Supplementary Material

ksrates: positioning whole-genome duplications relative to speciation events using rate-adjusted mixed paralog–ortholog K_S distributions

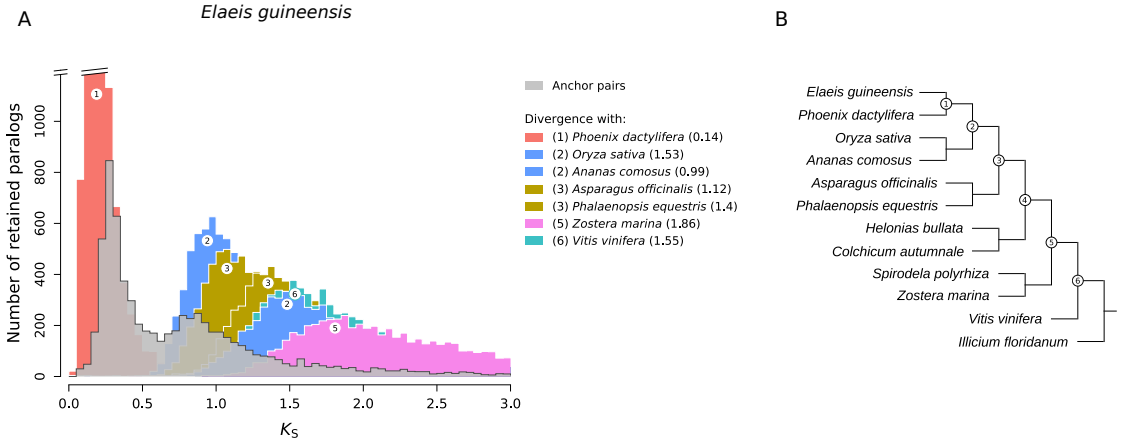
Cecilia Sensalari[1,2], Steven Maere[1,2,*] and Rolf Lohaus[1,2,*]

[1]Dept. of Plant Biotechnology and Bioinformatics, Ghent University, 9052 Ghent, Belgium
[2]VIB-UGent Center for Plant Systems Biology, 9052 Ghent, Belgium

∗ co-last authors

Contents

1 Supplementary figures



Supplementary Figure 1: Panel A shows a naive mixed paralog–ortholog K_S plot for oil palm (*Elaeis guineensis*) without any substitution rate-adjustment. The ortholog distributions representing the same speciation in the phylogeny share color and number. Panel B shows the input phylogenetic tree.

2 Supplementary methods

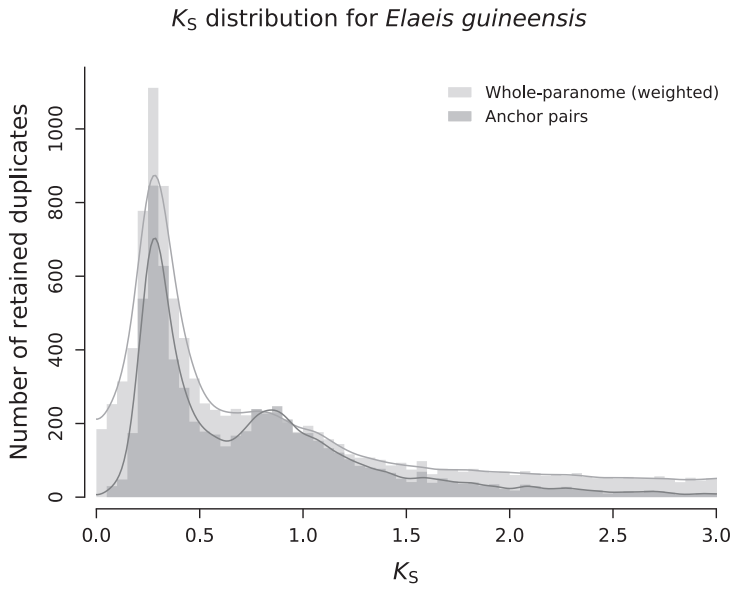
2.1 Construction of paralog and ortholog K_S distributions

The calculation of paralog and ortholog K_S values is performed by the *wgd* package (Zwaenepoel *et al.*, 2019), which largely follows the approach outlined in (Vanneste *et al.*, 2013). Genomic or transcriptomic CDS sequence data for each species need to be provided in FASTA format. One-to-one orthologs are recovered from all-versus-all BLASTp searches using the reciprocal best BLAST hit criterion. Paralogous gene families are constructed from all-versus-all BLASTp results using MCL clustering (Enright *et al.*, 2002). Paralogous gene families with more than 200 members are excluded by default from further analysis (customizable parameter). For the multiple sequence alignment step, *ksrates* uses the default aligner in *wgd*, MUSCLE (Edgar, 2004). K_S estimates are then calculated as described in (Zwaenepoel *et al.*, 2019). To compensate for the K_S estimate redundancy in paralog gene families (multiple K_S values may be estimated for the same duplication event (Maere *et al.*, 2005)), phylogenetic relationships in the gene family are reconstructed through FastTree (Morgan *et al.*, 2009), the default tool choice in *wgd*. Node weighting is then applied for constructing the whole-paranome K_S distribution as described in (Maere *et al.*, 2005), using weights derived from the phylogenetic relationships obtained by FastTree. *ksrates* generates K_S distributions up to K_S=5 for paralogs and K_S=10 for orthologs by default (customizable parameters).

If available, the genome structural information file (GFF3 file) for the focal species can also be provided to perform synteny analysis and construct an anchor pair K_S distribution (Supplementary Fig. 2). Anchor pairs are a subset of paralog pairs found in duplicated genomic regions with conserved gene order (collinear segments), which likely originated through a whole-genome duplication (WGD) or other large-scale duplication. i-ADHoRe (Proost *et al.*, 2012) is used to detect such collinear segments and their anchor pairs. K_S values for these anchor pairs are calculated in the same way as for ortholog and paralog pairs, and assembled in an anchor pair K_S distribution.

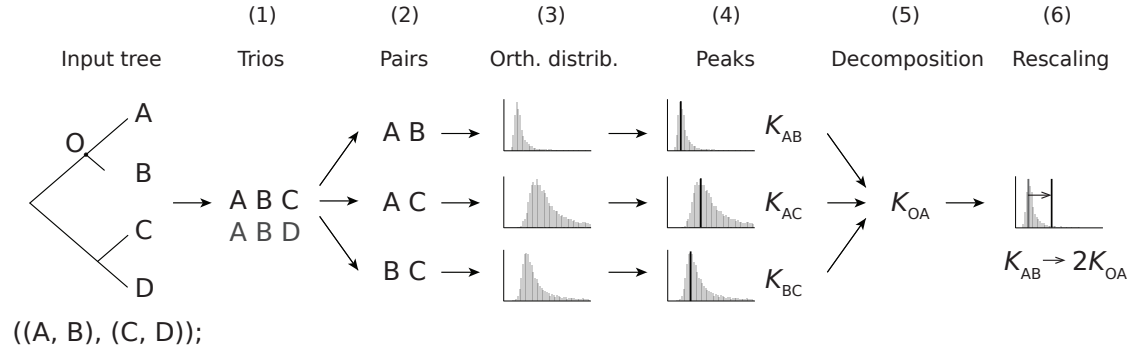
2.2 Ortholog K_S adjustment

The rate-adjustment pipeline is composed of several steps (Supplementary Fig. 3) that address first the detection of substitution rate differences between the focal species and the other species and subsequently the adjustment for such differences.



Supplementary Figure 2: Whole-paranome K_S distribution (light gray histogram and KDE curve) and anchor pair K_S distribution (dark gray histogram and KDE curve) for oil palm (*Elaeis guineensis*). Two WGD peaks are visible on the left side of the distribution.

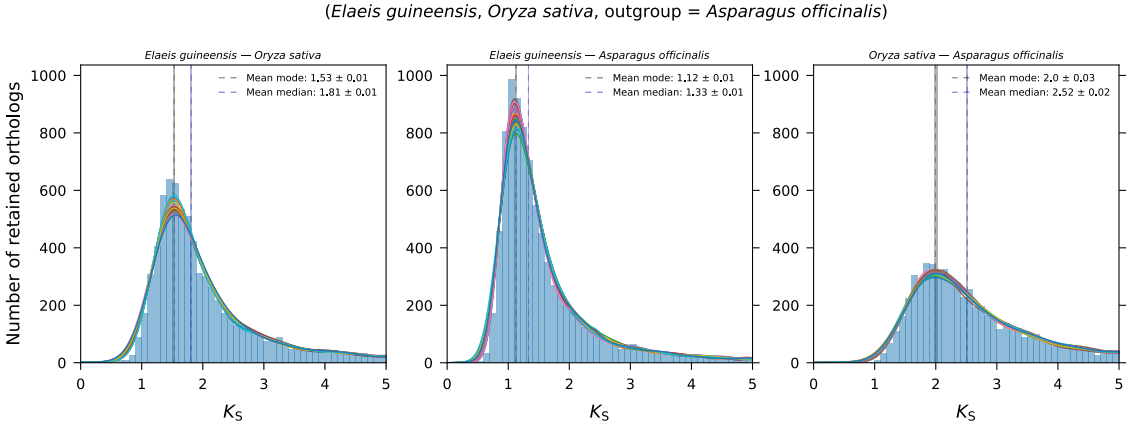
Ortholog K_S estimate decomposition Ortholog K_S adjustment requires known phylogenetic relationships between the focal species and all other species in the dataset. These relationships are provided to *ksrates* as input using the Newick tree format. Supplementary Fig. 3 shows an example input phylogenetic tree in which A is the focal species. To adjust the K_S estimate of each divergence event between the focal species and other species in the phylogeny, each raw ortholog K_S estimate is decomposed into branch-specific contributions. For example, the K_S estimate of the divergence of focal species A and species B in the example tree is decomposed into the contributions from branches O–A and O–B, with O being the last common ancestor of species A and B. For this decomposition an outgroup species is needed (here, e.g. C) and hence each ortholog species pair in the dataset is required to have at least one outgroup.



Supplementary Figure 3: Steps of the substitution rate-adjustment of ortholog K_S estimates

In the first step, the input phylogenetic tree is broken down into species trios, each composed of the focal species A, a diverged species in the dataset (here, B) and an outgroup reference (e.g. C). If the tree is large or complex enough it is possible to have more than one outgroup (here, e.g. D), and thus multiple trios, per diverged ingroup species pair.

Each species trio is then broken down into the three possible species pairs (step 2), namely a pair of the two ingroup species (here, A and B), and two pairs of each ingroup species together with the outgroup (here, e.g. A and C, and B and C). For each species pair of each species trio,



Supplementary Figure 4: K_S mode estimates of the three ortholog K_S distributions generated from the trio *Elaeis guineensis*, *Oryza sativa* and outgroup *Asparagus officinalis*. The ortholog K_S histograms are shown in light blue and the first 20 bootstrapped KDEs are depicted as colored curves. The estimated mean mode and median of each distribution are shown as dashed gray and blue lines. Colored boxes around these estimates range from one standard deviation (sd) below to one sd above the mean mode or median estimate. Note that the *Elaeis guineensis*–*Asparagus officinalis* distribution has a much younger mode than the *Oryza sativa*–*Asparagus officinalis* distribution, suggesting that *Elaeis guineensis* has a lower synonymous substitution rate than *Oryza sativa*.

the one-to-one orthologs are detected and their K_S values estimated using the *wgd* package, and an ortholog K_S distribution is built (step 3, also see Supplementary Fig. 4).

A single K_S estimate for the divergence time of each species pair is then obtained from its ortholog K_S distribution (step 4). By default, we use bootstrapped kernel density estimation (KDE) to estimate the mode of the ortholog K_S distribution. The ortholog K_S data is bootstrapped 200 times (customizable parameter) and each time the mode of a KDE with Gaussian kernel is computed. The final divergence K_S estimate is then calculated as the mean of the bootstrapped modes together with an associated standard deviation. Alternatively, the tool can be configured to use the median of each bootstrapped ortholog K_S data sample instead of the KDE mode (Supplementary Fig. 4). We denote the three divergence K_S estimates in our example trio as K_{AB}, K_{AC} and K_{BC}.

To decompose the ortholog K_S estimate of the ingroup species pair, K_{AB}, we use simple equations from relative rate testing (Graur, 2016; Sarich and Wilson, 1973). The ortholog K_S estimate K_{AB}, i.e. the number of synonymous substitutions per synonymous site between the focal species A and a diverged species B, is the sum of the number of substitutions that occurred on branch O–A (K_{OA}) and on branch O–B (K_{OB}) (see example tree in Supplementary Fig. 3):

$$K_{AB} = K_{OA} + K_{OB} \tag{1}$$

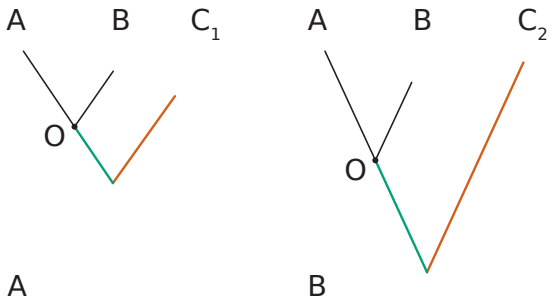
With the help of the outgroup reference species in each trio (here, e.g. C), and the two corresponding ortholog K_S estimates K_{AC} and K_{BC}, we can calculate the value of K_{OA}:

$$K_{OA} = \frac{K_{AC} + K_{AB} - K_{BC}}{2} \tag{2}$$

K_{OA} is an estimate of the number of substitutions that occurred on branch O–A, i.e. the contribution of the branch O–A to K_{AB}, and depends only on the synonymous substitution rate experienced in the lineage leading to focal species A.

The error associated to K_{OA} can be derived from the estimated standard deviations (s) through error propagation rules:

$$s_{K_{OA}} = \frac{\sqrt{s^2_{K_{AC}} + s^2_{K_{AB}} + s^2_{K_{BC}}}}{2} \tag{3}$$



Supplementary Figure 5: Species C_1 in panel A is a better outgroup candidate than species C_2 in panel B because it is more closely related to the two ingroup species A and B (shorter O–root segment, in green) and because it has a lower substitution rate (shorter root–C branch, in orange).

Ortholog K_S estimate rescaling To adjust the K_S estimate for the divergence of species A and B, K_{AB}, to the K_S-timescale of the focal species A we then simply rescale the contribution of the diverged branch K_{OB} to the same value as K_{OA}, or, in short, we rescale K_{AB} to $2K_{OA}$:

$$K_{AB} \rightarrow K_{OA} + K_{OA} = 2K_{OA} \tag{4}$$

The error associated to the rescaled K_{AB} can be derived through error propagation rules:

$$s_{K_{AB}} = \sqrt{s^2_{K_{OA}} + s^2_{K_{OA}}} = \sqrt{2} s_{K_{OA}} \tag{5}$$

In other words, we calculate a rescaled K_{AB} from a hypothetical set of ortholog pairs that diverged at time point O and evolve under the same synonymous substitution rate history, the rate history experienced by the lineage leading to focal species A. Alternatively, to think of these adjustments of K_S estimates of species divergence times completely within the *paralog* K_S-timescale of a focal species, one can imagine these sequences that diverge at a time point O not as ortholog pairs in diverging species but as paralog pairs that originate in a duplication event in the lineage of the focal species A coinciding with the speciation of A and B.

When multiple ortholog K_S adjustments are computed for the same speciation event, i.e. on species trios having the same two ingroup species but different outgroup species, the consensus for the adjusted K_S estimate is taken as the mean of all the adjusted K_S estimates with its standard deviation computed via error propagation rules. Alternatively, *ksrates* can be configured to get the final adjusted K_S estimate from the "best" outgroup species, thought of as most likely providing the most reliable ortholog K_S estimate decomposition. This "best" outgroup is taken to be the outgroup with the shortest O–C branch length (Supplementary Fig. 5), i.e. smallest K_{OC}, which is internally computed using an equation similar to Equation 2. The more closely related the outgroup is to the two ingroups, and the lower its rate is, the better it is as an outgroup reference species. One should thus strive to pick such species to compile a good input dataset. All calculated values for all species trios, including adjusted ortholog K_S, K_{OA} and K_{OC}, are also written as tabular data to an output file. These can thus be inspected for consistency and used to manually calculate and use alternative adjustments. The structure of this table is explained in Supplementary Table 1.

Ortholog K_S adjustment results in a shift of the estimated K_S position of a speciation event proportional to the estimated substitution rate difference between the diverged lineage and the lineage of the focal species. In case the focal species has a lower rate, the adjustment scales down the excess contribution of the diverged species to the K_{AB} value, and the adjusted divergence K_S estimate will be smaller than the original estimate. In case the focal species has a higher rate, the adjustment scales up the lower contribution of the diverged species to the K_{AB} value, and the adjusted divergence K_S estimate will be larger than the original estimate.

Node	Species	Sister_Species	Out_Species	Ks_OC	Mode	SD	Ks_Species	Ks_Sister
1	Elaeis guineensis	Phoenix dactylifera	Oryza sativa	1.458204	0.148908	0.013058	0.074454	0.063973
1	Elaeis guineensis	Phoenix dactylifera	Ananas comosus	0.917902	0.145957	0.011897	0.072978	0.065448
1	Elaeis guineensis	Phoenix dactylifera	Asparagus officinalis	1.038387	0.17099	0.015711	0.085495	0.052932

Supplementary Table 1: First rows of the correction table obtained for *Elaies guineensis*: each row shows the correction data between oil palm and *Phoenix dactylifera* with the use of a different outgroup (*Oryza sativa, Ananas comosus or Asparagus officinalis*). In column `Ks_OC`, the synonymous distance accumulated in the O–C branch relative to the given outgroup is shown. In column `Mode` and `SD` the rate-adjusted modes are shown with their standard deviation. In columns `Ks_Species` and `Ks_Sister` the synonymous distances accumulated in the *Elaies guineensis* and *Phoenix dactylifera* lineages are shown, respectively.

2.3 Mixture modeling of paralog K_S distributions

The interpretation of mixed paralog–ortholog K_S distributions is sometimes challenged by the fact that paralog WGD peaks are often not clearly distinguishable due to progressive WGD signal erosion over time and due to potential overlaps between peaks of successive WGDs. In order to more objectively define the K_S age of WGD peaks, a clustering feature based on mixture modeling has been implemented in *ksrates*.

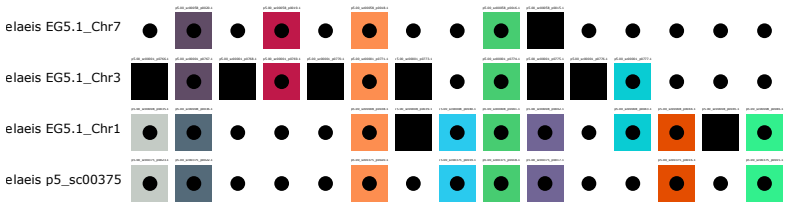
Depending on the available input data and analysis configuration (`paranome` and `collinearity` settings in the *ksrates* configuration file), different methods are applied to analyze the paralog K_S distribution(s). If only collinearity analysis is selected (`paranome = no` and `collinearity = yes`), the default method performed is a clustering based on the anchor pair K_S values in collinear segment pairs (see below). Lognormal mixture modeling of the anchor pair K_S distribution is optional. If only whole-paranome analysis is selected in the *ksrates* configuration file (`paranome = yes` and `collinearity = no`), exponential-lognormal mixture modeling of the whole-paranome K_S distribution is performed by default. Lognormal-only mixture modeling is optional. If both analysis types are selected in the configuration file, the anchor pair K_S clustering is performed by default and the others are optional. Supplementary Table 2 summarizes this.

Method	Collinearity-only	Paranome-only	Collinearity and paranome
Anchor K_S clustering	**X**		**X**
Exponential-lognormal mixture model		**X**	x
Lognormal mixture model on anchor pairs	x		x
Lognormal mixture model on paranome		x	x

Supplementary Table 2: Mixture model techniques used to analyze paralog K_S distribution signals. The applied methods depend on the required analysis type (`collinearity` and/or `paranome`) and on optional request for additional analyses (expert mode). Default methods are marked by a bold capitalized `X`, optional methods are marked with a lower-case `x`.

2.3.1 Anchor K_S clustering

In order to classify anchor pair K_S values into groups tentatively representing different WGDs in the focal species' ancestry, *ksrates* uses a clustering approach. *ksrates* does not cluster the anchor pair K_S values directly, but instead clusters median K_S values for the collinear segment pairs, i.e. pairs of sequence regions with conserved gene content and order, that the anchor pairs reside on. The collinear segments are detected and aligned by i-ADHoRe (Proost *et al.*, 2012), generating so-called multiplicons (Supplementary Fig. 6). The number of segments aligned in a multiplicon defines its level. A multiplicon may contain segment pairs that originated through different WGD



Supplementary Figure 6: Example of an i-ADHoRe multiplicon obtained from *Elaeis guineensis*, composed of four collinear segments (multiplicon level 4). Aligned anchor genes are depicted by boxes of the same color. Non-anchor genes in the segments are shown in black and alignment gaps in white.

events, but each pair of segments in a multiplicon traces back to a single WGD event.

Before the clustering of segment pairs, a cleaning step is performed. Some segment pairs are (partially) redundant because of the way multiplicons are generated. In the filtering step, segment pairs are compared between multiplicons and redundant ones are discarded according to the following criteria:

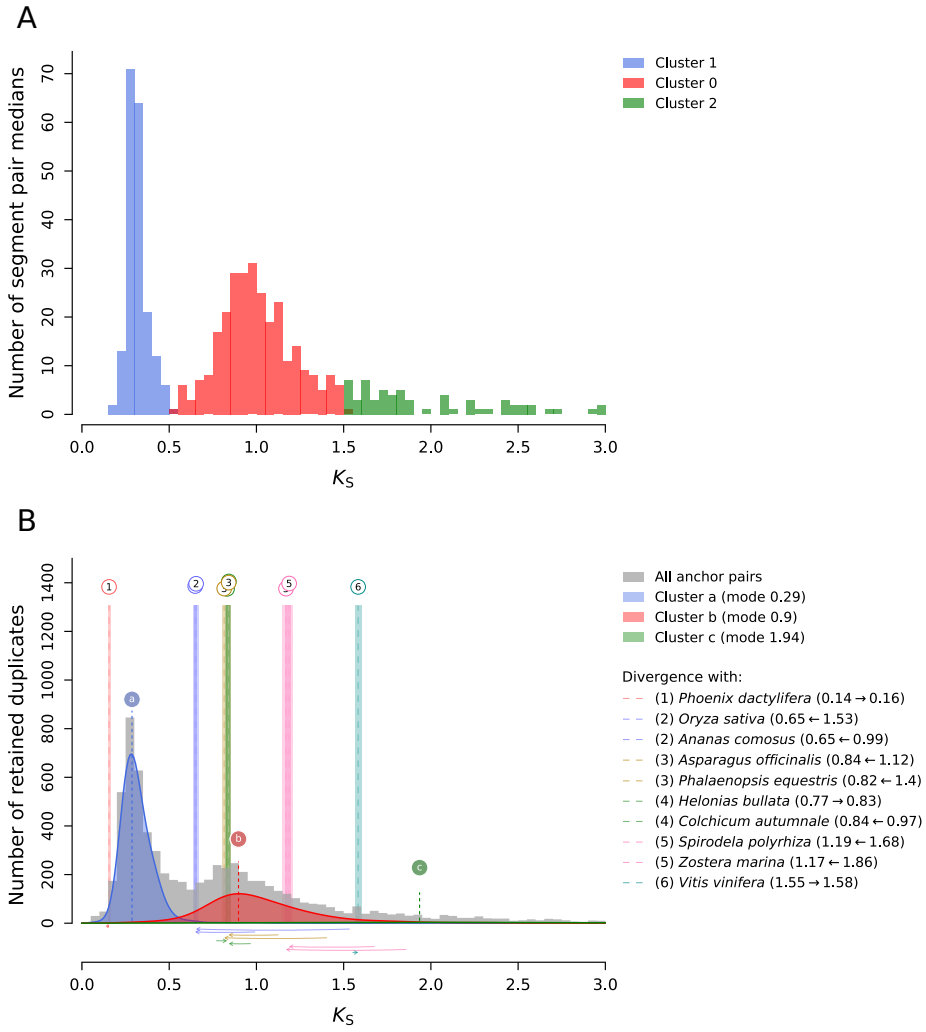
- Segment pairs whose anchor-pair list is a subset of the anchor-pair list of another segment pair are removed because they are fully redundant.
- If a segment pair has an anchor-pair list that partially overlaps ($> 1/3$) with the anchor-pair list of another segment pair, the segment pair with the shorter anchor-pair list is removed.

Smaller overlaps are tolerated in order not to lose too much collinear information. Note that the unitary element to be removed during the filtering is the entire segment pair and not individual redundant anchor pairs on it, to be consistent with the fact that collinearity traces are structured in blocks.

After obtaining the cleaned dataset, each segment pair is assigned a representative K_S age. Even though all anchor pairs laying on a segment pair result from the same large-scale duplication, the K_S estimates of the individual anchor pairs can vary substantially, e.g. due to stochastic effects in the synonymous substitution process, K_S saturation effects and K_S estimation errors (Vanneste *et al.*, 2013), and the resulting estimates frequently contain outliers. Anchor-pair K_S lists with more than 5 elements are therefore pruned by removing the values falling outside the interval defined by the median $\pm$ median absolute deviation. Then, the representative K_S age of each segment pair is computed as the median of the remaining anchor-pair K_S list.

The segment-pair median K_S values are log-transformed and then clustered through Gaussian mixture modeling (GMM), which is equivalent to employing lognormal mixture modeling on the untransformed data. WGD peaks in K_S distributions are preferentially modeled by lognormal distributions due to the fact that they generally exhibit positive skewness (Tiley *et al.*, 2018; Morrison, 2008). The clustered datapoints are back-transformed for cluster visualization on the original non-log K_S scale (Supplementary Fig. 7A).

GMM requires the number of clusters to be found to be set in advance. Usually, GMM is performed multiple times with a different target number of clusters, and the best number of clusters is selected based on e.g. the Bayesian Information Criterion (BIC) or Akaike's Information Criterion (AIC). As this strategy in practice tends to overestimate the number of clusters (and hence WGDs), *ksrates* follows an alternative, more pragmatic approach. As the number of clusters should reflect the number of (detectable) WGDs in the lineage under study, *ksrates* estimates this number of WGDs beforehand based on the highest multiplicon level reached in the collinearity analysis. For example, the presence of 8 collinear segments can be explained by three whole-genome duplications. This approach is limited by the fact that a given maximum multiplicon level may be caused by different duplication scenarios, in particular taking into account that extensive post-WGD rearrangements and duplicate gene and segment losses are known to occur over evolutionary time. A maximum multiplicon level of 8 may for instance be caused by three whole-genome duplications



Supplementary Figure 7: Panel A shows a GMM clustering of segment pair medians for *Elaeis guineensis*, where three clusters have been detected (blue, red and green). Panel B shows the mixed paralog–ortholog K_S plot after each median has been substituted by its original anchor-pair K_S list. The K_S clusters (filled blue, red and green KDE curves) are labeled with letters. The original anchor pair K_S distribution is visible as a gray histogram in the background.

or by two whole-genome triplications (theoretical multiplicon level 9) followed by extensive gene loss. *ksrates* takes an upper boundary of the minimum number of duplication events across these scenarios as the GMM cluster number estimate, by assuming only whole-genome duplications have happened. A maximum multiplicon level of 9 for instance requires at least 4 whole-genome duplications, although it may be explained by e.g. 2 whole-genome triplications or 2 whole-genome duplications and a triplication. Superfluous clusters caused by overestimating the number of WGDs are filtered out afterwards to the extent possible, as described below.

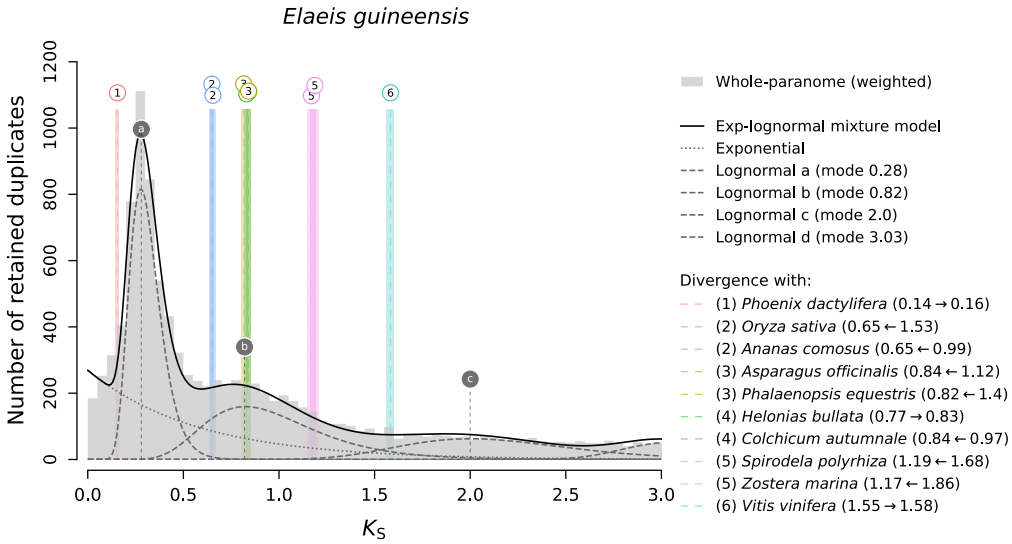
Given the cluster number estimated from the maximum multiplicon level, the GMM is initialized and fitted on the segment-pair medians K_S dataset multiple times, and the best model is taken to be the one with the largest log-likelihood. Subsequently, the segment-pair median K_S values are replaced by the original K_S list for the segment pair to obtain the anchor K_S clusters (Supplementary Fig. 7B). After initial clustering, anchor K_S clusters for which a link to a real WGD event is ambiguous or unlikely are removed from the dataset, along with the segment pairs defining them. Removed clusters meet at least one of the following empirical criteria (thresholds are based on cluster shapes obtained for test species, see section 4):

- The cluster is poorly populated (it contains <= 10% of the total number of K_S values). These are likely clustering artifacts, in particular when the cluster has a young median K_S age.
- The cluster K_S estimate is too old to be reliably associated with a WGD (the median K_S of the cluster is >= 3). Although ancient WGD events may have happened beyond K_S>= 3, finding reliable evidence for those in K_S plots is challenging, as K_S peaks beyond K_S>= 3 are increasingly likely to be artifacts caused by K_S saturation effects (Vanneste *et al.*, 2013).
- The cluster has a flat, stretched-out peak signal (the inter-quartile range of the cluster is >= 1.1 K_S). These are likely artifacts, in particular when the cluster has a young median K_S age.

These criteria mostly remove clustering artifacts in the anchor K_S distribution tail and small K_S clusters that overfit particular distribution features, often at K_S values close to 0 or in between K_S peaks that represent genuine WGDs. If one or more clusters are filtered out, a second round of GMM is performed on the remaining segment pairs using the remaining number of clusters. For example, the green 'c' cluster in Supplementary Fig. 7 is removed because it matches two of the aforementioned criteria (see also Fig. 1A in the main text).

2.3.2 Exponential-lognormal mixture model

ksrates implements the expectation-maximization algorithm described in Zhang *et al.* (2019) to fit exponential-lognormal mixture models to whole-paranome K_S distributions. The exponential component models the L-shaped background K_S distribution generated by small-scale duplications (SSDs)(Blanc and Wolfe, 2004), while the lognormal components model WGD peaks (Supplementary Fig. 8). The modes of the lognormal components are taken as a proxy for the WGD age. The K_S range in which the fitting is applied can be varied by the user.



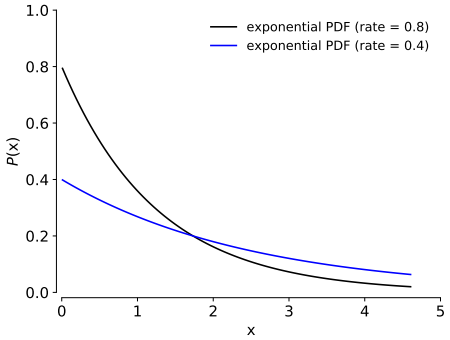
Supplementary Figure 8: Mixed paralog–ortholog K_S plot with superimposed exponential-lognormal mixture model. The overall mixture model (dark solid line) is composed of an exponential component (dotted gray curve) and lognormal components (dashed gray curves). Each lognormal component is labeled with a letter (vertical dashed gray lines with circular labels). Components a and b are matching the expected position of the two visible WGD peaks. The "buffer" component d covers the right-most side of the K_S range (label not shown on figure).

Since adequate initialization of the component parameters is crucial for obtaining decent mixture modeling results, *ksrates* uses three different initialization approaches and ultimately chooses

the best one based on BIC. In all three strategies, an extra "buffer" lognormal component is initialized by default at the right boundary of the K_S range to avoid that other components stretch towards higher values in an attempt to fit the hard-to-fit distribution tail.

Data-driven initialization In this method the initialization of component parameters is guided by the shape of the paranome K_S distribution. The paranome K_S histogram is converted from count data to a probability density histogram for the component initialization.

The initialization of the exponential component takes advantage of the fact that the intercept of the exponential probability density function with the y-axis is equal to the decay rate of the function (Supplementary Fig. 9). The height of the first histogram bin, considered here as an approximation of the intercept, is thus used to initialize the decay rate of the exponential component. The bin width—which indirectly influences the bin height—is set to 0.1 K_S, a traditional choice for representing K_S distributions (Lynch and Conery, 2000). This width is narrow enough to have a good resolution on the left edge of the probability density of the K_S distribution and at the same time it is wide enough to reduce the risk of catching possible (artificial) high densities close to 0 (often caused by sequencing artifacts) followed by a sudden drop.



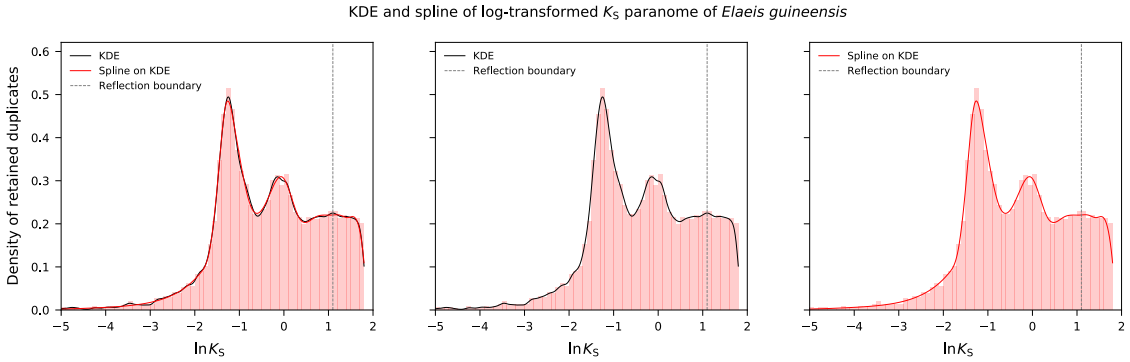
Supplementary Figure 9: The decay rate of the exponential probability density function (black and blue curves) is equal to the intercept with the y-axis.

To initialize the lognormal means and standard deviations, the paranome distribution is first log-transformed so that the putative lognormal WGD peaks assume a Gaussian shape. Then, the transformed distribution is searched for peaks as follows. First, a KDE is computed on an extended distribution in which data on the original right boundary is reflected to the other side, to prevent KDE edge effects (Supplementary Fig. 10). Subsequently, a smoothing spline is computed on the KDE in order to smooth out small irregularities (Supplementary Fig. 10). The spline may still exhibit small noise peaks. In the attempt to filter away these small peaks and retain only presumedly real WGD peaks, the distribution is mirrored around each peak in both directions (Supplementary Fig. 11B and C). The peak is retained as a possible WGD peak only if its prominence in the mirrored background is significant in at least one reflection. The significance threshold (0.06) is based on empirical results coming from test species (see section 4). The log K_S-coordinates of significant peaks are then taken as the mean of the Gaussian components, while the associated peak width is taken as a proxy for the standard deviation.

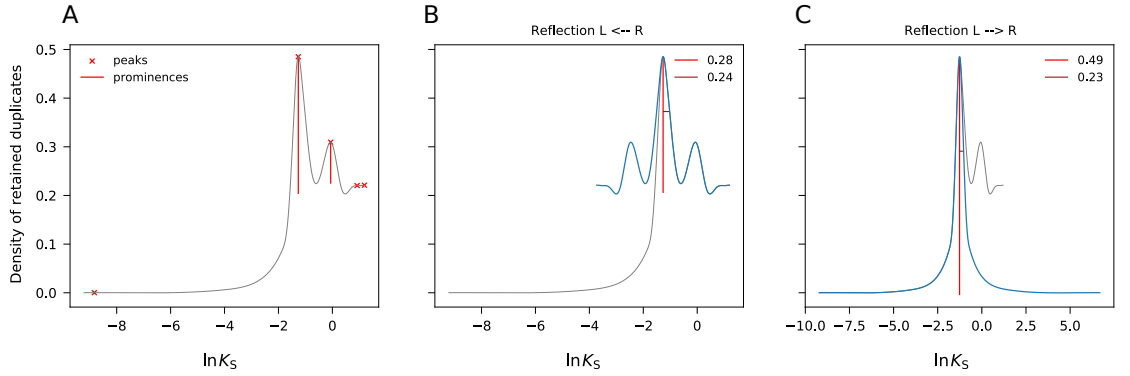
Random initialization In the random initialization method, the components are initialized by randomly drawing their parameter values from a range of reasonable values empirically obtained from test species (see section 4):

- the exponential rate is chosen between 0.2 and 1 in steps of 0.1
- the normal mean is chosen between -0.5 and 0.9 in steps of 0.1
- the normal standard deviation is chosen between 0.3 and 0.9 in steps of 0.1

A mixture model is by default fitted with two to five random components (in addition to the "buffer" component which is always present). For each number of components, the mixture model is initialized multiple times and the best fit is chosen according to the lowest BIC score.



Supplementary Figure 10: KDE (in black) and spline (in red) obtained from the log-transformed paranome K_S distribution of *Elaeis guineensis*. The spline smooths out irregularities in the KDE and is later used for data-driven peak detection. The underlying paralog distribution has been partially reflected across the right boundary (dashed vertical line) to account for edge effects.



Supplementary Figure 11: Panel A shows a whole-paranome smoothing spline (in gray) for *Elaeis guineensis* with peaks (red crosses) and prominences (red lines) highlighted. Panels B and C show the reflections (blue curves) around the tallest peak in both directions. The peak has a significant prominence in both reflections and is therefore used as a normal component mean. The peak width (horizontal dark red line) is used to initialize the standard deviation of the component.

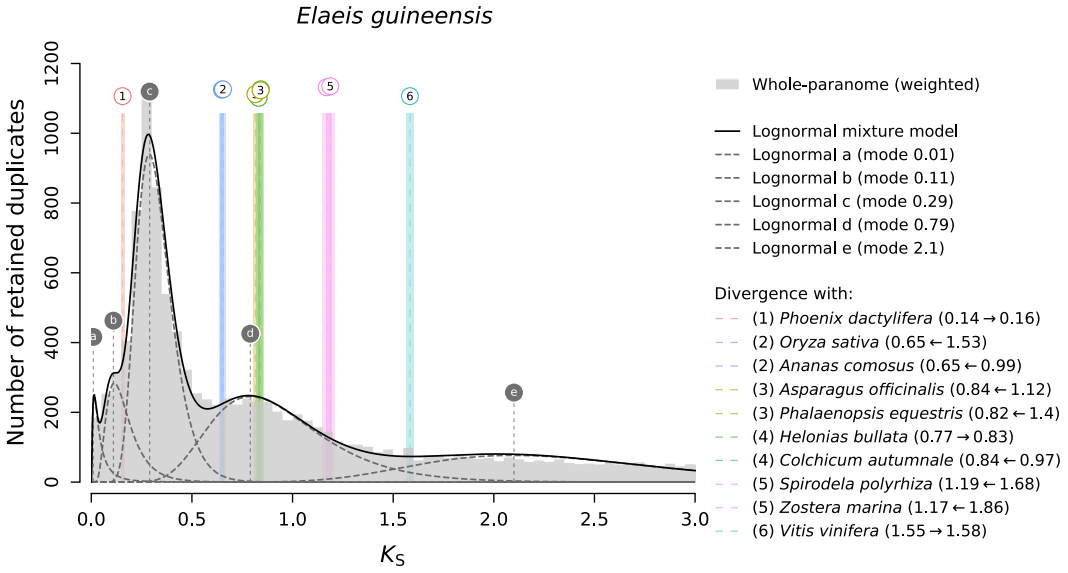
Hybrid initialization For the hybrid method, the mixture model is initialized with the components previously computed through the data-driven approach, but with the addition of a single lognormal component whose parameters are randomly drawn from the ranges used in the random initialization method. This method is an attempt to reintroduce peaks that were overlooked by the data-driven approach. The mixture model is initialized multiple times and the best fit is chosen according to the lowest BIC score.

Model evaluation After having run models with all three initialization approaches, the model with the lowest BIC value is incorporated in the mixed K_S distribution plot (Supplementary Fig. 8).

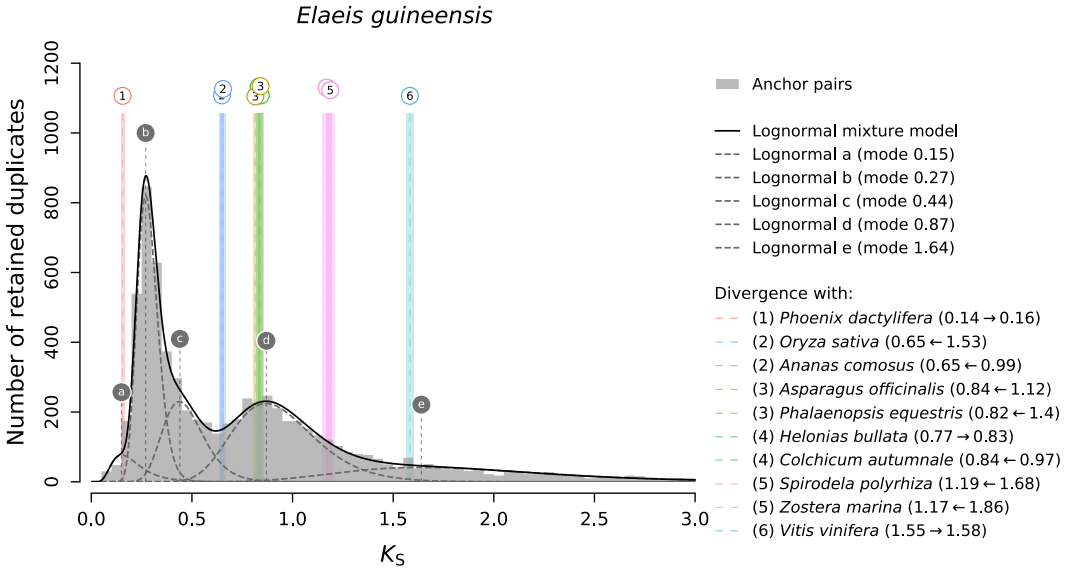
2.3.3 Lognormal mixture models

In the lognormal mixture model approach used optionally for both whole-paranome K_S distributions (Supplementary Fig. 12) and anchor pair K_S distributions (Supplementary Fig. 13), a Gaussian mixture model is fitted on log-transformed K_S data, which are subsequently back-transformed to obtain the lognormal mixture model.

Lognormal-only mixture models are by default fitted with two to five components. For each number of components the mixture model is initialized multiple times and the best fit is chosen according to the largest log-likelihood. Among the resulting four models (one for each number of components), the best fitting model is taken to be the one with the lowest BIC score.



Supplementary Figure 12: Mixed paralog–ortholog K_S plot showing the whole-paranome K_S distribution for *Elaeis guineensis* in light gray with superimposed the best-fitting lognormal-only mixture model. The overall lognormal mixture model (solid black curve) is composed of multiple components (dashed gray curves), which are labeled with letters (vertical dashed gray lines with circular labels).



Supplementary Figure 13: Mixed paralog–ortholog K_S plot showing the anchor pair K_S distribution for *Elaeis guineensis* in dark gray with superimposed the best-fitting lognormal-only mixture model. The overall lognormal mixture model (solid black curve) is composed of multiple components (dashed gray curves), which are labeled with letters (vertical dashed gray lines with circular labels).

3 Data sources

Sources of input CDS FASTA and GFF data for all species used in the use case analysis (Fig. 1 in the main text and most supplementary figures here) are provided in Supplementary Table 3. Genome data were used for most of the species, while transcriptome data were used for *Helonias bullata*, *Colchicum autumnale* and *Illicium floridanum*.

Species	Version / Accession	Source database
Elaeis guineensis	EG5.1	Monocots PLAZA 4.5 (Locus FASTA Data and Structural Annotation GFF) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Phoenix dactylifera	GCF_000413155.1	NCBI ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/413/155/GCF_000413155.1_DPV01
Oryza sativa ssp. japonica	v7_JGI	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Ananas comosus	v3	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Asparagus officinalis	v1.1	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Phalaenopsis equestris	v1.0	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Helonias bullata	OOSO	One Thousand Plants Project (1KP) http://www.onekp.com/public_data.html
Colchicum autumnale	QNPH	One Thousand Plants Project (1KP) http://www.onekp.com/public_data.html
Spirodela polyrhiza	v2	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Zostera marina	v2.2	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Vitis vinifera	Genoscope.12X	Monocots PLAZA 4.5 (Locus FASTA Data) https://bioinformatics.psb.ugent.be/plaza/versions/plaza_v4_5_monocots/download
Illicium floridanum	VZCI	One Thousand Plants Project (1KP) http://www.onekp.com/public_data.html

Supplementary Table 3: Data source table. The left column specifies the species, the middle column specifies the genome version or the data accession code and the right column specifies the source database and the link to the associated website's download page.

4 Parameters used

For the construction and analysis of the *Elaeis guineensis* mixed plot the following parameters were set in the *ksrates* configuration file. The paranome and anchor pair K_S distributions have been constructed up to K_S=5 and plotted up to K_S=3, using a histogram bin width of K_S=0.05. Ortholog K_S distributions were constructed up to K_S=10 and plotted up to K_S=5. Modes of the ortholog K_S distributions were estimated using 100 bootstrapped KDEs. Each ortholog K_S estimate was adjusted using up to three (closest) outgroup species and the mean of these adjusted values was taken as the consensus adjusted value.

To obtain the mixture models, the number of initializations for the expectation-maximization algorithm was set to 10, the maximum number of iterations was set to 300 and the convergence

value was set to 1e-6. The maximum number of components was set to 5 and the maximum K_S of the range in which to perform the mixture modeling was set to K_S=3. This range is internally increased by 0.5 K_S during mixture modeling to avoid edge effects by truncating the distribution at the end of the region of interest.

Besides on *Elaeis guineensis*, the mixture models have also been tested on other test species: *Arabidopsis thaliana*, *Oryza sativa*, *Musa acuminata*, *Ananas comosus* and *Asparagus officinalis* (data not shown).

5 Availability

ksrates is implemented in Python 3 and is available under the GNU GPL v3 license in a public repository at the VIB-PSB GitHub page (`https://github.com/VIB-PSB/ksrates`). Linked documentation explains installation, container usage, pipeline configuration and input and output files, and contains a tutorial based on an example dataset that is available for download (see below).

The full workflow implemented in *ksrates* is made available as a Nextflow (`https://www.nextflow.io/`) pipeline that eases execution on a variety of compute clusters such as SGE. In addition, *ksrates* can be executed manually using a command-line interface. *ksrates* is available as Docker and Singularity containers that bundle all required external software dependencies, and as a Python package to allow integration into existing genomics toolsets and workflows.

We also provide example datasets. A small test dataset composed of truncated sequence files can be used to perform a quick check whether *ksrates* is correctly installed and fully functioning, which should take only a few minutes to run. A full example dataset contains the sequence data needed to reproduce the use case scenario described in the tutorial and documentation. This dataset requires the use of a compute cluster, as running *ksrates* to process it using an average laptop or desktop may take several hours.

6 Supplementary references

Blanc,G. and Wolfe,K.H. (2004) Widespread paleopolyploidy in model plant species inferred from age distributions of duplicate genes. *Plant Cell*, **16**, 1667–1678.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, **32**, 1792–1797.

Enright,A.J. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, **30**, 1575–1584.

Graur,D. (2016) *Molecular and Genome Evolution*. Sinauer Associates, Sunderland (Massachusetts).

Lynch,M. and Conery,J.S. (2000) The evolutionary fate and consequences of duplicate genes. *Science*, **290**, 1151–1155.

Maere,S. *et al.* (2005) Modeling gene and genome duplications in eukaryotes. *Proc Natl Acad Sci U S A*, **102**, 5454–5459.

Morgan,N.P. *et al.* (2009) FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. *Mol Biol Evol*, **26**, 1641–1650.

Morrison,D.A. (2008) How to summarize estimates of ancestral divergence times. *Evol Bioinform Online*, **4**, 75–95.

Proost,S. *et al.* (2012) i-ADHoRe 3.0—fast and sensitive detection of genomic homology in extremely large data sets. *Nucleic. Acids. Res.*, **40**, e11.

Sarich,V.M. and Wilson,A.C. (1973) Generation time and genomic evolution in primates. *Science*, **179**, 1144–1147.

Tiley,G.P. *et al.* (2018) Assessing the performance of *Ks* plots for detecting ancient whole genome duplications. *Genome. Biol. Evol.*, **10**, 2882–2898.

Vanneste,K. *et al.* (2013) Inference of genome duplications from age distributions revisited. *Mol Biol Evol*, **30**, 177–190.

Zhang,X. *et al.* (2019) Lognormal-based mixture models for robust fitting of hospital length of stay distributions. *Oper Res Health Care*, **22**, 100184.

Zwaenepoel,A. and Van de Peer,Y. (2019) wgd—simple command line tools for the analysis of ancient whole-genome duplications. *Bioinformatics*, **35**, 2153–2155.