00  NA  NA  NA       1083
0.14772300
    3046884879    30.0      0.90      0.91      1.00      2.00  NA  NA  NA        849
0.09440700
    3198632652    30.0      0.80      0.81      1.00      2.00  NA  NA  NA       1157
0.21403300
    1986441690    30.0      0.80      0.81      1.00      2.00   3   0   4        914
0.10946100
    1773048665    30.0      0.80      0.81      1.00      2.00   2   1   4        780
0.11172300
    1838291634    30.0      0.70      0.71      1.00      2.00  NA  NA  NA       1167
0.15020400
    2143237569    30.0      0.70      0.71      1.00      2.00  NA  NA  NA       1086
0.15805000
    2303470289    30.0      0.70      0.71      1.00      2.00  NA  NA  NA        945
0.16666900
    2645221831    30.0      0.60      0.61      1.00      2.00   3   0   3        747
0.11292900
    3243300085    30.0      0.60      0.61      1.00      2.00   0   2   4        744
0.07948900
    1864865057    30.0      0.60      0.61      1.00      2.00   2   1   3        699
0.10499500
    2547814861    30.0      0.50      0.51      1.00      2.00  NA  NA  NA        942
0.09895000
    1921909335    30.0      0.50      0.51      1.00      2.00   3   0   3        822
```

```
0.11927000
   3185200706   30.0      0.50      0.51      1.00      2.00   5   0   2          820
0.10595400
   1627110433   30.0      0.40      0.41      1.00      2.00  NA  NA  NA         1010
0.22907700
   3056188657   30.0      0.40      0.41      1.00      2.00  NA  NA  NA          894
0.09132400
   1914316890   30.0      0.40      0.41      1.00      2.00   5   0   2          934
0.11731400
   3310285152   30.0      0.30      0.31      1.00      2.00  NA  NA  NA          931
0.18293100
   2830896468   30.0      0.30      0.31      1.00      2.00   5   0   2          715
0.09527100
   1765279040   30.0      0.30      0.31      1.00      2.00   5   0   2          723
0.10761400
   3393915342   30.0      0.20      0.21      1.00      2.00   5   0   2          723
0.10024600
   3187940384   30.0      0.20      0.21      1.00      2.00  NA  NA  NA          868
0.08212500
   2780186279   30.0      0.20      0.21      1.00      2.00  13   0   0          836
0.25446000
   2378498273   30.0      0.10      0.11      1.00      2.00   5   0   2          702
0.09103200
   1683804573   30.0      0.10      0.11      1.00      2.00  13   0   0          820
0.28792300
   2844859340   30.0      0.10      0.11      1.00      2.00   5   0   2          702
0.09583900
   3245102482   30.0      0.01      0.02      1.00      2.00   5   0   2          702
0.09950500
   1961747575   30.0      0.01      0.02      1.00      2.00   9   0   1          889
0.10874100
   2780771945   30.0      0.01      0.02      1.00      2.00   7   5   0          736
0.09567800
   2276621186   30.0      1.00      1.10     -1.00     -1.00  NA  NA  NA         1169
0.21339400
   2917570376   30.0      1.00      1.10     -1.00     -1.00  NA  NA  NA         1432
0.20109900
   2549774585   30.0      1.00      1.10     -1.00     -1.00  NA  NA  NA         1589
0.25289100
```

The output table gives: - Seed: the random seed needed to rerun this particular iteration (demonstrated below) - Temp: the annealing temperature used (default: 30) - SE_cost, RM_cost, R_cost, RR_cost: the cost parameter for a sequencing error, recurrent mutation, single recombination, and two consecutive recombination events, respectively - SE, RM, R: the number of each type of event in the solution - N_states: the total number of states considered when constructing one-step neighbourhoods - Time: CPU time for each iteration.

Lines where number of events is shown as 'NA' correspond to runs which were identified as sub-optimal and terminated before completion.

KwARG can also be run with specified costs, as follows:

[3]:
```
!./kwarg -S0.2,0.4 -M0.3,0.5 -Q5 < kreitman_snp.txt
```

```
          Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   2611612860   30.0     0.20     0.30     1.00     2.00   8   2   0        715
0.09174800
   2186755964   30.0     0.20     0.30     1.00     2.00   6   0   2        813
0.12789900
   3641081466   30.0     0.20     0.30     1.00     2.00  NA  NA  NA       1297
0.13407100
   2458405652   30.0     0.20     0.30     1.00     2.00   5   0   2        702
0.10066300
   2963344908   30.0     0.20     0.30     1.00     2.00  NA  NA  NA        937
0.11827100
   2681191801   30.0     0.40     0.50     1.00     2.00   4   1   2        770
0.08341100
   2159394232   30.0     0.40     0.50     1.00     2.00  NA  NA  NA       1088
0.12439800
   2526916363   30.0     0.40     0.50     1.00     2.00  NA  NA  NA        814
0.29432900
   2930465543   30.0     0.40     0.50     1.00     2.00  NA  NA  NA        792
0.11778200
   2775244447   30.0     0.40     0.50     1.00     2.00  NA  NA  NA       1038
0.09858000
```

The same number of SE_cost and RM_cost parameters should be specified, separated by commas.

The options '-R' and '-C' can be used to specify the cost of single and double recombination events, this defaults to 1.0 and 2.0, respectively. If these options are used, then SE_cost and RM_cost must also be provided.

1.2.1 Input formats

Binary data The input data can be in 0/1 binary format, and any other symbols will be interpreted as missing data.

[4]:
```
!cat example_data_1.txt
```

```
00010010
1001101x
11001x01
01100101
```

Sequence and site labels can also be provided, as follows:

[5]:
```
!cat example_data_2.txt
```

```
#> Seq1
00010010
#> Seq2
1001101-
#> Seq3
11001-01
#> Seq4
01100101
#positions: 2 3 7 9 10 11 13 14
```

FASTA format Input data in fasta format can be provided, for instance:

```
[6]: !cat example_data_3.fasta
```

```
>Seq1
ACTTAAGG
>Seq2
TCTTTAGN
>Seq3
TGTATNAA
>Seq4
AGCAATAA
```

In this case, to denote that the data is provided in fasta format in nucleotide representation, the '-f' and '-n' flags should be used:

```
[7]: !./kwarg -f -n -S0.5 -M0.9 < example_data_3.fasta
```

```
         Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   2616606438   30.0     0.50     0.90     1.00     2.00   2   0   0         36
0.00114700
```

1.2.2 Annealing parameter

We can change the annealing temperature using the '-T' option: - '-T30' is the default temperature of 30. - '-T0' means the next step is chosen uniformly at random among all available moves (this is not recommended). - '-T-1' signifies $T = \infty$, so the next step is chosen among all available moves with the minimum score.

```
[8]: !./kwarg -T-1 -Q3 < kreitman_snp.txt
```

```
         Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   2616606438   -1.0    -1.00    -1.00     1.00     2.00   0   0   7        146
0.01183900
   1863276342   -1.0    -1.00    -1.00     1.00     2.00   0   0   7        146
0.01072300
```

```
   1634361799   -1.0    -1.00    -1.00     1.00     2.00   0   0   7         167
0.01147600
   2120612875   -1.0     1.00     1.01     1.00     2.00   1   0   6         797
0.10581000
   2768325270   -1.0     1.00     1.01     1.00     2.00   1   0   6         797
0.09598300
   3390484970   -1.0     1.00     1.01     1.00     2.00   1   0   6         797
0.09088400
   3398681837   -1.0     0.90     0.91     1.00     2.00   3   0   3         773
0.09291300
   2177631974   -1.0     0.90     0.91     1.00     2.00   3   0   3         773
0.10779600
   2872255873   -1.0     0.90     0.91     1.00     2.00   3   0   3         773
0.10341900
   3637292166   -1.0     0.80     0.81     1.00     2.00   3   0   3         789
0.09916600
   2229965343   -1.0     0.80     0.81     1.00     2.00   3   0   3         789
0.09505000
   2546633691   -1.0     0.80     0.81     1.00     2.00   3   0   3         789
0.09396800
   2918284069   -1.0     0.70     0.71     1.00     2.00   3   0   3         789
0.09800800
   2944500523   -1.0     0.70     0.71     1.00     2.00   3   0   3         789
0.09861100
   1872404186   -1.0     0.70     0.71     1.00     2.00   3   0   3         789
0.10707600
   1710428850   -1.0     0.60     0.61     1.00     2.00   3   0   3         789
0.09456800
   1991187006   -1.0     0.60     0.61     1.00     2.00   3   0   3         789
0.09400000
   2011993707   -1.0     0.60     0.61     1.00     2.00   3   0   3         789
0.09674300
   2817200563   -1.0     0.50     0.51     1.00     2.00   5   0   2         791
0.10336500
   2265418323   -1.0     0.50     0.51     1.00     2.00   5   0   2         791
0.11140600
   2065884106   -1.0     0.50     0.51     1.00     2.00   5   0   2         791
0.10161900
   2194177202   -1.0     0.40     0.41     1.00     2.00   5   0   2         723
0.13337700
   3484274397   -1.0     0.40     0.41     1.00     2.00   5   0   2         723
0.11459100
   2622644005   -1.0     0.40     0.41     1.00     2.00   5   0   2         723
0.12060700
   2837983743   -1.0     0.30     0.31     1.00     2.00   5   0   2         723
0.10573600
   2572373776   -1.0     0.30     0.31     1.00     2.00   5   0   2         723
0.10198800
```

```
   3485362873   -1.0      0.30      0.31      1.00      2.00   5   0   2         723
0.10146700
   1941959837   -1.0      0.20      0.21      1.00      2.00   5   0   2         723
0.10453900
   3690368037   -1.0      0.20      0.21      1.00      2.00   5   0   2         723
0.10957000
   3011058638   -1.0      0.20      0.21      1.00      2.00   5   0   2         723
0.10870000
   2981580430   -1.0      0.10      0.11      1.00      2.00   5   0   2         702
0.10073700
   3621259533   -1.0      0.10      0.11      1.00      2.00   5   0   2         702
0.09977600
   2285808715   -1.0      0.10      0.11      1.00      2.00   5   0   2         702
0.09485400
   3548241670   -1.0      0.01      0.02      1.00      2.00   5   0   2         702
0.09851200
   3359367899   -1.0      0.01      0.02      1.00      2.00   5   0   2         702
0.10339400
   3409163983   -1.0      0.01      0.02      1.00      2.00   5   0   2         702
0.10292400
   1662077177   -1.0      1.00      1.10     -1.00     -1.00  16   0   0         724
0.11880500
   2811479899   -1.0      1.00      1.10     -1.00     -1.00  15   0   0         716
0.11003600
   2016335473   -1.0      1.00      1.10     -1.00     -1.00  15   0   0         716
0.10752000
```

Several values of the annealing parameter can be provided, separated by commas: for instance '-T10,20,50'.

1.2.3 Turning off recurrent mutations

Specifying '-S-1 -M-1' turns off recurrent mutations, so in this case an upper bound on Rmin is computed under the infinite sites assumption:

[9]:
```
!./kwarg -S-1 -M-1 -T-1 -Q5 < kreitman_snp.txt
```

```
         Seed   Temp   SE_cost   RM_cost   R_cost   RR_cost   SE   RM   R   N_states
Time
   2623264542   -1.0     -1.00     -1.00      1.00      2.00   0   0   7         146
0.01144100
   2147798063   -1.0     -1.00     -1.00      1.00      2.00   0   0   7         167
0.01107900
   1986743730   -1.0     -1.00     -1.00      1.00      2.00   0   0   7         146
0.01033200
   1996696503   -1.0     -1.00     -1.00      1.00      2.00   0   0   7         146
0.01076200
```

```
   2947321370   -1.0    -1.00    -1.00     1.00     2.00   0   0   7         167
0.01370200
```

1.2.4 Turning off recombination

Specifying '-R-1 -C-1' turns off the possibility of recombinations, so an upper bound on the minimum parsimony score is computed:

[10]:
```
!./kwarg -S1.0 -M1.1 -R-1 -C-1 -Q5 < kreitman_snp.txt
```

```
         Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   2623264542   30.0     1.00     1.10    -1.00    -1.00  18   8   0       2018
0.43616500
   1996696503   30.0     1.00     1.10    -1.00    -1.00  NA  NA  NA       2672
0.68387300
   2947321371   30.0     1.00     1.10    -1.00    -1.00  24   2   0       1875
1.12214200
   2838510949   30.0     1.00     1.10    -1.00    -1.00  20   1   0       1683
0.34656700
   1792471208   30.0     1.00     1.10    -1.00    -1.00  NA  NA  NA       1691
0.43594300
```

1.2.5 Specifying the ancestral (root) sequence

It is possible to specify a particular sequence as ancestral to the sample (corresponding to the root of the ARG), using the '-k' option.

If the input data is in binary format, then the all-zero sequence will be assumed to be ancestral (whether or not this is included in the data).

If the input data is in nucleotide or amino acid representation, then the first sequence in the data will be taken as ancestral.

The effect on the resulting history is illustated further below.

1.3 Outputs

We can re-run a particular instance via specifying the random seed, and output the history as text. Here we are choosing seed Z3391369848, SE_cost=0.01 and RM_cost=0.02. We expect to see 5 sequencing errors, 0 recurrent mutations and 2 recombinations in the output.

The "-b" option will save the history in file "example_history.txt".

The "-d" option will output a picture of the corresponding ARG in DOT format and save it in the file "example_arg.dot". The "-e" options tells KwARG to label the edges with mutations.

If an output option is specified, an output file name must also be given.

[11]:
```
!./kwarg -S0.01 -M0.02 -Z3391369848 -bexample_history.txt -dexample_arg.dot -e␣
↪< kreitman_snp.txt
```

```
         Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   3391369848   30.0     0.01     0.02     1.00     2.00   5   0   2        702
0.12403500
```

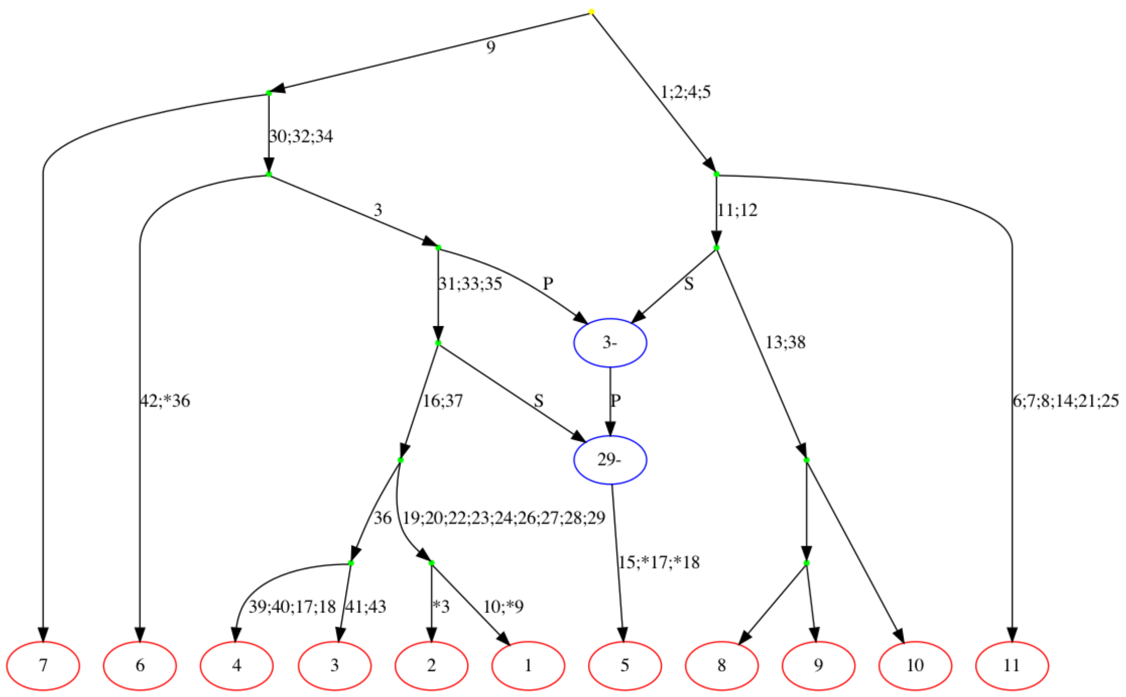
[12]:
```
!cat example_history.txt
```

```
Mutation of site 6 in sequence 11
Mutation of site 7 in sequence 11
Mutation of site 8 in sequence 11
Mutation of site 10 in sequence 1
Mutation of site 14 in sequence 11
Mutation of site 15 in sequence 5
Mutation of site 21 in sequence 11
Mutation of site 25 in sequence 11
Mutation of site 39 in sequence 4
Mutation of site 40 in sequence 4
Mutation of site 41 in sequence 3
Mutation of site 42 in sequence 6
Mutation of site 43 in sequence 3
Coalescing sequences 8 and 9
Coalescing sequences 8 and 10
Mutation of site 13 in sequence 8
Mutation of site 38 in sequence 8
---->Sequencing error at site 36 in sequence 6
---->Stretch of sequencing errors spanning 2 sites:
---->Sequencing error at site 17 in sequence 5
---->Sequencing error at site 18 in sequence 5
Mutation of site 17 in sequence 4
Mutation of site 18 in sequence 4
Coalescing sequences 3 and 4
Mutation of site 36 in sequence 3
---->Sequencing error at site 9 in sequence 1
---->Sequencing error at site 3 in sequence 2
Coalescing sequences 1 and 2
Mutation of site 19 in sequence 1
Mutation of site 20 in sequence 1
Mutation of site 22 in sequence 1
Mutation of site 23 in sequence 1
Mutation of site 24 in sequence 1
Mutation of site 26 in sequence 1
Mutation of site 27 in sequence 1
Mutation of site 28 in sequence 1
Mutation of site 29 in sequence 1
Coalescing sequences 1 and 3
```

```
Mutation of site 16 in sequence 1
Mutation of site 37 in sequence 1
---->Recombination in sequence 5 after site 29; suffix is new sequence 12
Coalescing sequences 1 and 12
Mutation of site 31 in sequence 1
Mutation of site 33 in sequence 1
Mutation of site 35 in sequence 1
---->Recombination in sequence 5 after site 3; prefix is new sequence 13
Coalescing sequences 1 and 13
Mutation of site 3 in sequence 1
Coalescing sequences 8 and 5
Coalescing sequences 1 and 6
Mutation of site 11 in sequence 8
Mutation of site 12 in sequence 8
Mutation of site 30 in sequence 1
Mutation of site 32 in sequence 1
Mutation of site 34 in sequence 1
Coalescing sequences 1 and 7
Coalescing sequences 8 and 11
Mutation of site 1 in sequence 8
Mutation of site 2 in sequence 8
Mutation of site 4 in sequence 8
Mutation of site 5 in sequence 8
Mutation of site 9 in sequence 1
Coalescing sequences 1 and 8
Total: 5 sequencing errors, 0 recurrent mutations, 2 recombinations.
```

If GraphViz is installed, it can be used to convert the DOT file to a png.

[13]:
```
!dot -Tpng -o example_arg.png example_arg.dot
```



Recurrent mutations are labelled with '*'. Recombination nodes are coloured blue, with the number inside the node denoting the recombination breakpoint. The parts of the genome to the left and right of the breakpoint are inherited from two different parent nodes. The edges leading to these nodes are labelled 'P' and 'S', which stands for 'prefix' and 'suffix', respectively.

See the help for a full list of possible output formats. For instance, we can also print out the local trees for each interval in Newick format as follows:

[14]:
```
!./kwarg -S0.01 -M0.02 -Z3257491408 -texample_local_trees.txt -I < kreitman_snp.
↪txt
```

```
         Seed    Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R   N_states
Time
   3257491408    30.0     0.01     0.02     1.00     2.00   5   0   2        702
0.16322600
```

The '-I' option means a tree is given for each interval between recombination breakpoints. Not specifying this option will mean that one tree is produced for each site.

[15]:
```
!cat example_local_trees.txt
```

```
((((((1,2),(3,4)),5),6),7),(((8,9),10),11))1-3;
(((((1,2),(3,4)),6),7),((5,((8,9),10)),11))4-29;
((((((1,2),(3,4)),5),6),7),(((8,9),10),11))30-43;
```

1.3.1 Specifying the ancestral sequence

Compare the output ARGs when the ancestral sequence is and is not specified:

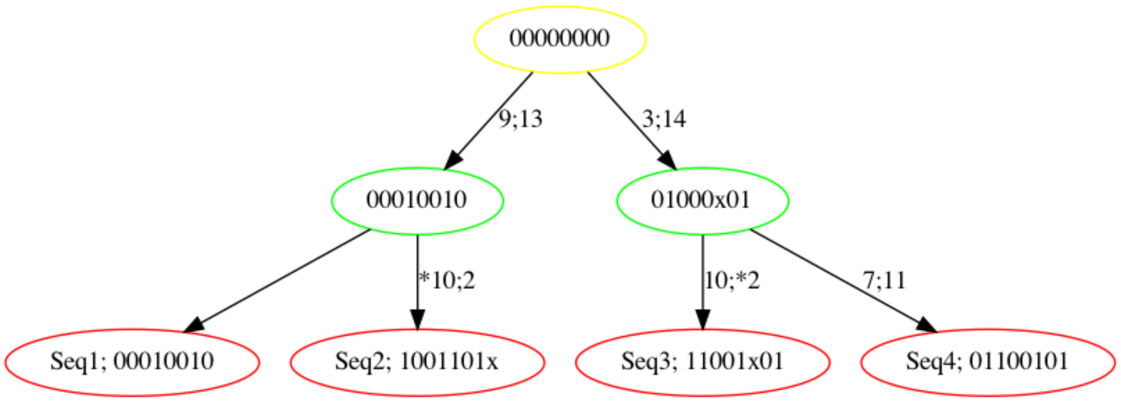
```
[16]: !cat example_data_2.txt
```

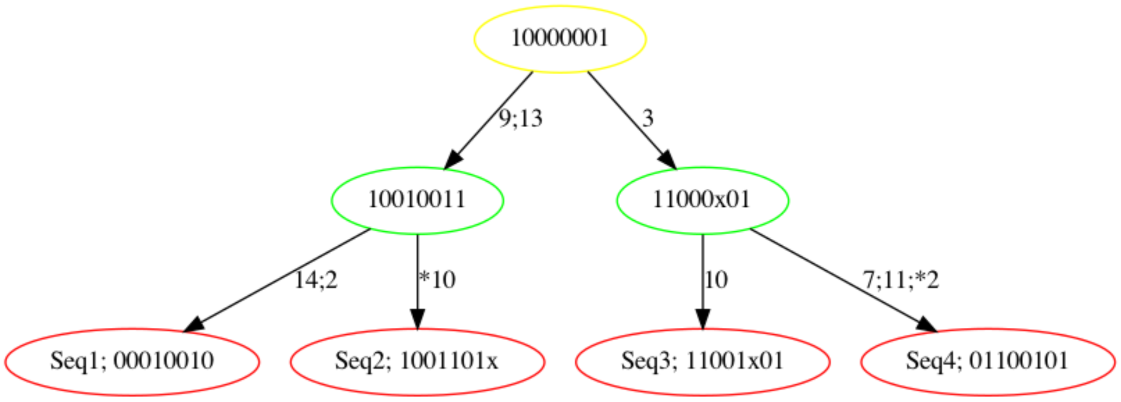
```
#> Seq1
00010010
#> Seq2
1001101-
#> Seq3
11001-01
#> Seq4
01100101
#positions: 2 3 7 9 10 11 13 14
```

```
[17]: !./kwarg -L1 -Z3066074262 -dexample_ancestral.dot -k -e -vboth < example_data_2.
       ↪txt
      !./kwarg -L2 -Z3066074262 -dexample_nonancestral.dot -e -vboth -s <␣
       ↪example_data_2.txt
      !dot -Tpng -o example_ancestral_arg.png example_ancestral.dot
      !dot -Tpng -o example_nonancestral_arg.png example_nonancestral.dot
```

```
       Ref            Seed   Temp  SE_cost  RM_cost   R_cost  RR_cost  SE  RM   R
N_states              Time
         1      3066074262   30.0     0.50     0.90     1.00     2.00   2   0   0
55      0.01355600
         2      3066074262   30.0     0.50     0.90     1.00     2.00   2   0   0
36      0.00166900
```

Here '-vboth' means that the leaves are marked with their corresponding reference (if specified), and all nodes are labelled with the corresponding sequence.





1.4 Simplify

This is a small program which reduces the input dataset using the 'Clean' algorithm, removing all mutations and coalescing all possible sequences until no further reduction is possible. The input data types are the same as for KwARG.

[18]:
```
!./simplify < kreitman_snp.txt
```

```
Input dataset: 11 sequences, 43 sites
Reduced dataset: 9 sequences, 16 sites
0100010101010101
0001010101010101
0101010001010111
0101011001010111
0110101001010100
0001000000000010
0001000010101000
1010100010101000
1010000010101000
Sequences:
X 1 2 3 4 5 6 X 10
Sites:
X 2 X 8 X 15 X X 29 30 31 32 33 34 35 36
```

The output sequence labels show either the number of the sequence in the input dataset, or 'X' if the sequence in the output has been coalesced with another. The site labels show either the number of the site as in the input dataset, or 'X' if the column corresponds to multiple sites collapsed into one.

1.5 Flip

This is a small program which flips the nucleotides at the given sequence and site (from 0 to 1) and outputs the resulting amended dataset.

[19]:
```
!cat kreitman_snp.txt
```

```
0010000000100000010011011101111010101010000000
0000000010000000100110111011110101010100000000
0010000001000000100000000000000001010111000101
0010000001000000111000000000000001010111011000
0011100000110010110000000000000010101000000000
0000000010000000000000000000000000000010000010
0000000010000000000000000000000010101000000000
1101100000111000000000000000000010101000100000
1101100000111000000000000000000010101000100000
1101100000111000000000000000000010101000100000
1101111100000100000001000100000101010000000000
```

[20]:
```
!./flip -q6,8 -s37,1 < kreitman_snp.txt
```

```
0010000000100000010011011101111010101010000000
0000000010000000100110111011110101010100000000
0010000001000000100000000000000001010111000101
0010000001000000111000000000000001010111011000
0011100000110010110000000000000010101000000000
0000000010000000000000000000000000000011000010
0000000010000000000000000000000010101000000000
0101100000111000000000000000000010101000100000
1101100000111000000000000000000010101000100000
1101100000111000000000000000000010101000100000
1101111100000100000001000100000101010000000000
```

Using the option ‘-bfilename.txt’ will save the resulting dataset to filename.txt.