Supplementary material

1 RESTRICTION RULES FOR ARGS

In the following a formal description of the restriction rules (introduced in Section 2.3 of the article) imposed to an ARG by a classification is given (in brackets we indicate which symbols are used in Figure 2 of the article and in Figure 12, resp., for the sequences):

- A pair of sequences can only coalesce if
  - both sequences belong to the same subtype ($\times$, $\times$) or CRF ($\times$, $\times$), or
  - the sequences ($*$, $*$)
    - are the only sequences of their subtype left or
    - belong to more than one subtype and are the parent of a coalescent event.

  Here, a sequence generated by a coalescent event is defined to belong to the same subtype(s) or CRF, resp., as its children, i.e.,
  - the parent of two subtype A sequences belongs to subtype A,
  - the parent of two CRF1 sequences belongs to CRF1,
  - the parent of one subtype A and one subtype B sequence belongs to subtype A and subtype B.

  A sequence generated by a recombination event belongs to the subtype(s) its segments belong to.
- The sequences of a CRF must all coalesce before they undergo a recombination event. Only the last sequence left ($\times$, $\times$) is allowed to recombine. (Multiple) breakpoints have to be chosen such that the parental subtypes get separated and recombination events have to take place until all parental subtypes are separated.

2 MCMC DETAILS AND MOVE TYPES

The Markov chain Monte Carlo algorithm for ARGs fulfilling the restrictions imposed by a given classification is described, including proposal mechanism used.

Let G and H be ARGs. Then the change from G to H is accepted if

$$r := \frac{P(D|H)P(H|\Theta)Q(H,G)}{P(D|G)P(G|\Theta)Q(G,H)} > u$$

where u is sampled from a uniform distribution on $[0,1]$. $Q(G,H)$ denotes the proposal probability specifying the probability to generate H in the next step given G is the current ARG.

Note that, if

$$Q(H,G) = CP(G|\Theta), \quad Q(G,H) = CP(H|\Theta) \tag{1}$$

with $C > 0$,

$$r = \frac{P(D|H)}{P(D|G)} \tag{2}$$

Hence, if a proposal ARG is sampled with respect to a conditional coalescent distribution, r only depends on the probability of the data with respect to the genealogy.

In total, we apply five different types of proposal mechanisms (moves), chosen such that the whole space of legal (i.e. fulfilling the classification-given restrictions) ARGs can be entirely be traversed and the MCMC algorithm converges fast into areas of ARGs with high likelihood. The last three of them fulfill (2). Except the first move (which is a global rescaling operation), all perform local rearrangements, i.e., among all subgraphs fulfilling specific topological and typological properties one subgraph is chosen randomly and is rearranged.

In the description of the moves, we will use the following notation:

- Given an ARG G, its nodes are denoted by $N = N_G$. Let $\mathrm{Tip}(G)$ be the tip nodes of G and $\mathrm{Int}(G) = N_G \setminus \mathrm{Tip}(G)$ the internal nodes of G. Denoting the subtypes of the classification by $S = \{S_1, \dots, S_{m_p}\}$ and its CRFs by $C = \{C_1, \dots, C_{m_r}\}$, we define $\mathrm{Type} : N \to S \cup C \cup \{\mathrm{Imp}\}$

$$n \to \begin{cases} S_i, & \text{if } n \text{ belongs only to subtype } S_i \\ C_i, & \text{if } n \text{ belongs to CRF } C_i \\ \mathrm{Imp}, & \text{else} \end{cases}$$

  where Imp is a symbol standing for "impure".
- The child(ren) and parent(s), resp., of a node $n \in N$ is denoted by $\mathcal{C}(n) \in \wp(N)$ and $\mathcal{P}(n) \in \wp(N)$, resp., with $\wp(N)$ denoting the power set of N. If n has only one child or parent, resp., $\mathcal{C}(n)$ or $\mathcal{P}(n)$, resp., are also interpreted as elements of N. If n has two children or parents, resp., they are denoted by $\mathcal{C}_1(n)$ and $\mathcal{C}_2(n)$ or $\mathcal{P}_1(n)$ and $\mathcal{P}_2(n)$, resp. In case n has only one child, it has to have a spouse, which is denoted by $\mathcal{S}(n)$. Furthermore, we define

$$\mathcal{P}^d(n) := \begin{cases} \mathcal{P}(n), & \text{if } d = 1 \text{ and } \#\mathcal{P}(n) = 1 \\ \mathcal{P}(\mathcal{P}^{d-1}(n)), & \text{if } d > 1 \text{ and } \#\mathcal{P}(\mathcal{P}^{d-1}(n)) = 1 \\ \text{undefined}, & \text{else} \end{cases}$$

  for $d \in \mathbb{N}$. Moreover, the age (i.e. time of generation) of $n \in N$ is denoted by $T(n)$.
- The container of $n \in N$ is defined by

$$B(n) := \begin{cases} \{n\}, & \text{if } n \in \mathrm{Tip}(G) \text{ or } \#\mathcal{C}(n) = 2 \\ \{n, \mathcal{S}(n)\}, & \text{if } \mathcal{S}(n) \text{ is defined} \end{cases}$$

  ("B" stands for "Box"). We denote the set of all containers of G by $B = B_G$, i.e.,

$$B := \{B(n) : \ n \in N\}.$$

In detail, the five moves are:

1. Scaling move: For all non-tip nodes n, $T(n)$ is multiplied by $c \sim U([1-\delta, \frac{1}{1-\delta}])$ with $0 < \delta \ll 1$.
2. Branch-Swapping move: This move is similar to the Wilson-Balding move described in Drummond *et al.* (2002). Among the set

$$\{n \in N : \ \#\mathcal{C}(n) = 2\}$$

   one (target) node n_t is chosen randomly. Then, one (destination) node n_d is chosen randomly among the set

$$\{n \in N : \ \#\mathcal{P}(n) = 1, \ \mathrm{Type}(n) = \mathrm{Type}(n_t), \\ T(n) < T(n_t) < T(\mathcal{P}(n))\},$$

   Finally, n_t is moved with one of its children such that n_t becomes the parent of n_d and the child of the former parent of

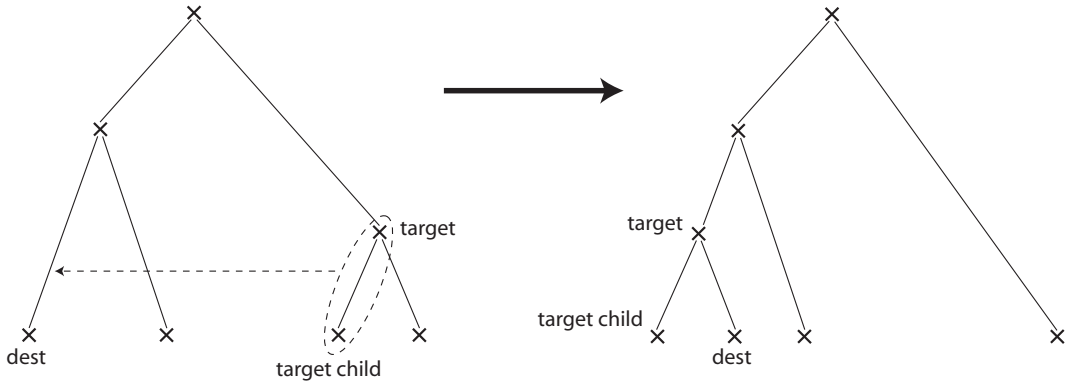

Figure 1. The branch-swapping move

n_d, and the other child of n_t becomes the child of the former parent of n_t (see Figure 1).

3. Node age move: Let $b_1, \ldots, b_n$ be the non-tip containers ordered by age, i.e., $T(b_i) \leq T(b_j)$ for $i < j$. For this move, a non-root container $b_i \in B$ is chosen randomly among $b_1, \ldots, b_{n-1}$. Then one of the three following moves are carried out with equal probability: The age of the container is drawn from a conditional coalescent distribution (with given population parameters) conditioned on that
   a. the order of the containers does not change
   b. $T(b_1) \leq \ldots \leq T(b_{i+1}) \leq T(b_i) \leq T(b_{i+2}) \leq \ldots \leq T(b_n)$ (this move is forbidden if b_{i+1} contains the parent(s) of b_i)
   c. $T(b_1) \leq \ldots \leq T(b_{i-2}) \leq T(b_i) \leq T(b_{i-1}) \leq \ldots \leq T(b_n)$ (this move is forbidden if $i = 1$ or b_{i+1} contains a child of b_i)

   The move under b) is also called an "up move", the one under c) a "down move".
4. Coalescent move: A (target) node n_t is chosen at random from

$$\{n \in N : \#\mathcal{C}(n) = 2, \#\mathcal{P}(n) = 1\}.$$

   The so-called neighborhood of rearrangement consists of the target node, its children, parent, and parent's other child. This move makes changes of two kinds: it may reassign the three children among target and parent, and it modifies the branch lengths within the neighborhood. The new branch lengths must remain within the constraints imposed by the times of the three children and of the parent's ancestor (if existing); these times define the boundaries of the neighborhood. Conceptually, the portion of the genealogy involving these nodes is erased and must now be redrawn. This move is based on the rearrangement move introduced by Kuhner *et al.* (1995) (Large parts of this description were taken from Kuhner *et al.* (1995)). Technical details about this move for ARGs without recombination events are described in Kuhner *et al.* (1995), our extension to ARGs with recombination events is not shown due to the length of our deduction.
5. Recombination move: This is the most complicated move and is introduced in order to reorder nodes involved in recombination events. Among

$$\{n \in N : \#\mathcal{P}(n) = 2\}$$

   a (target) node n_t is chosen randomly. Let the sets $\{R_i\}_{i \in \mathbb{N}_0}$ and R be defined by

$$\begin{aligned} R_0 &:= \{n_t\}, \\ R_i &:= \{n \in N : \exists n_0 \in R_{i-1} : \#\mathcal{P}(n_0) = 2, n \in \mathcal{P}(n_0)\}, \\ &\quad i \in \mathbb{N}, \\ R &:= \bigcup_{i \in \mathbb{N}} R_i \end{aligned}$$

   and

$$H := \{n \in N : \exists n_0 \in R : \#\mathcal{P}(n_0) = 1, \mathcal{P}(n_0) = n\}$$

   (cf. Figure 2a). All nodes belonging to R and H (except n_t) are removed from the ARG and

$$\begin{aligned} &\forall h \in H \; \forall n_0 \in \mathcal{C}(h), \; n_0 \notin R : \; \mathcal{P}(n_0) \leftarrow \mathcal{P}^d(n_0), \\ &d = \min\{i \geq 2 : P^i(n_0) \notin H\} \end{aligned}$$

   (cf. Figure 2b). Realize that $d = 2$ if no unknown subtype occurs and at least two subtypes have to be present in order to make this move work. Now, denote by

$$(\{S_1, \ldots, S_n\} \rightarrow \{S_{i_1}, \ldots, S_{i_{n_1}}\}, \{S_{j_1}, \ldots, S_{j_{n_2}}\})$$

   a recombination event which separates the subtypes $\{S_1, \ldots, S_n\}$ into the subtypes $\{S_{i_1}, \ldots, S_{i_{n_1}}\}$ and $\{S_{j_1}, \ldots, S_{j_{n_2}}\}$ (called R-event) and by

$$s \rightarrow T$$

   the event of a node belonging to subtype s being connected to the rest of the ARG by a coalescent event (called C-event). Let M be the set of finite sequences of R- and C-events such that, if carried out chronologically on n_t, lead to a legal ARG. As next step of the move, $m \in M$ is chosen randomly and n_t is reconnected according to m, where the age of the newly

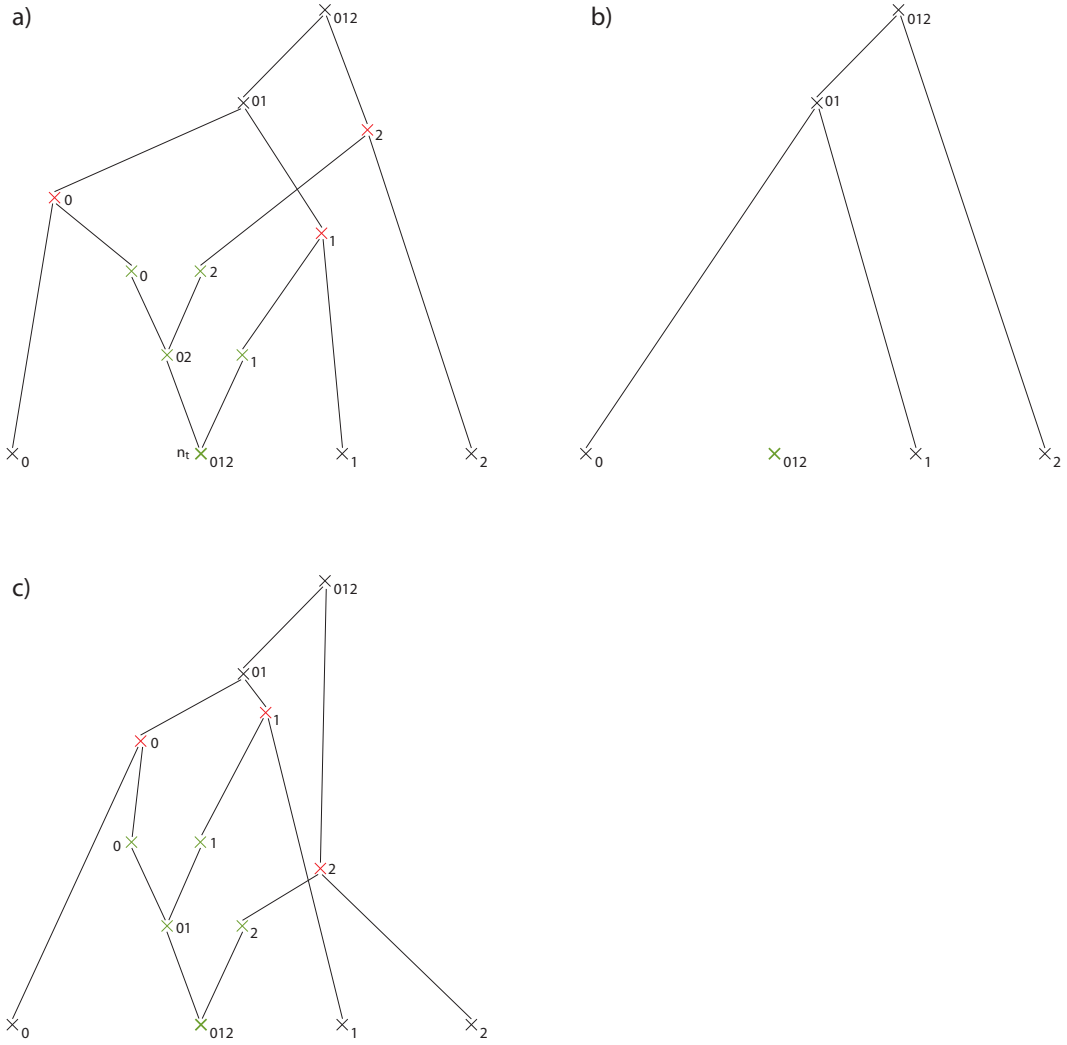

Figure 2. The "recombination move". a) Nodes belonging to R are colored in green, the ones belonging to H in red. Numbers denote to which subtypes the nodes belong to. b) All nodes belonging to R and H except the target node have been removed and the non-R children of H have been connected to the next valid ancestor. c) The removed part of the ARG has been regenerated.

generated nodes is chosen randomly from a simply sampleable distribution (not the conditional coalescent distribution). E. g.,

$$\{0,1,2\} \to \{0,1\},\{2\}, \quad 2 \to T, \quad \{0,1\} \to \{0\},\{1\},$$
$$0 \to T, \quad 1 \to T$$

would lead to an ARG like shown in Figure 2c. Then a fixed number of extended "node age moves" is applied to the newly generated nodes, where "extended" means that, additionally to the move described under 3., a movement of nodes of H beyond their parent and children is allowed under suitable circumstances (cf. Figure 3). In more detail, we relax the conditions (b) and (c) under 3. by allowing "up moves" also if b_{i+1} is non-root and the parent of b_i and "down moves" if $\#\mathcal{C}(b_{i-1}) = 2$. Such moves are carried such that the ARGs yielded by a "recombination move" are samples with respect to a conditional coalescent distribution.

After having carried out these moves, we have to reconnect the nodes accordingly to the ARG like follows:

- "down move": Sample $j \sim U(1,2)$, and set $\mathcal{P}(n_d) \leftarrow \mathcal{P}(n_t)$, $\mathcal{P}(n_t) \leftarrow n_d$, $\mathcal{P}(\mathcal{C}_j(n_d)) \leftarrow n_t$
- "up move": Let $n_c := \{n \in \mathcal{C}(n_t) : \#\mathcal{C}(n) \neq 1\}$ and set $\mathcal{P}(n_t) \leftarrow \mathcal{P}(n_d)$, $\mathcal{P}(n_d) \leftarrow n_t$, $\mathcal{P}(n_c) \leftarrow n_d$.

The ARG obtained by this procedure is the result of the "recombination move". Notice that this move would only not violate (1) if $\mathcal{P}(T_{m_1}) = \mathcal{P}(T_{m_2})$ for $m_1, m_2 \in M$, where T_m is the set of ARGs which could be generated according to m. But since all m involve the same number of coalescent and recombination events and we do not sample ARGs, but seek a maximum, this seems to be an acceptable compromise between exactness on the one hand and complexity and speed on the other hand. In case we intend to sample ARGs in the future,

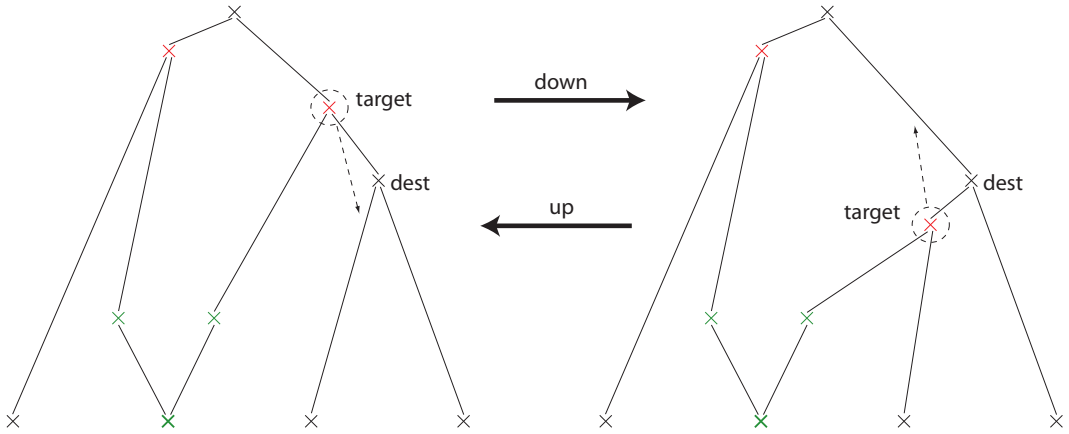

Figure 3. The modified "up move" and "down move". Left to right shows a "down move", right to left an "up move".

we will have to allow for moves between T_{m_1} and T_{m_2} for different m_1 and m_2 instead of only carrying out extended "node age moves".

3 ALGORITHM

3.1 jpHMM

Applying jpHMM only to one sequence of each CRF (and assigning the calculated segmentation to all sequences belonging to this CRF) could seem questionable if jpHMM yielded strongly dissimilar results for different sequences s_a and s_b of the same CRF. But such diverging results of jpHMM would also indicate that the whole classification is rather poor since one should obviously not assign s_a and s_b to the same CRF. Hence, the behavior of ARGUS to reconstruct a genealogy of low likelihood in this case (due to assigning an inappropriate segmentation to either s_a or s_b) will correct for the restricted application of jpHMM.

3.2 Coalescent model

In coalescent theory, time is traversed backwards starting at the tips, generating genealogical events (i.e. coalescent events and recombination events) according to their rate, until only one node is left (called the root node). The rate of coalescence is $k(k-1)/\Theta$, where k is the number of active lineages, and the rate of recombination is rs, where s is the length of the genome region in which a valid recombination event might occur, summed over all lineages (valid means not to be discarded because it does not contribute to the sample, cf. Section 3.2 of the article). The prior probability of the ARG G is

$$P(G|\Theta, r) = \left(\frac{2}{\Theta}\right)^{N_C} r^{N_R} \exp\left[\sum_i -\left(\frac{k_i(k_i-1)}{\Theta} + rs_i\right) t_i\right]$$

where N_C is the number of coalescent events and N_R the number of recombination events in G, k_i the number of active lineages between the ith and $(i+1)$th genealogical event, s_i the sum of valid sites in that interval, t_i the length of the time interval between the ith and $(i+1)$th genealogical event (see Kuhner *et al.*, 2000).

Since we assume that mutations at different sites are independent, $P(D|G)$ can be easily calculated sitewise (Felsenstein, 1981). For the mutation process, a General Time Reversible (GTR) model (Lanave *et al.*, 1984) with mutation rate varying among the sites is used. The variation is modeled by a gamma distribution (Yang, 1994) and the parameters of the GTR model were estimated with Findmodel (www.hiv.lanl.gov/content/sequence/findmodel/ findmodel.html).

3.3 Scoring

Instead of Equation (1) one could also interpret the likelihood of the ARG

$$\max_{i=1,\ldots,n} P(D|G_i)P(G_i|\Theta, r, R)$$

as a score for the classification. Nevertheless, using $P(G|\Theta, r, R)$ to score a classification makes only sense if the tip sequence data is sampled randomly. Obviously, this is absolutely not the case for our applications. Hence, we neglect $P(G|\Theta, r, R)$ and only consider $P(D|G)$ (Users who find a way to estimate r for their data can incorporate this knowledge manually). Anyways, normally the difference between $P(D|G)P(G|\Theta, r, R)$ for different classifications is strongly dominated by $P(D|G)$.

3.4 Influence of parameters

In this section we shortly discuss the influence of recombination and mutation rate parameter r and Θ on the accuracy of the classification procedure.

r: Since the number and type of recombination events is determined by the classification, the influence of r is very small (recall that $P(D|G)P(G|\Theta, r, R)$ is strongly dominated by $P(D|G)$). Nevertheless, one has to keep in mind that a very small recombination rate would probably imply that only classifications composed only of pure subtypes would be reasonable and that ARGUS is not designed for this use case (see Section 4 in the article).

Θ: Simulation and scoring runs for C1.1 and C2.1 indicate that ARGUS can be expected to run reliably for Θ down to 0.05 (with the other parameters chosen as in the article). Since HIV is one of the most strongly organisms, using a much higher Θ than in our setting does not seem practical relevant.

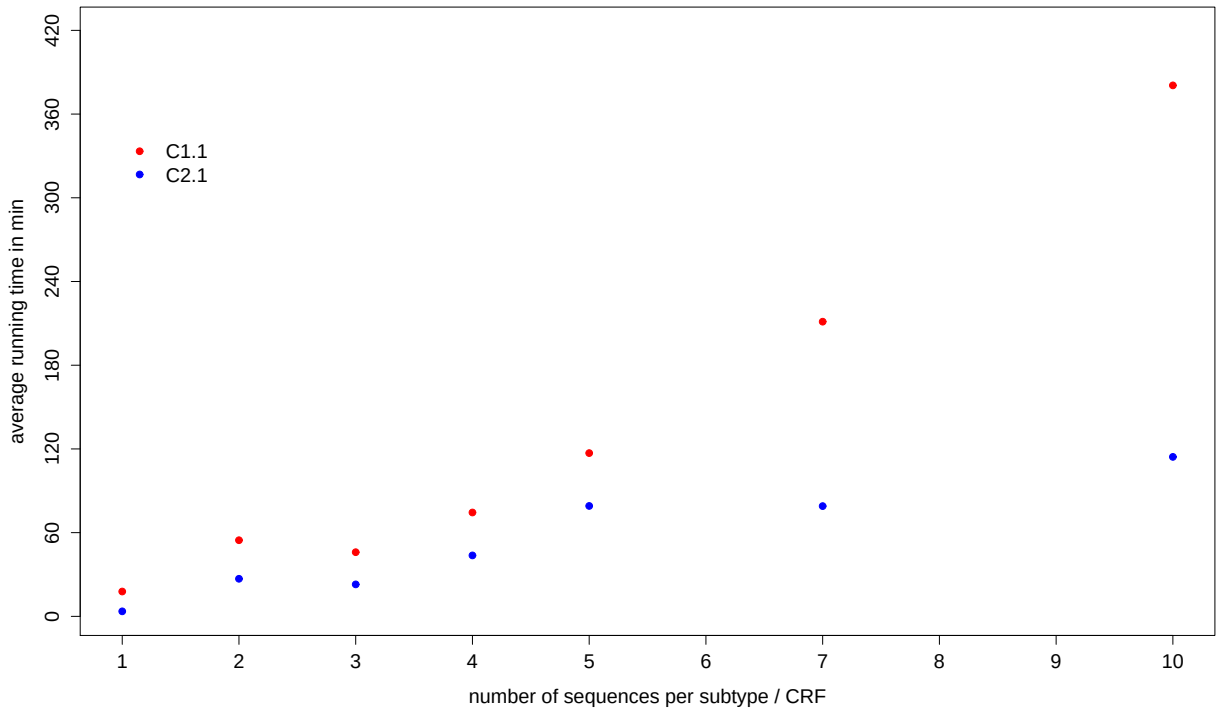

Figure 4. Modified versions of classifications C1.1 and C2.1 (differing in the number of sequences per subtype/CRF) were used for simulation (5 ARGs per classification, 5 mutation simulations per ARG) and the resulting sequence data set were scored for the same classification (10 scoring runs per sequence data set). The average running time for scoring is given.

3.5 Running time

The running time for a iteration step of the MCMC algorithm is dominated by the calculation of $P(D|G)$. Moreover, the computing time of $P(D|G)$ scales linear in

- the length of the input sequence data,
- the number of input sequences.

Hence, the running time per iteration step is approximately linear in the length of the input sequence data and the number of input sequences.

The total running time of ARGUS is roughly linear in the number of MCMC steps carried out, with MCMC steps reordering recombination events being considerably more expensive than other types of steps (this partly explains the difference in running time between C1.1 and C2.1 in the presentation below). The total number of MCMC steps is very difficult to estimate since it is influenced by many (partly random) factors.

In order to give an idea of the dependence of the running time against the size of the input classification, we run ARGUS with input classifications of various sizes. More precisely, we use the classifications C1.1 and C2.1, but with 1 to 10 sequences per subtype resp. CRF instead of 3 (i.e. same number for all subtypes and CRFs in a classification). We use each of these modified classifications for simulation and then score the resulting sequence data set for the same classification. The resulting average running times for scoring are plotted in Figure 3.5.

3.6 Extension to unknown subtypes

In the genome of several CRFs, segments are commonly classified to belong to an unknown subtype. In order to address classification problems involving unknown subtypes, we extend the restriction rules: We additionally interprete a sequence generated by a recombination event and belonging only to one, unknown subtype as the only sequence of its subtype left (cf. rules for coalescent events in Section 2.3 of the article. This case is illustrated in Figure 5.

jpHMM is not able to detect segments belonging to an unknown subtype and, to our knowledge, up to now no tool is available for automatically segmenting sequences into known and unknown subtypes. Hence, segments belonging to an unknown subtype have to be added manually in the classification after the application of jpHMM.

4 SEPARATING AND NOISE DISTANCE

Let

$$N_i^p = \{n \in N : \text{ Type}(n) = S_i\}$$

for $i \in \{1, \ldots, m_p\}$ and

$$n_i^f = \text{argmax}_{n \in N_i^p} T(n).$$

Moreover, for $n_1, n_2 \in N$, $n_1 \neq n_2$, let $n_{mrca}(n_1, n_2)$ be the most recent common ancestor node of n_1 and n_2. Then the separating distance d_{sep} is defined by

$$d_{sep} = \sum_{i=1}^{m_p} \sum_{j \neq i} 2 \cdot T(n_{mrca}(n_i^f, n_j^f)) - T(n_i^f) - T(n_j^f)$$

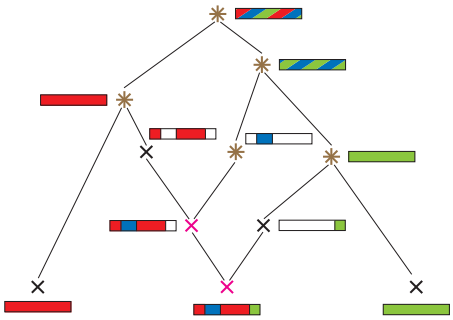

Figure 5. A legal ARG corresponding to a classification having an unknown subtype (blue). For details, see Section 3.3 of the article and 3.6 and Figure 2 of the article.

and the noise distance d_{noise} by

$$d_{noise} = \sum_{i=1}^{m_p} T(n_i^f).$$

5 SIMULATION STUDIES

5.1 T1 - Without recombination

As an initial test and to verify that the method can correctly perform the easier task of constructing a phylogenetic tree (without recombinations), 40 representative HIV-1 Gr. M sequences (7 from subtype A, 7 B, 11 C, 3 D, 3 F, 3 G, 2 H, 2 K, 2 J) are chosen (using FigTree, see http://tree.bio.ed.ac.uk/software/figtree/). Then ARGUS is applied to score the trivial classification (i.e., all sequences belong to one 'subtype'). The most likely ARG achieved by the MCMC algorithm is compared to the phylogenetic tree in Figure 7 of Schultz *et al.* (2006). As desired, in our tree all sequences belonging to the same subsubtype or subtype, resp., first coalesce with the other sequences of the subsubtype or subtype, resp., before coalescing with sequences from another subsubtype or subtype, resp. The remaining sequences of the subtype cluster like follows:

$$((((A,G)(H,J))((B,D)(F,K))),C)$$

In Schultz *et al.* (2006) the tree has the form

$$((((A,G),J),(C,H)),((B,D)(F,K))).$$

We consider this sufficiently similar given that the branch lengths before the split into subtypes J, C, and H are very short.

5.2 T2 - With recombination

We choose two original classifications and for each original classification a number of alternative classifications for testing (Figures 6, 7). We perform the following steps for each original (true) classification in our test setting:

1. Simulate an ARG according to the original classification
2. Simulate the mutation process on the ARG (from the root downwards), thereby obtain simulated tip sequences
3. Score both the original as well as one or more plausible alternative classifications using the simulated tip sequences

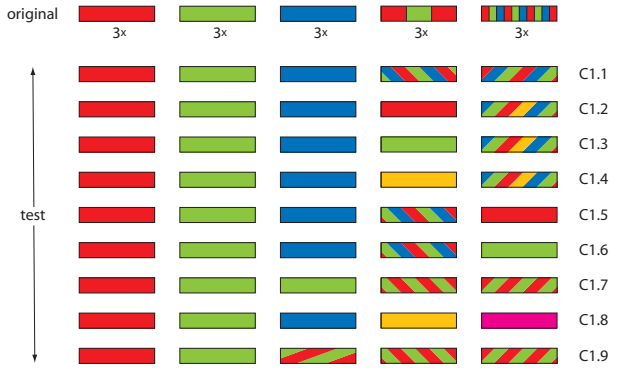

Figure 6. First test of test setting T2. On the top, the original classification is given. Single-color boxes symbolize triples of sequences belonging to a pure subtype (same colors indicate same subtype). The multicolor boxes symbolize sequences belonging to a CRF, showing its segmentation (which has to be provided in order to generate an ARG according to the classification and simulate the mutation process). In the lower part, the tested classifications are given. Single- and multicolor boxes symbolize the same as for the original classifications except that the segmentations of the CRFs are not given (the segmentations used by ARGUS are determined by jpHMM). Instead, the different diagonal patterns symbolize the different CRFs, the colors indicating the subtypes the CRF can be composed of (jpHMM always uses all subtypes available for determination of the segmentation of a CRF).

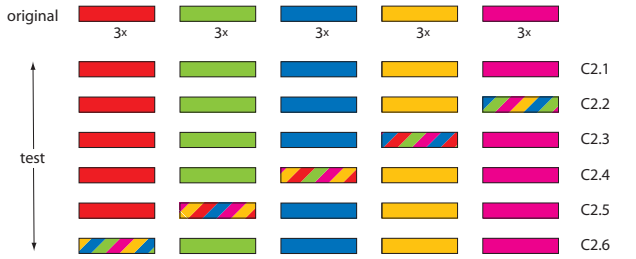

Figure 7. Second test of test setting T2. The same symbolism as in Figure 6 is used.

When the original classification scores higher than the test classifications, this indicates that ARGUS works for the analyzed setting. In the first part, we test 9 classifications of 15 sequences, in the second part 6 classifications of 15 sequences (Figures 6 and 7).

The ARGs are simulated by sampling them with respect to the coalescent distribution, conditioned on the ARG fulfilling the restrictions imposed by the original classification. Notice that sequence data stemming from such ARGs in general does not pose the typical application situation for ARGUS: Normally a classification algorithm is applied to (sub-)species well separated by founder effects (Rambaut *et al.*, 2004). Nevertheless, the chosen testing method allows for highlighting the boundaries of applicability of ARGUS.

In the first test, the original classification has three pure subtypes and two CRFs with three sequences each. The first CRF is equidistantly segmented into three parts belonging to the first two subtypes and the second CRF is equidistantly segmented into ten parts from all three subtypes. The first tested classification (denoted by C1.1)

matches the original classification. The other eight (false) classifications (denoted by C1.2-C1.9) are slight modifications of the original one:

- in C1.2 and C1.3, resp., the fourth triple does not belong to a CRF but to the first and second subtype, resp.,
- in C1.4 the fourth triple does not belong to a CRF but constitutes a fourth subtype,
- in C1.5 and C1.6, resp., the last triple belongs to the first and second subtype, resp.,
- in C1.7 the second and third triple belong to the same subtype,
- in C1.8 all triples belong to distinct subtypes,
- in C1.9 the third triple constitutes a third CRF.

Notice that one could make the task more difficult for ARGUS by also testing classifications only differing from C1.1 by one or two sequences (and not a triple), but we suppose that in real-world applications the input sequences are in general groupable with respect to similarity.

In the second test all triples belong to different subtypes. The original classification again constitutes the first test classification (denoted by C2.1). The other five classifications (C2.2.-C2.6) differ from the original one by one triple being assigned to a CRF.

Especially for the first test, the choice of tested classifications is somehow arbitrary. We plan to overcome this drawback by traversing the space of possible classifications automatically in the future.

Notice that comparing two classifications both having no CRFs is not always reasonable. E.g., the classifications assigning the same subtype to all sequences and a different one to each sequence, resp., always score highest among the CRF-free classifications (assuming the MCMC algorithm finds the global maximum).

For both tests of T2 we simulate 9 ARGs and for each ARG we simulate 5 sets of tip sequences, yielding 90 individual tests. The results are shown in Figures 8 and 9.

ARGUS computed a higher score for the original classification than for the alternative classifications in all cases except the following ones. For the first test, ARGUS fails for 2 out of 9 simulated ARGs to always (i.e. for all simulated tip sequences sets) score the original classification highest: For one tip sequences set of the 5th ARG, C.1.7 scores higher than C.1.1 and for one tip sequences set of the 7th ARG jpHMM fails to find any breakpoint in one of the CRFs of C1.1.

For the second part, ARGUS fails for 1 out of 9 simulated ARGs to always score the original classification highest: C.2.2 scores highest for one tip sequences set of the 5th ARG. jpHMM always finds breakpoints in both CRFs of C2.1.

REFERENCES

Drummond, A. J., Nicholls, G. K., Rodrigo, A. G., and Solomon, W. (2002) Estimating Mutation Parameters, Population History and Genealogy Simultaneously From Temporally Spaced Sequence Data. *Genetics*, **161**, 1307–1320.

Felsenstein, J. (1981) Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.*, **17**, 368–376.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (1995) Estimating Effective Population Size and Mutation Rate From Sequence Data Using Metropolis-Hastings Sampling. *Genetics*, **140**, 1421–1430.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (2000) Maximum Likelihood Estimation of Recombination Rates From Population Data. *Genetics*, **156**, 1393–1401.

Lanave, C., Preparata, G., Sacone, C., and Serio, G. (1984) A new method for calculating evolutionary substitution rates. *J. Mol. Evol.*, **20**, 86–93.

Rambaut, A., Posada, D., Crandall, K. A., and Holmes, E. C. (2004) The causes and consequences of HIV evolution. *Nat. Rev. Genet.*, **5**, 52–61.

Schultz, A.-K., *et al.* (2006) A jumping profile Hidden Markov Model and applications to recombination sites in HIV and HCV genomes. *BMC Bioinformatics*, **7**, 265.

Yang, Z. (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J. Mol. Evol.*, **39**, 306–314.

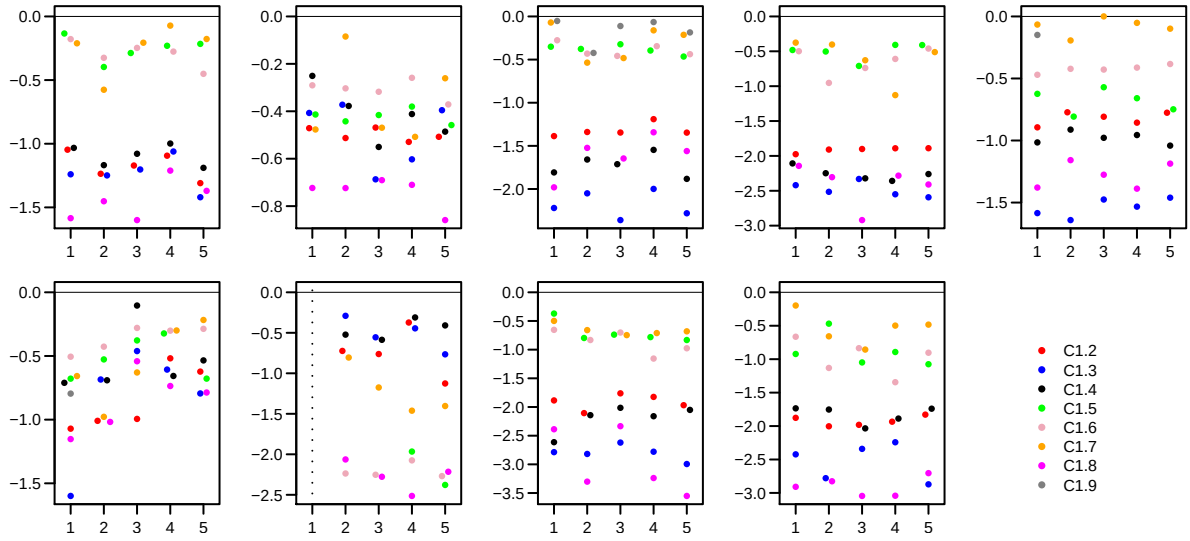

Figure 8. Results for the test setting described in Figure 6. On the vertical axis, $(\log P(D|G_T) - \log P(D|G_O)) \cdot 10^{-3}$ is given, with G_T the most likely ARG for the test (false) classifications C1.2-C.1.9 and G_O the most likely reconstructed ARG for the original classification C1.1. On the horizontal axis, the number of the set of simulated tip sequences is given. All points on a vertical represent tests conducted for the same tip sequences data (For clarity, points with very similar y-values were shifted slightly horizontally). In case a test classification contains one or more CRFs, but jpHMM was not able to detect all (i.e. at least one alleged CRF were diagnosed to belong to a pure subtype), the test results are omitted. In case that jpHMM designated at least one CRF of the original classification C1.1 to belong to a pure subtype, all test results for this tip sequences data set are omitted and a vertical dotted line is drawn instead. Depending on the stability of the results, 10-30 different initial ARGs were used for the MCMC algorithm, but always the same number for tests belonging to the same simulated ARG.

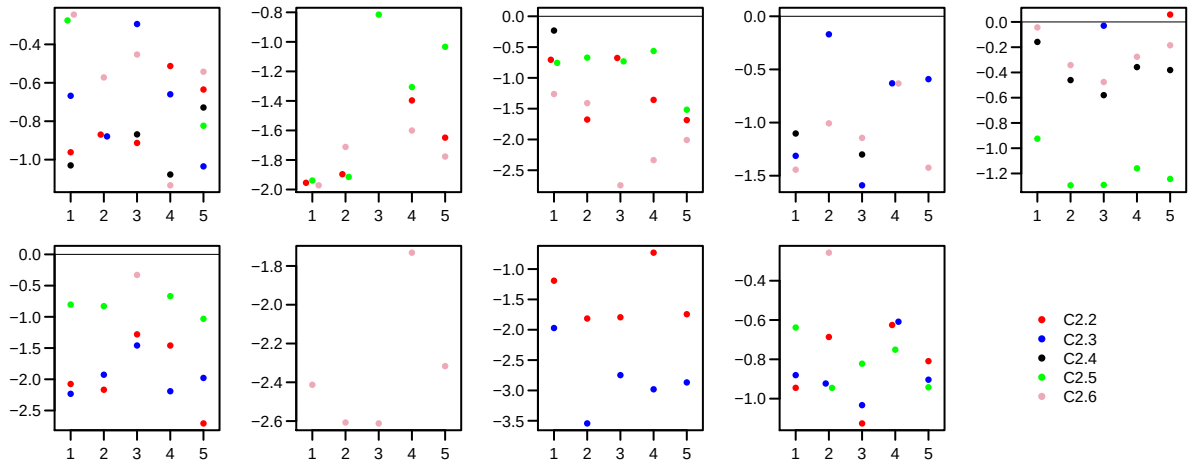

Figure 9. Results for the test setting described in Figure 7. We used 10-100 different initial ARGs for the MCMC algorithm. For details, see Figure 8.

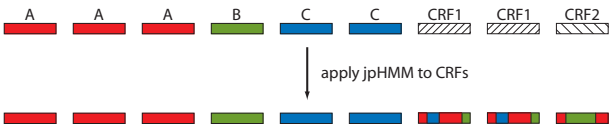

Figure 11. Colored version of Figure 1 in the article. Example of a classification of 9 sequences into 3 subtypes (A, B, C) and 2 CRFs (CRF1, CRF2). At the bottom the recombinants have been segmented and the segments assigned a subtype by jpHMM.

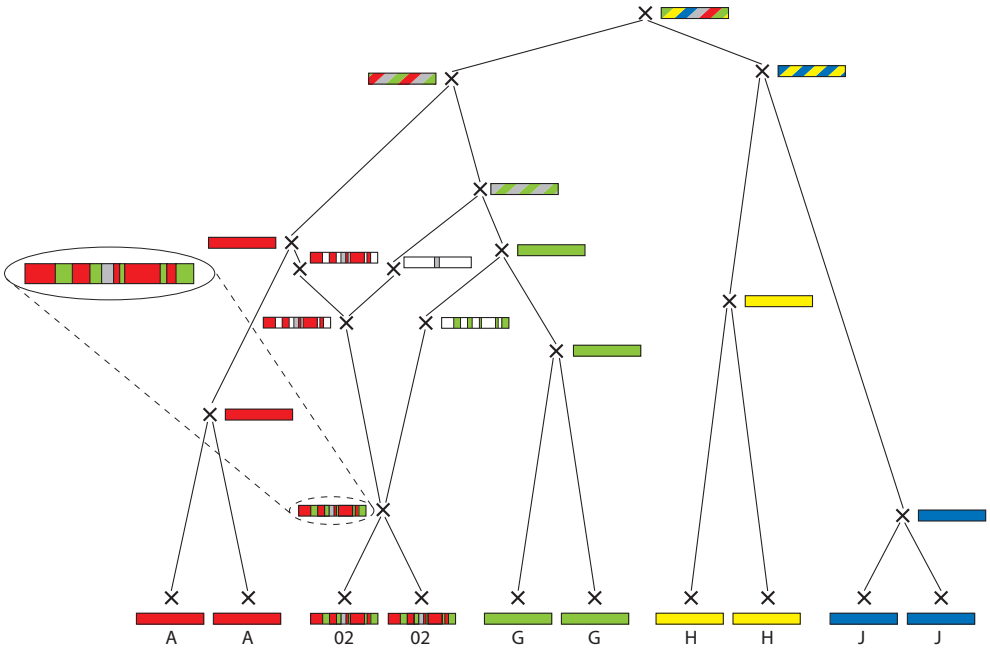

Figure 10. The most likely ARG found by the MCMC algorithm applied to C.02 (see Figure 3 of the article) using real HIV-1 Gr. M sequences. The vertical distance of the internal nodes to the tip nodes is drawn proportionally to their time of generation. The genome of one CRF02 sequence is shown magnified. For details about the symbolism used in the ARG, see Figure 2 of the article.

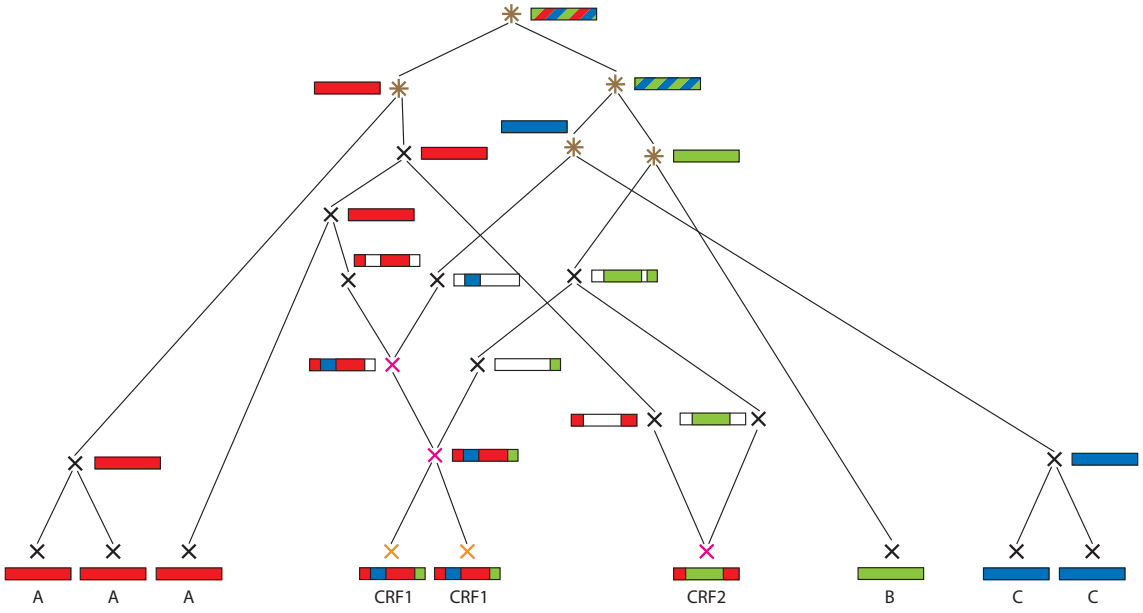

Figure 12. Colored version of Figure 2 in the article. A legal ARG corresponding to the classification given in Figure 11. At the bottom, the nine input (tip) sequences with their classification are shown. The tip sequences are defined to be generated at time zero. Looking from bottom to top (i.e. into the past), two nodes coalescing to one (parental) node, represent the event of these two nodes finding their most recent common ancestor. A node splitting into two parental nodes represents a recombination event. Single-color boxes show the subtype of the node. Horizontally segmented boxes show for a recombinant sequence the parental subtypes of each segment. Diagonally shaded boxes show the different subtypes the node belongs to. White parts in boxes indicate positions not contributing to the tip sequences and, hence, of which we do not keep track. For recombination events, they also illustrate the positions of the recombination breakpoints. For further details, see Section 2.3 of the article.

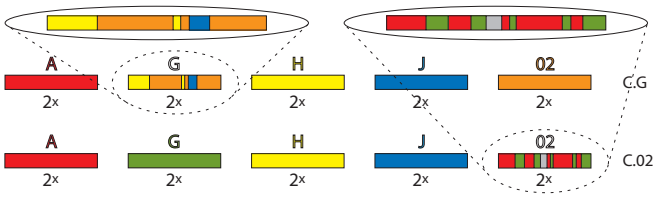

Figure 13. Colored version of Figure 8 in the article. Classifications used in Section 3.2 of the article for deciding whether subtype G or CRF02 (=02) is a pure subtype or a recombinant form, resp. The gray segment in the lower segmentation of CRF02, indicates a part of the genome designated to stem from an unknown subtype. Above the classifications, the segmentation of the alleged CRFs is shown magnified.