Learning the fitness dynamics of pathogens from phylogenies

Authors:
Noémie Lefrancq[1,2.3,*], Loréna Duret[1], Valérie Bouchez[4,5], Sylvain Brisse[4,5], Julian Parkhill[2,+], Henrik Salje[1,+]

Affiliations:

1. Department of Genetics, University of Cambridge, Cambridge, UK
2. Department of Veterinary Medicine, University of Cambridge, Cambridge, UK
3. Department of Biosystems Science and Engineering, ETH Zurich, 4009, Basel, Switzerland
4. Institut Pasteur, Université de Paris, Biodiversity and Epidemiology of Bacterial Pathogens, Paris, France
5. National Reference Center for Whooping Cough and Other Bordetella Infections, Paris, France

+ Joint senior authors
* Corresponding author: ncmjl2@cam.ac.uk

Introductory paragraph
The dynamics of pathogen genetic diversity, including the emergence of lineages with increased fitness, is a foundational concept of disease ecology with key public health implications. However, the identification of such lineages and estimation of associated fitness remain challenging, and is rarely done outside densely sampled systems[1,2]. Here, we present a novel scalable approach (*phylowave)* that summarises changes in population composition in phylogenetic trees, allowing for the automatic detection of lineages based on shared fitness and evolutionary relationships. We present our approach on a broad set of viruses and bacteria (SARS-CoV-2, H3N2 influenza, *Bordetella pertussis* and *Mycobacterium tuberculosis*), which includes both well-studied and understudied threats to human health. We show that *phylowave* recovers the main known circulating lineages for each pathogen, and can detect specific amino acid changes linked to fitness changes. Additionally, *phylowave* identifies previously undetected lineages with increased fitness, including three co-circulating *Bordetella pertussis* lineages. Inference using *phylowave* is robust to uneven and limited observations. This widely applicable approach provides an avenue to monitor evolution in real-time to support public health action and explore fundamental drivers of pathogen fitness.

One sentence summary
Using an agnostic approach we shed light on changes in population composition in phylogenetic trees, allowing for the automatic detection of emerging lineages and estimation of fitness dynamics.

Main text

For most pathogens, there are constantly changing patterns of strain composition. Pressures to evade host immunity, environmental shifts or changing abilities to infect and disseminate in hosts result in the emergence of some lineages with increased fitness and the extinction of others. These dynamic patterns of genetic diversity are a fundamental aspect of disease ecology. They also have potentially critical public health implications, including signifying immune or vaccine escape or improved transmissibility. It has, however, been difficult to identify lineages with differential levels of fitness, *i.e.* different ability to spread in the population, especially outside highly genetically sampled pathogen systems such as SARS-CoV-2 or influenza[1–3]. Identifying lineages with improved fitness at the population level would allow focused public health response, through *e.g.*, targeted vaccination, as well as provide key insights into the underlying ecology of disease systems.

Existing methods to monitor the fitness of strains at the population level rely on independently defined strain or clade definitions, for example, Pango lineages[4] or Nextstrain Clades[5], the global clades for influenza[6], or strains defined by pre-determined single mutations for *Bordetella pertussis*[7]. Strain fitness can be estimated using models that capture the changing proportion of individual lineages through time, typically with multinomial logistic models. These models are computationally efficient and provide key insights, for example, to track the effect of amino-acid substitutions[8], or vaccine implementation[3,9] on fitness. However, these approaches rely on an ability to group individual sequences into different lineages, which is usually based on consensus opinion, arbitrary thresholds in amino acid difference and importantly, unlinked to underlying differences in fitness. This is problematic as it means we are not reliably capturing emergent lineages with increased fitness.

Phylogenetic tree-based methods provide an alternative strategy to uncover strain fitness. Strains with increased fitness will transmit more frequently, leading to a higher branching rate in the phylogeny and more sampled descendants. The fitness of lineages can therefore be inferred from their branching pattern in a phylogeny using phylodynamic approaches such as birth-death models[10]. Multi-type birth-death models extend this idea by allowing the birth and death rate of lineages, and thereby fitness, to depend on a lineage's state or type, which may be known (e.g. genotype, mutations[11,12]) or inferred[13]. However, these models are computationally challenging to run, especially given the large amount of data now being generated. They are also susceptible to sampling biases in both space and time, which are common in phylogenetic analyses. There are alternative approaches that focus on the broad population structure[14] or changes in effective population size[15] but are not able to capture lineage fitness. Other works[1,10,16] have been done at a more granular level, but do not allow for a broad understanding of fitness changes through time.

Here we present *phylowave*, a novel agnostic approach that summarises the changes in population composition in phylogenetic trees through time, allowing for the automatic detection of circulating lineages based on differences in fitness, which we quantify and link back to specific amino acid changes. We initially explore the robustness of our approach using a simulation study where the underlying fitness difference between strains is known. We then apply this approach to SARS-CoV-2, influenza H3N2, *Bordetella pertussis (B. pertussis)* and *Mycobacterium tuberculosis (M. tuberculosis)*. We selected these respiratory pathogens as they present a diverse set of viruses and bacteria at both local and global scales, and include both well-studied and understudied threats to human health. Taking each pathogen in turn, we use *phylowave* to make critical insights into the set of discrete

lineages circulating over time, their individual fitness, as well as the genomic changes linked to quantified shifts in fitness.

Agnostic identification of lineages. *Phylowave* builds on a genetic distance-based index that measures the epidemic success of each node (internal or terminal) in a time-resolved phylogeny (Fig. 1a)[16]. This measure is based on the expectation that nodes sampled from an emerging fitter lineage will be phylogenetically closer than the rest of the population at that time, as they will all share the same recent ancestor. The index of each node is derived from the distance distribution from that node to all other nodes that circulate at that time, weighted by a kernel with a set timescale. This weight allows us to track lineage emergence dynamically, focusing on short distances between nodes (containing information about recent population dynamics) rather than long distances (containing information about past evolution). The timescale is tailored to the specific pathogen studied and its choice will depend on the molecular signal, as well as the transmission rate. Here we used timescales ranging from months (typical of RNA viruses) to years (typical of bacteria). Using the principles of coalescent theory in structured populations[17–19], we derive the expected index dynamics through time in the case of an emerging successful lineage (Fig. 1a, derivation in Supplementary Text 1). Once we have calculated the index value for each sequence, we implement a tree partitioning algorithm using generalised additive models that finds the set of lineages (*i.e.*, groups of tips and nodes) that best explains the observed index dynamics.

To quantify the fitness of each lineage, we developed a multinomial logistic model to fit the proportion of tips and nodes that belong to each lineage through time. We assumed each lineage has a constant fitness through time, defined as its relative growth rate in the population. By taking into account lineage emergence based on their Most Recent Common Ancestor (MRCA), our model does not estimate proportions for lineages that do not exist yet in the population, as opposed to implementations in other studies[8].

To assess the performance of our approach, we repeatedly simulated phylogenetic trees where one lineage expands with a known fitness advantage compared to a background population (Extended Data Fig. 1). We found that *phylowave* was indeed able to identify fitter emerging lineages (Extended Data Fig. 2a-c). Its ability to recover lineages depends on the time between emergence and the dates of sequences, with lineages with only small fitness advantages requiring sequences covering longer time periods (Extended Data Fig. 2e). Nevertheless, where the fitness difference between the two strains was greater than 0.02/year, our method was able to consistently identify the emerging lineage, noting that for lineages with lower fitness advantages the time to become the dominant lineage is >200 years. We further found that the sampling intensity was substantially less important than the sampling period (Extended Data Fig. 2f). Finally, we compared *phylowave* with alternative approaches (fastbaps[14] and treestructure[15]) (Supplementary Fig. 1). While *treestructure* and *fastbaps* did find the emerging lineage in some cases, they always found additional lineages within the phylogenies that did not have any true selective advantage (Supplementary Fig. 2).

Application to pathogens. We applied *phylowave* to four viral and bacterial pathogens: SARS-CoV-2 (N=3129 global whole genome sequences), influenza H3N2 (N=1476 global hemagglutinin [HA] sequences), *B. pertussis* (N=1248 whole genome sequences from France) and *M. tuberculosis* (N=998 whole genome sequences from Samara, Russia[20]) (Fig. 1b-e and Extended Data Fig. 3-4). We found

that for each pathogen considered, *phylowave* produced evidence of lineages with clear fitness differences, as evidenced by sub-populations of genetically-related strains with discrete index dynamics (Figure 1b-e). Taking each pathogen in turn, we compared our lineage assignments with existing lineage definitions.

We computed the Adjusted Rand-Index (ARI) to measure the agreement between classifications, accounting for random clustering[21] (Fig. 2 and Extended Data Fig. 5). An ARI value of 1 corresponds to perfect agreement with previously defined lineages, whereas a value of 0 would be expected if clusters were assigned at random. We found that across the pathogens the level of concordance was high (ARI range 0.62-0.94). For example, the previously defined SARS-CoV-2 Variants of Concern (Alpha [B.1.1.7; 20I], Beta [B.1.351; 20H], Gamma [P.1.*; 20J], Delta [B.1.617.2/AY.*; 21A/21J], and Omicron [BA.1.1.529/BA.*; 21K]) and other previously defined sub-variants closely matched *phylowave* defined lineages (Fig. 2a and Extended Data Fig. 5)[22,23]. Lineages generated by fastbaps[14] (v1.0.8) and treestructure[15] were less consistent with these predefined lineages (Supplementary Fig. 3). The existing definitions of global H3N2 clades also closely matched *phylowave* lineages (*e.g.*, 3C.3a, 3C.2a3 and 3C.2a1b.1b), with the occasional discrepancy in the exact node of emergence (*e.g.*, 3C, 3C.2 and 3C.3). *Phylowave* also identified previously defined *B. pertussis* clades (ARI = 0.63), including those defined by changes in alleles of the promoter of the pertussis toxin (*ptxP*) and fimbriae 3 gene (*fim3*)[7]. In addition, *phylowave* identified three extra *B. pertussis* lineages with clear distinct index dynamics (Fig. 1d, pink, red and purple lineages), that have not been previously identified. Finally, we recovered the known lineages and sublineages (ARI = 0.92) that were present in the *M. tuberculosis* dataset[20,24,25].

Across the pathogens, previously defined sub-variants that reached a maximum prevalence of under 5% at any time in the datasets were generally not identified by *phylowave* (*e.g.* Eta/B.1.525, Mu/B.1.621 and EU1 for SARS-CoV-2, clades 1* of H3N2 and Central Asian Strain (CAS) and East African Indian (EAI) tuberculosis lineages). The exact limits when *phylowave* can identify discrete lineages will depend on underlying prevalence, the level of sampling and fitness differences. For example, replicating the SARS-CoV-2 analysis by continent, we did obtain *phylowave* lineages that matched previously identified variants of interest that were mainly contained to those continents and that we did not identify when using the global dataset (*e.g.* Eta/B.1.525 in Africa, Mu/B.1.621 in the Americas and EU1 in Europe) (Extended Data Fig. 6)[26–28]. These findings show that previous attempts to identify discrete lineages often resulted in lineage classifications with distinct fitness, even if the various algorithms to define the lineages did not use fitness as a metric.

We next estimated the fitness of each lineage using our logistic growth model. This simple model was able to capture the lineage dynamics of each pathogen, despite substantially different trends across the pathogens investigated (Figure 3a-d and Supplementary Fig. 4-7). We found that the underlying fitness of each emerging lineage was non-null, in line with the lineages identified having true different levels of fitness (Extended Data Fig. 7). We further computed the inferred real-time fitness of each lineage in the population. While our model estimates a constant fitness parameter for each lineage, their actual fitness through time depends on what other lineages are circulating at that time. For SARS-CoV-2, we found that lineage 1, corresponding to Omicron XBB1.5, had the best maximal real-time fitness, followed by lineages 5 and 7, corresponding to Omicron BA.5 and BA.1 (Fig. 3e and Extended Data Fig. 7). H3N2 lineages' fitness was more homogeneous across the population, with lineages persisting on average 3.9 years after their emergence (Fig. 3f and Extended Data Fig. 7)[29,30]. For *B.*

pertussis, our results are consistent with those of previous studies[3], noting that three lineages (labelled 1, 2 and 3) emerged following the implementation of a new acellular vaccine in France in 1998[31] (Fig. 3g and Extended Data Fig. 7). These three lineages had the highest fitness of all *B. pertussis* strains, pointing towards immune pressure on lineage dynamics from the new vaccine. *M. tuberculosis* lineage fitness was the most stable of the four pathogens explored, reflecting its long-established diverse population. The only exception is the comparatively recent emergence of lineages 1 and 2[20] (Fig. 3h and Extended Data Fig. 7). These lineages are rising sharply in the population, and have a relative fitness per year of 1.0057, 95%CI:[1.0055, 1.0060] and 1.00087, 95%CI:[1.00077, 1.00098], respectively.

Lineage-defining mutations. We next explored whether specific changes in the genomes were linked to lineage fitness by identifying lineage-defining mutations (Fig. 4). We defined such mutations as (i) present in at least 80% of the sequences in that lineage and (ii) not present in the ancestral lineage. We did not infer lineage-defining mutations directly from the tree as there might be some uncertainty in the exact timing of the emerging lineage, and the tree itself will depend on sequencing intensity over time. In particular, a lineage can take some time to start growing and be detectable. We instead used a comparison of sequences between different lineages to identify the specific mutations that are different. While we focus on mutations, we note *phylowave* is applicable to other covariates, both for the analysis of genotypes (*e.g*. indels, or gene gain/loss), or phenotype (*e.g.* resistance to antimicrobial drugs). For each pathogen, we looked at where those mutations are located in their genomes, and how functionally relevant each of them are. For SARS-CoV-2, we found that the highest density of lineage-defining amino-acid substitutions was located in the Receptor Binding Domain (RBD) of the spike protein, with low densities in ORF1a and ORF1b, and no mutation in ORF10 (Fig. 4a-e-i and Supplementary Fig. 8-9). Our lineage-defining mutations were consistent with those described in a previous analysis that estimated nucleotide positions linked with shifts in fitness across 6 million SARS-CoV-2 genomes[8]. We found that our screening recovered all of the 55 fittest mutations, and 86% of the top 100 fittest mutations (Fig. 4i). The mutations missed by our method are mainly linked to small subclades of variants, and they seem to have spread in those clades only. We obtained similar results with H3N2, for which most of the lineage-defining amino-acid substitutions are located in the HA1 domain (Fig. 4b-f-j and Supplementary Fig. S10). We then investigated specifically if the mutations that we found were located in previously described antigenic sites[32]. We found that the antigenic sites had the highest proportion of amino acid substitutions compared to the rest of the gene, and that within those, the Koel sites had the highest proportions of substitutions[33] (Fig. 4j). Among the Koel sites, 86% of positions (N=6, out of 7) were recovered by *phylowave*. The only position missed, 155, is the oldest variable position, and is not covered by our dataset. We also recovered the main previously-described *B. pertussis* lineage-defining mutations, namely in *ptxP* and *fim3* (Fig. 4c-g-k). Further, we found a selection of other associated mutations that had not been previously described, with two distinct non-synonymous mutations in *sphB1* being of particular interest as they suggest convergent evolution (Supplementary Fig. S11). *sphB1* encodes a protease which is involved in the extracellular release of the pertussis filamentous haemagglutinin, a *B. pertussis* acellular vaccine antigen and key host-interaction factor[34]. Overall, we found that virulence-associated genes had the highest proportion of lineage-defining mutations (Fig. 4k). Lastly, we investigated the mutations associated with the most recent clades of *M. tuberculosis* (clades 1 and 2 from Fig. 3h). As reported previously[20] we found that antimicrobial resistance-associated genes had the highest proportion of lineage-defining mutations (Fig. 4d-h-l and Supplementary Fig. 12).

Tracking lineages in real-time. *Phylowave* enables us to track population composition changes through time, with a direct link to fitness. As our method relies on the estimation of the pairwise distance distribution for each node in a tree, the number of sequences does not impact the index dynamics, as long as sequences are representative of the diversity (Fig. 5a). To demonstrate this robustness to sampling biases in time, we conducted a sensitivity analysis using the SARS-CoV-2 dataset by repeatedly removing a subset of genomes, including in a temporally uneven manner, and re-estimated the circulating lineages each time. We were still able to detect virtually all the lineages, even when using heavily biassed datasets (Fig. 5b, mean ARI of 0.90). As with other phylodynamic methods, we note that *phylowave* is sensitive to biases in the source of sequence where the sequenced pathogens are not representative of the diversity of the underlying population. Finally, we explored how quickly *phylowave* was able to detect newly emerging lineages. We truncated our full global SARS-CoV-2 dataset every two weeks and reran the detection algorithm. We found that our model was able to capture each lineage, with a median delay of 2.2 months after emergence, with only 10 sequences required (Fig. 5c). Considering that the SARS-CoV-2 dataset used in this study comes from NextStrain and was composed of only 3129 sequences (approximately 0.02% of all sequences available on GISAID at the time of the study), the time to lineage identification could be further shortened with larger datasets.

Conclusion. In this study, we presented a novel approach that can agnostically track changes in population composition in phylogenetic trees, even in situations of heavily biassed availability of sequences. Across a broad range of pathogens, we have shown we can recover the main known circulating lineages for each pathogen, as well as identify new, previously unknown lineages, with significant changes in fitness. We can quantify the relative fitness of each lineage and identify genetic changes linked to the emergence of new, fitter lineages. This approach can have important implications for public health surveillance. There is increased interest in the systematic sequencing of pathogens detected in healthcare settings. By integrating such sequencing efforts into *phylowave*, public health agencies will be able to identify emergent strains in a timely manner, which can be used to promote targeted interventions. *phylowave* is also able to make fundamental insights into pathogen ecology. By quantifying the relative fitness advantage of new strains, *phylowave* can help us identify potential drivers of emergence, including the role of population immunity from natural infection or vaccination. Finally, by identifying the specific genomic changes linked to fitness changes, this work provides testable biological hypotheses about genetic variants in each pathogen that are driving the changes in population fitness of that pathogen.

References

1. Luksza, M. & Lässig, M. A predictive fitness model for influenza. *Nature* **507**, 57–61 (2014).
2. Meijers, M., Ruchnewitz, D., Eberhardt, J., Łuksza, M. & Lässig, M. Population immunity predicts evolutionary trajectories of SARS-CoV-2. *Cell* (2023) doi:10.1016/j.cell.2023.09.022.
3. Lefrancq, N. *et al.* Global spatial dynamics and vaccine-induced fitness changes of Bordetella pertussis. *Sci. Transl. Med.* **14**, eabn3253 (2022).
4. Rambaut, A. *et al.* A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology. *Nat Microbiol* **5**, 1403–1407 (2020).
5. Aksamentov, I., Roemer, C., Hodcroft, E. & Neher, R. Nextclade: clade assignment, mutation

calling and quality control for viral genomes. *J. Open Source Softw.* **6**, 3773 (2021).
6. Influenza virus characterization - Summary Europe, December 2022. *European Centre for Disease Prevention and Control* https://www.ecdc.europa.eu/en/publications-data/influenza-virus-characterization-summary-europe-december-2022 (2023).
7. Bart, M. J. *et al.* Global population structure and evolution of Bordetella pertussis and their relationship with vaccination. *MBio* **5**, e01074 (2014).
8. Obermeyer, F. *et al.* Analysis of 6.4 million SARS-CoV-2 genomes identifies mutations associated with fitness. *Science* **376**, 1327–1332 (2022).
9. Belman, S. *et al.* Geographic migration and vaccine-induced fitness changes of Streptococcus pneumoniae. *bioRxiv* 2023.01.18.524577 (2023) doi:10.1101/2023.01.18.524577.
10. Neher, R. A., Russell, C. A. & Shraiman, B. I. Predicting evolution from the shape of genealogical trees. *Elife* **3**, (2014).
11. Stadler, T. & Bonhoeffer, S. Uncovering epidemiological dynamics in heterogeneous host populations using phylogenetic methods. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **368**, 20120198 (2013).
12. Kepler, L., Hamins-Puertolas, M. & Rasmussen, D. A. Decomposing the sources of SARS-CoV-2 fitness variation in the United States. *Virus Evol* **7**, veab073 (2021).
13. Barido-Sottani, J., Vaughan, T. G. & Stadler, T. A Multitype Birth-Death Model for Bayesian Inference of Lineage-Specific Birth and Death Rates. *Syst. Biol.* **69**, 973–986 (2020).
14. Tonkin-Hill, G., Lees, J. A., Bentley, S. D., Frost, S. D. W. & Corander, J. Fast hierarchical Bayesian analysis of population structure. *Nucleic Acids Res.* **47**, 5539–5549 (2019).
15. Volz, E. M. *et al.* Identification of Hidden Population Structure in Time-Scaled Phylogenies. *Syst. Biol.* **69**, 884–896 (2020).
16. Wirth, T., Wong, V., Vandenesch, F. & Rasigade, J.-P. Applied phyloepidemiology: Detecting drivers of pathogen transmission from genomic signatures using density measures. *Evol. Appl.* **13**, 1513–1525 (2020).
17. Kingman, J. F. C. On the Genealogy of Large Populations. *J. Appl. Probab.* **19**, 27–43 (1982).
18. Griffiths, R. C. & Tavaré, S. Sampling theory for neutral alleles in a varying environment. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **344**, 403–410 (1994).
19. Austerlitz, F., Jung-Muller, B., Godelle, B. & Gouyon, P.-H. Evolution of Coalescence Times, Genetic Diversity and Structure during Colonization. *Theor. Popul. Biol.* **51**, 148–164 (1997).
20. Casali, N. *et al.* Evolution and transmission of drug-resistant tuberculosis in a Russian population. *Nat. Genet.* **46**, 279–286 (2014).
21. Hubert, L. & Arabie, P. Comparing partitions. *J. Classification* **2**, 193–218 (1985).
22. Sanyaolu, A. *et al.* The emerging SARS-CoV-2 variants of concern. *Ther Adv Infect Dis* **8**, 20499361211024372 (2021).
23. Viana, R. *et al.* Rapid epidemic expansion of the SARS-CoV-2 Omicron variant in southern Africa. *Nature* **603**, 679–686 (2022).
24. Baker, L., Brown, T., Maiden, M. C. & Drobniewski, F. Silent nucleotide polymorphisms and a phylogeny for Mycobacterium tuberculosis. *Emerg. Infect. Dis.* **10**, 1568–1577 (2004).
25. Gagneux, S. *et al.* Variable host-pathogen compatibility in Mycobacterium tuberculosis. *Proc. Natl. Acad. Sci. U. S. A.* **103**, 2869–2873 (2006).
26. Olawoye, I. B. *et al.* Emergence and spread of two SARS-CoV-2 variants of interest in Nigeria. *Nat. Commun.* **14**, 811 (2023).
27. Laiton-Donato, K. *et al.* Characterization of the emerging B.1.621 variant of interest of SARS-

CoV-2. *Infect. Genet. Evol.* **95**, 105038 (2021).
28. Hodcroft, E. B. *et al.* Spread of a SARS-CoV-2 variant through Europe in the summer of 2020. *Nature* **595**, 707–712 (2021).
29. Russell, C. A. *et al.* The global circulation of seasonal influenza A (H3N2) viruses. *Science* **320**, 340–346 (2008).
30. Petrova, V. N. & Russell, C. A. The evolution of seasonal influenza viruses. *Nat. Rev. Microbiol.* **16**, 47–60 (2018).
31. Bouchez, V. *et al.* Evolution of Bordetella pertussis over a 23-year period in France, 1996 to 2018. *Euro Surveill.* **26**, (2021).
32. Wiley, D. C., Wilson, I. A. & Skehel, J. J. Structural identification of the antibody-binding sites of Hong Kong influenza haemagglutinin and their involvement in antigenic variation. *Nature* **289**, 373–378 (1981).
33. Koel, B. F. *et al.* Substitutions near the receptor binding site determine major antigenic change during influenza virus evolution. *Science* **342**, 976–979 (2013).
34. Coutte, L., Antoine, R., Drobecq, H., Locht, C. & Jacob-Dubuisson, F. Subtilisin-like autotransporter serves as maturation protease in a bacterial secretion pathway. *EMBO J.* **20**, 5040–5048 (2001).
35. Parkhill, J. *et al.* Comparative analysis of the genome sequences of Bordetella pertussis, Bordetella parapertussis and Bordetella bronchiseptica. *Nat. Genet.* **35**, 32–40 (2003).
36. Chitale, P. *et al.* A comprehensive update to the Mycobacterium tuberculosis H37Rv reference genome. *Nat. Commun.* **13**, 7068 (2022).

Figures

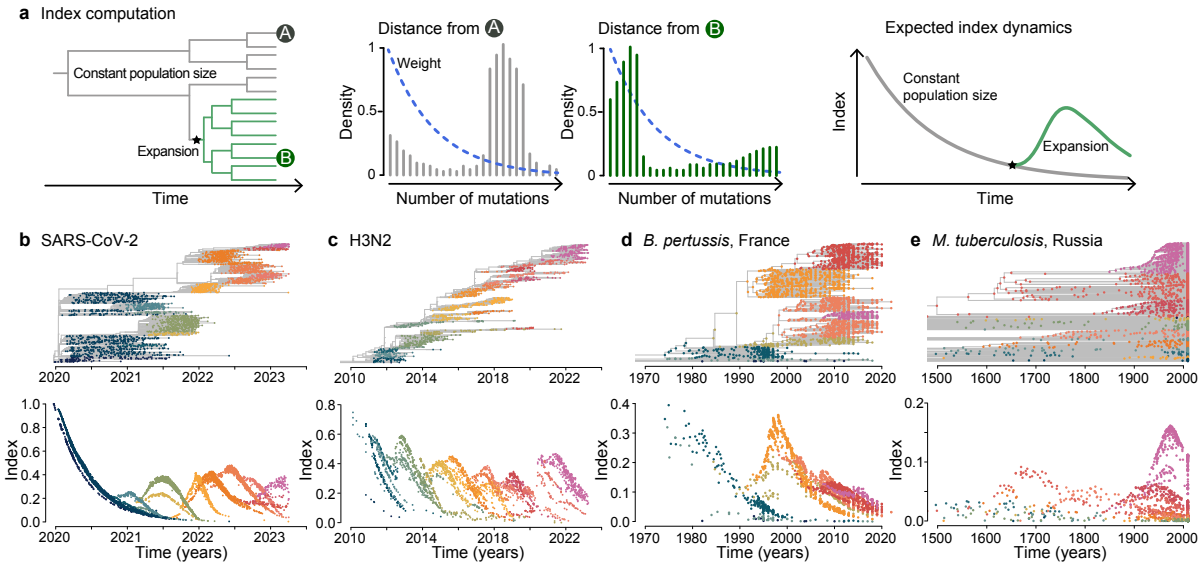


Figure 1: Tracking changes in population composition by following index dynamics.

(a) Schematics describing the principles of index computation. From left to right: example of a time-resolved phylogenetic tree with a background population (grey) and an emerging lineage (green); pairwise distance distribution from terminal node A, or terminal node B, respectively, to the rest of the population, with the dashed blue line denoting the geometric weighting; and expected index dynamics over time. See methods for details. **(b-e)** For each pathogen, SARS-CoV-2 (c), H3N2 (c), *B. pertussis* (d) and *M. tuberculosis* (e), we present the index dynamics computed at each node (terminal or internal). Colours represent the different lineages identified by their different index dynamics (Extended Data Fig. 3). Dynamics coloured by known lineages are presented in Extended Data Fig. 2.

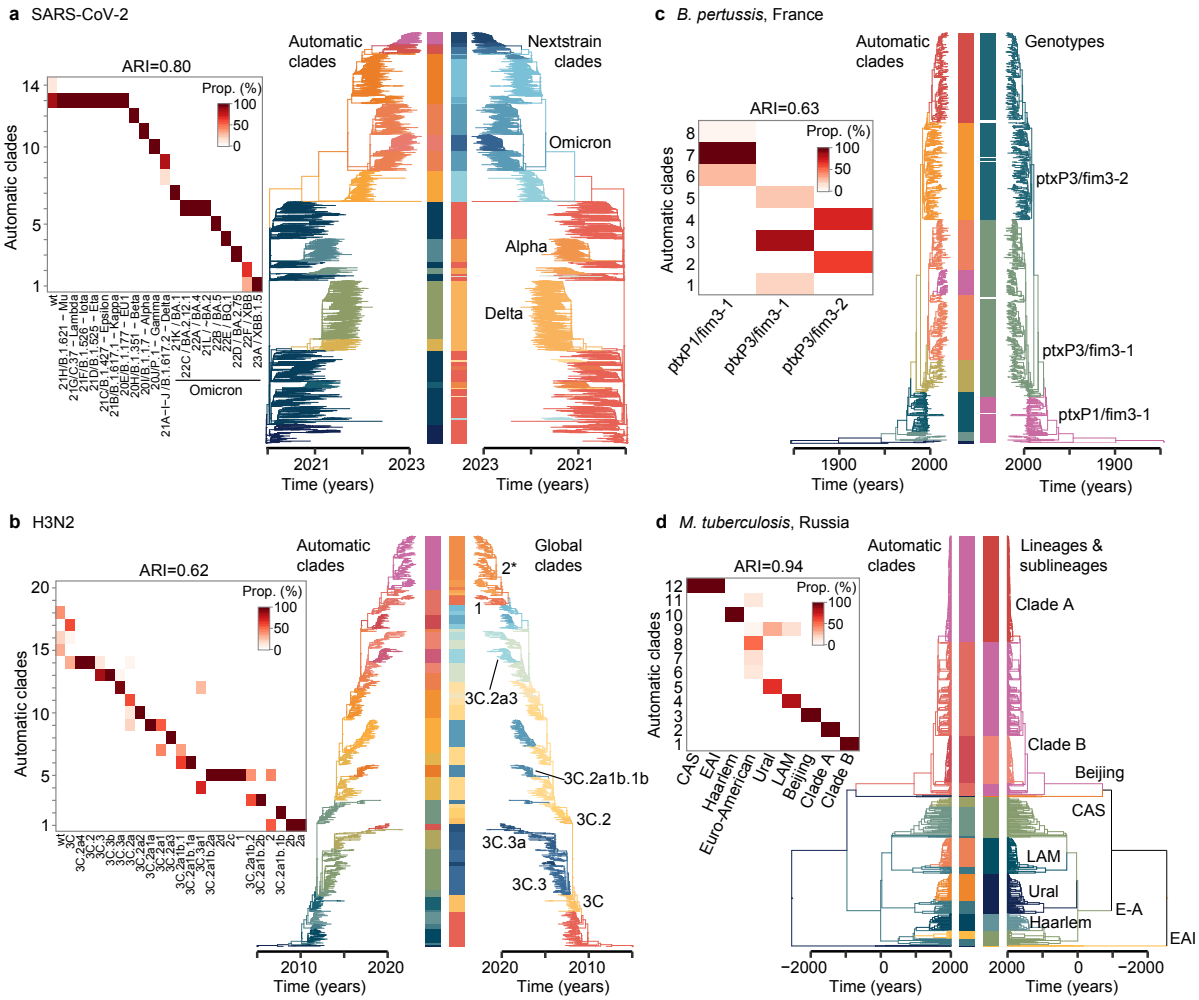


Figure 2: Comparison of the identified lineages to the known population composition.

We present time-resolved phylogenetic trees and heatmaps for SARS-CoV-2 **(a)**, H3N2 **(b)**, *B. pertussis* **(c)** and *M. tuberculosis* **(d)** to compare the identified lineages to the automatic clades found by *phylowave*. Darker colours in the heatmaps represent more agreement between both classifications. Contingency tables are presented in Supplementary Tables S2-5. The heatmaps are presented in large in Extended Data Fig. 5. Timed-resolved phylogenetic trees are coloured by respective lineage classifications: automatic clades on the left, and previously identified lineages on the right. The colours of the automatic clades are the same as in Fig. 1. For *M. tuberculosis*, LAM denotes the Latin American-Mediterranean lineage, E-A the Euro-American lineage, EAI the East African Indian lineage and CAS the Central Asian Strain lineage.

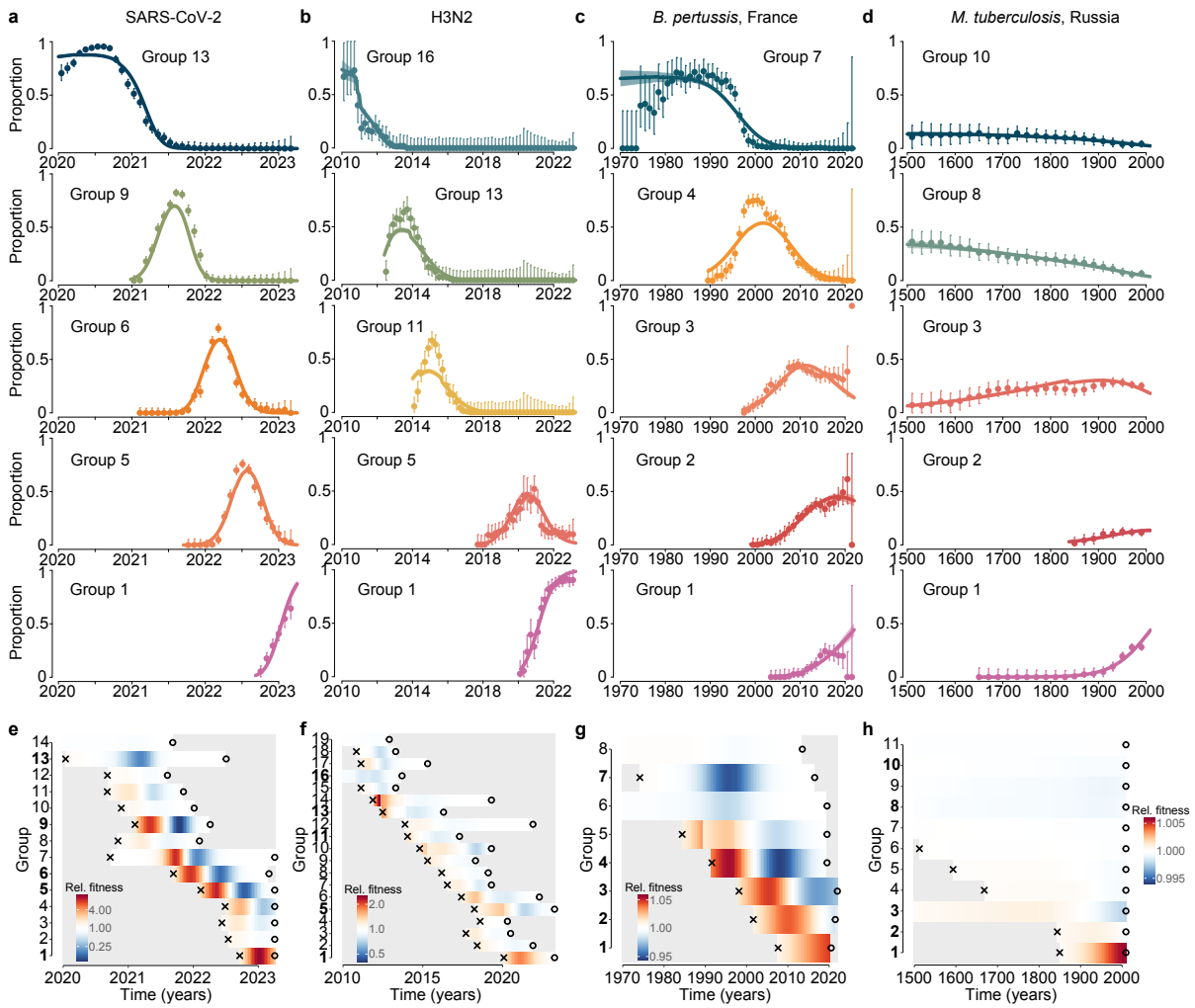


Figure 3: Estimation of the fitness of each lineage.

(a-d) Model fits per pathogen: SARS-CoV-2 (a), H3N2 (b), *B. pertussis* (c) and *M. tuberculosis* (d). For each pathogen, we present the fits for the five most prevalent groups. The fits for all groups are presented in Supplementary Fig. 4-7. Coloured dots represent data, bars denote 95% confidence intervals. Coloured lines and shaded areas represent the median and 95% credible interval of the posterior. **(e-h)** Relative fitness of each group, over time. Estimates for all groups are presented in Extended Data Fig. 7. Crosses indicate the group's MRCA. Open circles indicate the last isolate from each group, in our datasets.

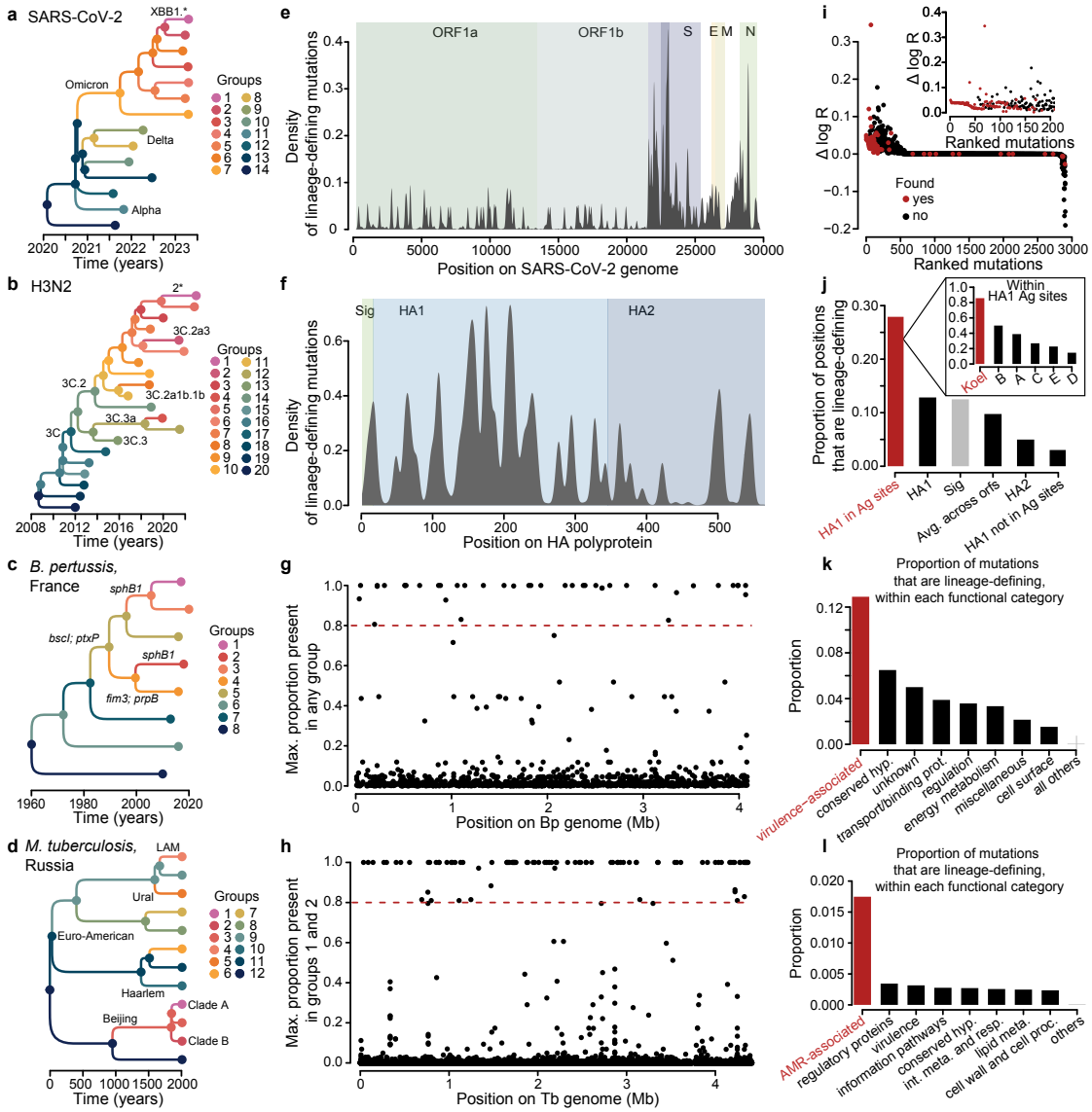


Figure 4: Lineage-defining genetic mutations.

For each pathogen, we present a summary of the genetic evolution of the lineages. **(a-d)** For each pathogen, we present the lineage trees representing the genealogical relationship between them. Key clades are highlighted. Colours indicate groups. **(e-h)** Lineage-defining mutations along the genome of each pathogen considered. For SARS-CoV-2 (e) and H3N2 (f) viruses we plot the density of lineage-defining mutations along the full genome (SARS-CoV-2) or HA polyprotein (H3N2). Colours indicate the main ORFs. For *B. pertussis* (g) and *M. tuberculosis* (h) we plot for each mutation the maximum proportion of that mutation that is present in any group (*B. pertussis*) or in groups 1 and 2 (*M. tuberculosis*). The dashed lines represent the 0.8 cutoff. **(i-l)** Functional relevance of the mutations identified. (i) For SARS-CoV-2, we compare the substitutions analysed by Obermeyer and colleagues[8] and the mutations found to be lineage-defining in our study. For each amino acid change, the figure presents the estimated increased of fitness Δ log R (y axis), as a function of its rank, inferred by statistical significance (x axis). Mutations in red are found by our method. (j) For H3N2, we plot the proportion of positions that are lineage-defining within each HA polyprotein subunit, and antigenic sites[32,33](insert). (k) For *B. pertussis*, we plot the proportion of mutations that are lineage-defining within each functional category[35] (l) Same as K, for *M. tuberculosis*[36]. The lists of lineage-defining mutations for each pathogen can be found in Supplementary Tables 6-9.

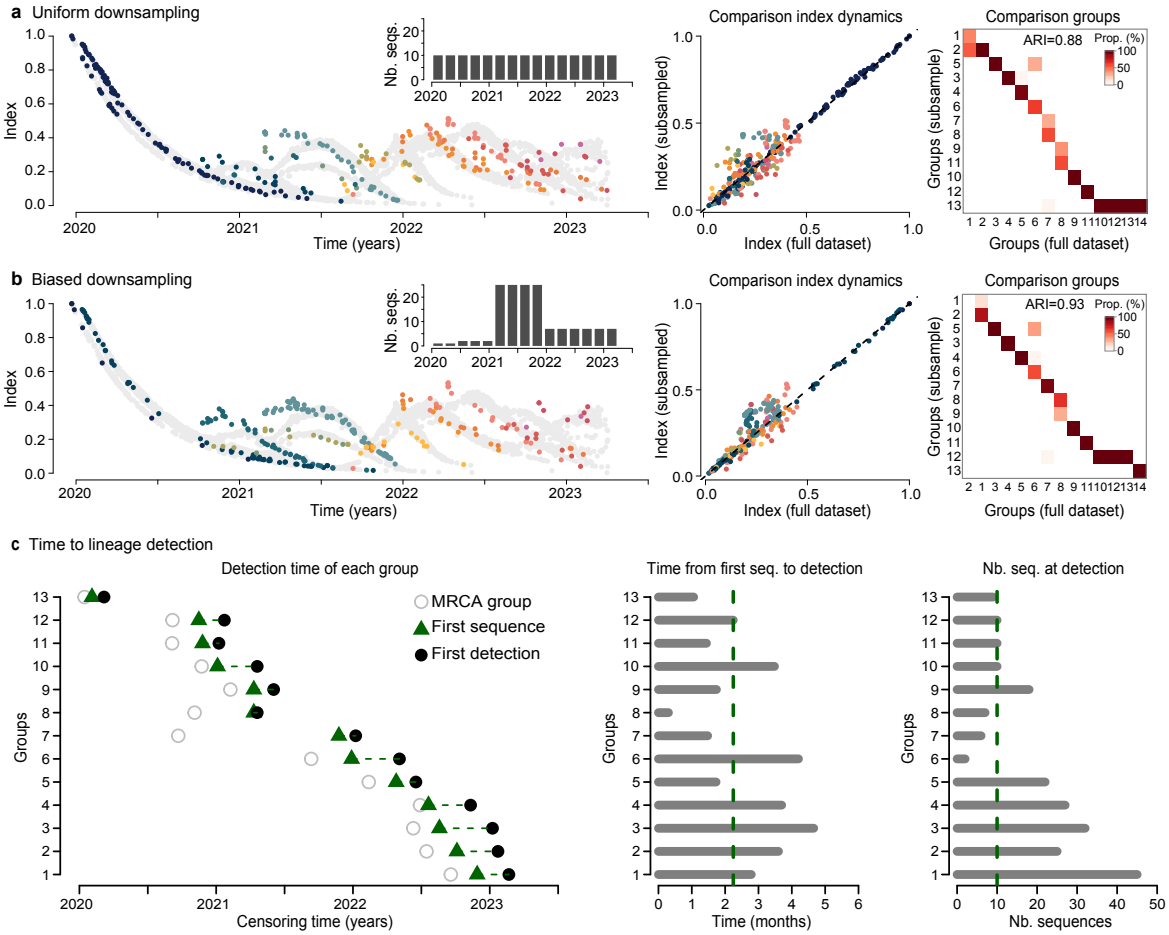


Figure 5: Robustness of *phylowave* to sampling intensities and time to lineage detection.
(a-b) Robustness to downsampling. We kept only 150 sequences from the global SARS-CoV-2 tree, either sampled uniformly through time (a) or in a temporally uneven manner (b). From left to right: Index dynamics computed on the subsampled trees, coloured by detected lineages, with temporal distribution of sequences in inserts; pairwise comparison of the index computed at nodes (internal and terminal) in the trees from the full dataset (x-axis) and subsampled datasets (y-axis); heatmap comparing the automatic clades found by *phylowave* on the full dataset (x-axis) to the automatic clades found on the subsampled datasets (y-axis). Darker colours on the heatmap denote more agreement between both namings. **(c)** Time to lineage detection. The full global SARS-CoV-2 dataset was censored every two weeks and reran the detection algorithm. From left to right: detection time of each group, with open circles denoting the group's MRCA in our tree, the green triangles denoting the first sequence of the group in our dataset, and the black dots denoting the first detection of the group by *phylowave*; time from first sequence isolated in our dataset to group detection; number of sequences within each group at the time of detection. The dashed lines denote the median time to detection, or number of sequences at detection, respectively.

Methods

Sequence data

For each pathogen, we compiled a dataset to investigate the changes in the population composition. For SARS-CoV-2 and Influenza H3N2, we extracted the datasets from the publicly available NextStrain[37] timed-resolved phylogenies accessed on 14 April 2023. These datasets are sub-samples from all publicly available sequences in GISAID, to represent the diversity as much as possible (we used the 'all-time' dataset for SARS-CoV-2 and the '12y' one for H3N2). In all, we have 3129 whole genome SARS-CoV-2 sequences sampled from 26 December 2019 to 3 April 2023, and 1476 Influenza H3N2 Hemagglutinin (HA) sequences from 1 January 2005 to 3 April 2023 (Supplementary Tables 10-11). For *B. pertussis*, we used 1248 sequences from 1953 to 2022, collected by the National Reference Center (NRC) for Whooping Cough and Other Bordetella Infections in France (Supplementary Table 12). This dataset is composed of 1023 sequences previously published[3,38–41] and 225 newly sequenced isolates. The new isolates have been sequenced with the same methods as previously described[3]. This dataset is representative of the *B. pertussis* diversity in France as the NRC is receiving isolates from 42 sentinel hospitals throughout France. For *M. tuberculosis*, we used 997 previously published sequences, isolated in 2008-2010 in Samara, Russia[20]. This dataset is also representative of *M. tuberculosis* sequence diversity at that location as isolates were prospectively collected from individual patients living in the region and representative of the entire population (Supplementary Table 13).

Multi-sequence alignment for each pathogen

We compiled alignments of all sequences being used. For SARS-CoV-2, we used the precomputed multi-sequence alignment provided by GISAID. For H3N2, we aligned all HA sequences using MAFFT[42] (v7.309), with default settings. We then manually checked that the alignment did not have any frameshift and had minimal gaps. For *B. pertussis* and *M. tuberculosis*, we worked from raw reads with a protocol previously described[3]. Reads were trimmed using Cutadapt[43] (v3.4) and quality was checked with FastQC[44] (v0.11.9). Using BWA-MEM[45] (v0.7.17), reads were mapped against the complete Tohama I reference genome (Accession number: NC_002929), or the complete H37Rv reference genome (Accession number: NC_000962.3). Using GATK[46] (v4.2.0.0), we kept variants that were present in at least 75% of reads, with a Phred quality score higher than 30, a minimum read depth of 5, a minimum mapping quality of 20 and a String Odd Ratio of less than 3. We masked all positions that were covered by less than 5 reads. Further, we filtered out regions which are notoriously difficult to map and/or sequence, similarly to previous studies[3,47]. Namely, for *B. pertussis* we filtered out repeated regions (IS481, IS1002 and IS1663)[35], and phage regions using PHASTER[48]; for *M. tuberculosis*, we filtered out the functional categories “PE/PPE” or “insertion sequences and phages''[47]. For *B. pertussis*, we also checked for recombination in our alignment using Gubbins[49] (v3.3.0). As a result, we obtained an alignment of 4701 SNPs for *B. pertussis* and 30533 SNPs for *M. tuberculosis*.

Reconstruction of timed-resolved phylogenies

For each pathogen, we obtained timed-resolved phylogenies. For SARS-CoV-2 and H3N2, we used the NextStrain trees, accessed on 14 April 2023 [37]. For *B. pertussis* and *M. tuberculosis*, we reconstructed the timed phylogenies specifically for this study, using the SNP-based alignments. We first built maximum-likelihood trees using IQ-tree[50] (v2.1.0), using a GTR+F+G substitution model. To assess the branch support, we used the ultrafast bootstrap approximation provided in IQ-tree, performing 1000

replicates for each dataset with the bnni option to reduce the risk of overestimating the branch support[51].

For *B. pertussis*, the time-tree was reconstructed using BEAST v1.10.4[52], under a GTR substitution model[18] accounting for the number of constant sites, a relaxed lognormal clock model[53] and a skygrid population size model[54]. Three independent Markov chains were run for 150 000 000 generations each, with parameter values sampled every 10,000 generations. Runs were optimised using the GPU BEAGLE library[55] (v4.0.0). Chains were manually checked for convergence (ESS values > 200) using the Tracer software[56] (v1.7.2). We manually removed a 10% burn-in.

For *M. tuberculosis*, as all sequences were isolated in 2008-2010, we could not infer a clock rate, but instead, we used a previously estimated clock rate[57] of 4.6 x 10^{-8} mutations/site/year. We used the software BactDating[58] (v1.1) to perform a bayesian reconstruction of the timed-tree. We used a fixed mean substitution rate, a relaxed clock rate and a constant effective population size. We ran the chain for 10,000,000 iterations and checked for convergence (ESS values > 200).

***Index* definition**

We developed an analytical approach that summarises the changes in population composition in phylogenetic trees at every time point. Our approach builds on a genetic distance-based index, the Timed Haplotype Density (THD)[16], that measures the epidemic success of individual sequences in a dataset. This measure is based on the expectation that sequences sampled from an emerging, fitter, lineage will be phylogenetically closer than the rest of the population at that time, as they will all share the same recent ancestor. We extend this method to track population changes in phylogenetic trees through time.

We define the *Index* of each isolate i in its population at time t as:

$$Index(i) = \sum_{d=0}^{\infty} D_i(d,t) \cdot b^d$$

[Eq. 1]

With $D_i(d,t)$ the distance distribution - in number of mutations or evolutionary time (branch length) - from the isolate i to the rest of the population (internal and terminal nodes) at that time t (Fig. 1) and b^d, the kernel setting the weight of each distance d. b is the bandwidth, $b \in]0,1[$, which is a parameter to set, linked to the timescale. We compute this index on each node in a tree (internal and terminal).

The weight allows us to track lineage emergence dynamically, focusing on short distances between nodes (containing information about recent population dynamics) rather than long distances (containing information about past evolution). The kernel is governed by the bandwidth b, which is a parameter to set. As b is dimensionless, it is hard to set. Instead, we use the notion of *timescale* 50 to choose it: the TMRCA such that pairs of isolates with shorter TMRCAs account for 50% of the kernel density[16]. This timescale is tailored to the specific pathogen studied and its choice will depend on the molecular signal, as well as the transmission rate. Here we used timescales ranging from 1.8 months (SARS-CoV-2, RNA virus) to 30 years (*Mycobacterium tuberculosis,* bacteria) (Table S1).

Our approach provides a quantitative index value to each node (internal and terminal) in the tree, independently of any lineage classification, which is main advantage compared to other methods that rely on lineage classification. This enables to agnostically summarise the changes in population composition in phylogenetic trees at every time point.

Our definition is virtually the same as the one used by Wirth and colleagues[16], with two critical differences: instead of computing the index by summing on each isolate in the population we now sum over the pairwise distance distribution, and we consider the collection time of each sequence to only compute the distance from i to the rest of the population that is circulating at that time.
This index is similar to the Local Branching Index (LBI)[10], which is defined as total surrounding tree length exponentially discounted with increasing distance from the isolate i. In our case, rather than considering the tree length, we compute the distance between nodes.

This index definition enables us to write an expectation of the index dynamics over time, as theoretical pairwise distance distributions can be approximated for different populations. In practice, to compute the index of each node in a phylogenetic tree, we sum over the distance to all nodes in the population, rather than the distance distribution (see section "Index computation on timed-tree with sequences sampled through time").

Linking the Index dynamics to population history.
The pairwise distance distribution $D_i(d,t)$, or more generally $D(d,t)$, can be seen as the probability, $P_c(s = \frac{d}{\mu l}, t)$, for any pair of sequences sampled at time t, to coalesce some time $s = \frac{d}{\mu l}$ in the past, with μ being the rate at which the pathogen accumulates mutations per site and per unit of time, and l the length of its genome.

$$D(d,t) = P_c(s = \frac{d}{\mu l}, t)$$

Therefore, at any time point, writing the probability of coalescing in the past enables us to compute the index in the population. We can update equation 1:

$$Index(t) = \int_0^{\mu l t} P_c(\frac{u}{\mu l}, t) \cdot b^u \; du$$

[Eq. 2]

We note that at time t, the maximum number of mutations accumulated is equal to $\mu l t$. For simplicity, we assume a linear accumulation of mutations through time in all the analytical expressions, though one could consider that mutations accumulate randomly given a Poisson distribution with rate $1/(\mu l t)$.

This probability $P_c(s = \frac{d}{\mu l}, t)$ is closely linked to the effective population size. In Supplementary Fig. 13, we show conceptually how, for different effective population sizes, the probability of coalescing changes, and how it impacts the index dynamics. Formal derivations are presented below in the supplementary text.

Index computation on timed-tree with sequences sampled through time
We use equation 1 to compute the index of each node (internal or terminal) in a timed-phylogenetic tree. To do this, for each node i, we compute its distance to all the other nodes present in the tree at

that time (see Supplementary Fig. 14 for notations). All the nodes that fall within the interval of time $[t_i - t_{wind};\ t_i + t_{wind}]$ are considered to be circulating at the same time as i; with t_i being the collection time of the node i, and t_{wind} the predefined time window width that is tailored to each pathogen. We also consider extant branches in the computation, as they are an evidence of past circulation.

For computation efficiency, similarly to Wirth and colleagues[16], we then compute:

$$Index(i) = \sum_{j \in nodes} I(t_j > t_i - t_{wind} \,\&\, t_j < t_i + t_{wind})\, d(i,j)\, b^{d(i,j)}$$

[Eq. 3]

Where *nodes* is the set of all nodes in the tree, and I is an indicator function.
This computation is efficient as it only requires i) the precomputation of the indicator function, ii) the precomputation of the distance matrix and iii) a matrix multiplication.

For the pathogens presented in our study we used:

- SARS-CoV-2: a timescale of *0.15* $years$, and a window of time $t_{wind} = 15\ days$
- H3N2: a timescale of *0.4* $years$, and a window of time $t_{wind} = 0.25\ years$
- *B. pertussis*: a timescale of *2* $years$, and a window of time $t_{wind} = 1\ years$
- *M. tuberculosis*: a timescale of *30* $years$, and a window of time $t_{wind} = 15\ years$

We illustrate the impact of the timescale on the index dynamics in Extended Data Fig. 8 on the global SARS-CoV-2 tree.

To test the robustness of the index computation to the exact tree topology, we ran a sensitivity analysis on 3000 trees sampled from the posterior of the BEAST run of *Bordetella pertussis*. We chose *Bordetella pertussis* for this analysis as it is the only pathogen for which we have a posterior distribution of trees. We repeatedly computed the index on each tree sampled from the posterior and computed the average index of each tip. While there is uncertainty in the exact value of the index for each tip, we found that the index dynamics of each lineage remained very consistent across the posterior of trees (Supplementary Fig. 15).

Agnostic detection of lineages

We develop an approach that is able to find the set of lineages in the tree that best explains the index dynamics. To do this, we build an algorithm based on generalised additive models (gam) that jointly uses the phylogenetic relationships between nodes in the tree and their index.

In this section, for modelling purposes, we define lineages as monophyletic clades formed by one internal node and all its descendants. Here, these lineages can overlap, meaning that some isolates can be included in multiple lineages. We assume the tree to be binary. For a rooted binary tree with n terminal nodes, there are $n - \boldsymbol{2}$ internal nodes that are not the root, and therefore $n - \boldsymbol{2}$ lineages possibilities, which is substantial. To keep the algorithm tractable, we limit the potential list of lineages to those starting with a internal node that has at least N_{off} offspring, which is chosen. We note the set of internal nodes to test Π. Further, to increase the accuracy of the detection, we only take into account internal nodes that have predefined characteristics:

- For *B. pertussis* and *M. tuberculosis,* as we constructed the bootstrap support of each node (see above), we only consider internal nodes that have a bootstrap support of at least 50% to be the potential start of lineages. This threshold is low, but effectively removes nodes that are not well supported.
- For SARS-CoV-2 and H3N2, instead of bootstrap support, we consider a minimum number of mutations. We only consider internal nodes that have a least 1 mutation on their directly upstream branch.

The algorithm models the log index through time. We use a log transformation to avoid having to restrict to model to positive values, but also to make sure the model does not overfit the index peaks.

The log index of each lineage l is modelled using a cubic spline $S_l(t, k)$ with a pre-defined number of knots k. This allows us to model the log index of each node i, sampled at time $\boldsymbol{t_i}$, given the lineage that it belongs to:

$$log(Index_i) \sim \beta_0 + S_0(t_i, k) + \sum_{l=1}^{L} I(i \in l)\, S_l(t_i, k)$$

Where β_0 is the intercept, $\boldsymbol{L}$ is the total number of lineages, $S_0(t, k)$ and $S_l(t, k)$ are penalised cubic regression splines with k knots[59]. One 'null' spline $S_0(t, k)$ is estimated to model the initial population, together with one spline for each of the $\boldsymbol{L}$ lineages. If $\boldsymbol{L = 0}$, then no $S_l(t, k)$ is estimated. $I()$ is the identity function.

Briefly, the algorithm runs as follows. We start by a null model M_0 that fits the index dynamics with one spline $S_0(t, k)$ (i.e. unstructured population with one single index dynamic, $L = 0$). We store the deviance explained Dev_0 by the model M_0. We then sequentially consider models with increasing complexity M_L: we start by first trying models with one lineage, $L = 1$. We go through the list of internal nodes Π that could be the start of a new lineage. When the deviance explained Dev_1 by the best model M_1 is increased compared to the one of previous null model Dev_0, we keep the lineage (effectively the node from Π) that explains best the dynamics. We then continue this procedure for increasing L. For each number L, we go through the list of internal nodes Π that could be the start of a new lineage. When the deviance explained Dev_L by the model M_L is increased compared to the one of previous model Dev_{L-1}, we keep the lineage (effectively the node from Π) that explains best the dynamics.

The algorithm is implemented in R v4.1.2, using the package mgcv v1.8[60] to implement the gam models.

As for any clustering algorithm, choosing the best number of lineages that describe the index dynamics is a challenging question. We took the approach of the elbow plot. We plot the deviances Dev_L explained by each best model M_L, as a function of the number of lineages $\boldsymbol{L}$. This approach enables us to see how well all the models are performing, and to choose the number $\boldsymbol{L}$ of lineages at which the deviance explained does not increase substantially anymore (Extended Data Fig. 9). From this selected best number of lineages $\boldsymbol{L_{best}}$, we then compute the equivalent set of non-overlapping lineages presented in this paper (Fig. 1-5 and Extended Data Fig. 4). We make sure the minimum number of nodes per non-overlapping lineage is at least N_{min} by merging the small lineages to its closest phylogenetically.

In simulations, we demonstrate a clear elbow that precisely identifies the optimal number of discrete lineages in the dataset (Supplementary Fig. 16). However, not all pathogens in real-world datasets will give clear 'elbows'. This may be due to insufficient sampling intensity and the presence of lineages with only very small differences in fitness. In practice, increasing the number of distinct lineages will progressively lead to the identification of lineages with increasingly reduced fitness differences, with increasing risks of falsely dividing lineages into subpopulations where no true difference exists.

For the pathogens presented in our study we found: SARS-CoV-2: 14 lineages; H3N2: 20 lineages; *B. pertussis*: 8 lineages; and *M. tuberculosis*: 12 lineages.

To compare the automatic lineages found by *phylowave* to those previously identified, we compute a contingency matrix C. Let U be the partition of the isolates by *phylowave*, and V the partition based on literature. Each element $C_{i,j}$ is the number of isolates in both clusters u_i and v_j. In Fig. 2 we plot the this matrix as a heatmap, normalised by column j. We computed the Adjusted Rand-Index (ARI) to measure the agreement between partitions, accounting for random clustering[21]. A value of 1 corresponds to perfect agreement with previously identified lineages, whereas a value of 0 would be expected if clusters were assigned at random.

We illustrate the impact of the timescale on the lineage detection in Extended Data Fig. 8 on the global SARS-CoV-2 tree.

Quantifying the fitness of each lineage

We developed a multinomial logistic model that takes into account the birth of lineages to fit the proportion of each lineage through time and quantify their fitness.

The proportion $p_{\bullet,t}$ of sequences at time t from each lineage is computed as the number of nodes (internal and terminal) divided by the total number of nodes (internal and terminal) in the population at that time. This proportion $p_{\bullet,t}$ is modelled by:

$$p_{\bullet,t} = softmax(\, log(\alpha_\bullet) + \beta_\bullet t)$$

With $\alpha_\bullet$ being the vector of intercept, denoting the initial relative prevalence of each lineage in the population and $\beta_\bullet$ the vector of relative growth rates of each lineage. We assume each lineage i has a

constant relative growth rate β_i in the population, i.e. each lineage has a constant relative fitness through time. We compute all the relative growth rates with reference to the oldest lineage.

We use a Laplace prior for the growth rate coefficient[8]:

$$\beta_\bullet \sim Laplace(0, 1)$$

We take into account lineage birth by only allowing $p_{i,t}$, the lineage i proportion in the population at time t, to be non-negative after the lineage's Most Recent Common Ancestor (MRCA). Formally, this is done by parameterizing $\alpha_\bullet$ as follows. We divide the lineages into two types, either 'ancestral', or 'non-ancestral':

- An 'ancestral' lineage is a lineage that is present at the beginning of the time series considered. The total number of ancestral lineages in noted G. For those lineages, we sample directly their starting proportions with prior:

$$\alpha_i \sim simplex(G); \quad if\ i \in ancestors$$

- A 'non-ancestral' lineage is a lineage that appears after some time - for example the Omicron variant. For those lineages, we assume that their starting frequency, at the time of emergence, is a function of the proportion of their parents in the population at that time. Thus we write:

$$\alpha_i = \gamma_i\, p_{j,t_{MRCA\,i}}; \quad if\ i \notin ancestors$$

Where j is the parent lineage of lineage i, $p_{j,t_{MRCA\,i}}$ is the proportion of the parent lineage j at the time of emergence $t_{MRCA\,i}$ of the offspring lineage i, and γ_i is the share of the parent lineage that is becoming the new lineage. We sample γ_i with a strong prior as we expect that the starting proportion of new lineages should be small:

$$\gamma_i \sim beta(1, 99); \quad if\ i \notin ancestors$$

Finally, we update the parent j proportion as follows:

$$p_{j,t_{MRCA\,i}+\delta} = (1-\gamma_i)\, p_{j,t_{MRCA\,i}}$$

While this parameterization is more complex than the previous efforts using a similar model[8], it enables us to take into account that lineages appear through time, which makes the model more biologically relevant (e.g., by not estimating the proportion of Omicron in the population in 2020). We chose to parameterize the starting proportions of the new lineages as a function of their parent's proportions so that i) the model is biologically sound, i.e. the starting proportion of a new lineage cannot be greater than the one of its parent, and ii) the starting proportions are constrained by the proportion of their parents, which makes is statistically easier to fit.

We use a multinomial likelihood to fit the count of sequences per lineage through time $y_{\bullet,t}$:

$$y_{\bullet,t} \sim multinomial(\sum_i y_{i,t}\ , p_{\bullet,t})$$

We further computed the inferred real-time growth rate (i.e. fitness) $r_i\,(t)$ of each lineage i in the population (Fig. 3e-h), to control for the varying presence of all circulating lineages through time. Indeed, while our model estimates a constant fitness parameter for each lineage, their actual fitness through time depends on what other lineages are circulating at that time.

$$r_i\,(t) = p_{i,t} \sum_{\substack{j \in \text{lineages}, \\ j \neq i}} p_{j,t}\,(\beta_i - \beta_j)$$

These results are more useful compared to the usual presentation of the parameters, which by default display the relative fitness compared to the ancestral lineage, in this case 19A (the lineage that includes the first SARS-CoV-2 sequences isolated in Wuhan, China).

The model was implemented in Stan, using the cmdstanr package[61]. We ran this model on 3 independent chains with 1,000 iterations and 50% burn-in for each pathogen. We used 2.5 and 97.5 quantiles from the resulting posterior distributions for 95% credible intervals of the parameters.

We fit the counts per lineage in windows of 1 month for SARS-CoV-2, 0.2 year for H3N2, 1 year for *B. pertussis* and 20 years for *M. tuberculosis*, with t counted in years for all pathogens.

Defining mutations of each lineage

We explored whether specific changes in the genomes were linked to lineage fitness by identifying lineage-defining mutations. We defined such mutations as:

- Mutations that are present in more than 80% of the nodes in that lineage
- While those mutations are not present in the set of defining mutations of the ancestral lineage.

For all pathogens, we reconstructed the mutations at each node in the trees using the ancestral state reconstruction implemented in the library ape. To maximise the correct assignment for nodes, we only consider nodes for which the state's probability was >0.9. Mutations were then classified as synonymous, non-synonymous, or extragenic. For *M. tuberculosis* and *B. pertussis* we also classified each mutation by functional category[35,36].

We computed the density of lineage-defining mutations along the SARS-CoV-2 full genome and H3N2 HA polyprotein with a kernel density estimate (Fig. 4e-f). We used a gaussian kernel with a bandwidth of 50 base pairs (bp) for SARS-CoV-2, and a bandwidth of 2.5 amino acid (AA) for H3N2. For *B. pertussis* and *M. tuberculosis* we plot for each mutation the maximum proportion of that mutation that is present in the set of groups considered.

To assess the function relevance of the mutations identified for each pathogen (Supplementary Tables S6-9), we compared them to the literature.

For SARS-CoV-2, we matched the amino acid substitution we found to the ones that Obermeyer and colleagues analysed[8]. The authors analysed 6.4 million genomes up to January 20, 2022 and estimated the fitness effect of 2904 substitutions. Although our global dataset is from an extended period of time (up to 3 April 2023), 83% (N=161) of the lineage-defining mutations were analysed by Obermeyer and colleagues. Our approach was able to recover every single one of the top 55 fittest mutations found by Obermeyer and colleagues. Among the top 100 fittest mutations, our approach recovered 86% of them. The mutations missed by our method are mainly linked to small subclades of variants, and they seem to have spread in those clades only (e.g. in subclades of delta 21I: ORF1a:T3750I and ORF1b:R188Q). One mutation was missed because of a lack of certainty in the ancestral state reconstruction around the root of the Omicron sublineages: S:T376A. From the mutations that were estimated to have no fitness increase, we found 7 (among 2331 analysed by Obermeyer et al.). These mutations include S:D614G and ORF1b:P314L, two mutations that are linked to an early lineage that eventually got fixed in the whole population. We also found ORF8:G8* and S:G252V, defining our lineage 1 (XBB1*), and ORF1a:L3829F, S:N460K and ORF1b:M1156I, defining

our lineage 4 (22E/BQ.1). These mutations were analysed by Obermeyer et al. but were only present in very small frequency in their dataset, as they are mainly linked to the emergence of recent variants, which emerged after their study.

For H3N2, we computed the proportion of positions that are lineage-defining within each HA polyprotein subunit, and antigenic sites[32,33]. A position is lineage-defining if it has at least one AA substitution that is lineage-defining. The proportion is computed as follows:

$$\pi_L = \frac{number\ of\ positions\ that\ are\ lineage-defining\ within\ L}{number\ of\ positions\ that\ are\ mutated\ within\ L}$$

Where L is the set of positions to be analysed (subunits or antigenic sites). We found that the Koel sites[33] had the highest proportion of lineage-defining mutations, with 86% of positions (N=6, out of 7) being recovered by *phylowave*. Specifically, the key positions 156, 159 and 193, defining multiple clades, were recovered. We also recovered positions that defined ancestral lineages, namely positions 145, 158 and 189. Lastly, the position 155 was not picked up by our method as it did not have any variability in our dataset. Indeed, this position was found to be linked to major antigenic changes in the 1960s and 1970s, which does not overlap with our study period (2005-2023).

For the bacteria *B. pertussis* and *M. tuberculosis* we employ a similar metric, by grouping mutations by gene functional categories. We compute:

$$\pi_F = \frac{number\ of\ AA\ substitutions\ that\ are\ lineage-defining\ within\ F}{number\ of\ AA\ substitutions\ within\ F}$$

Where F is the gene functional category considered[35,36]. As a sensitivity analysis, we also replicated this computation on synonymous nucleotide changes, as we expect these mutations to be neutral, and therefore not linked to any particular functional category (Supplementary Fig. 17). We found that, indeed, there was no particular functional category that had significantly more lineage-defining synonymous mutations than others, for both bacteria.

To further check our findings visually, we plotted the lineage-defining mutations for each pathogens next to their phylogenetic trees (Supplementary Fig. 9-12). To make sure the figures were interpretable, we plotted only the mutations in the spike protein for SARS-CoV-2 (Supplementary Fig. 9), the HA1 subunit for H3N2 (Supplementary Fig. 10), and the mutations defining lineages 1 and 2 for *M. tuberculosis* (Supplementary Fig. 12). For *B. pertussis*, we plotted all mutations (AA substitutions and promoter mutations) (Supplementary Fig. 11).

Robustness to sampling strategies

To demonstrate the robustness to sampling biases in time, we conducted a sensitivity analysis using the global SARS-CoV-2 dataset. We selected two random sets of 150 sequences from the 3129 sequences in our full dataset. We selected them either uniformly through time, or in a temporally uneven manner. To do so, we divided the sequences in 15 time windows of equal length (79 days). For the uniform sampling, we included 10 sequences per time bin, randomly selected. For the biassed sampling, we included the following number of sequences per bin (see insert on Fig. 5b):

- windows 1 and 2: 1 sequence per bin;
- windows 3 to 5: 2 sequences per bin;

- windows 6 to 9: 25 sequence per bin;
- windows 10 to 15: 7 sequences per bin.

After selecting the sequences, we pruned from the tree the ones that were not selected. We then performed the same analysis as described above. We also compared the groups found.

Analysis of time to detection

We explored how fast after emergence *phylowave* was able to detect lineages. To do this we truncated our full global SARS-CoV-2 dataset every two weeks. Overall, we obtained 81 datasets. Two examples of the index dynamics on censored data on 2021.26 and 2022.50 are presented in Extended Data Fig. 10. We then re-ran the detection algorithm on each dataset. To obtain the best set of lineages automatically for each dataset, we chose the set at which the log deviance explained did not increase by more than 0.01%.

Method References

37. Hadfield, J. *et al.* Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics* **34**, 4121–4123 (2018).
38. Bouchez, V. *et al.* First report and detailed characterization of B. pertussis isolates not expressing Pertussis Toxin or Pertactin. *Vaccine* **27**, 6034–6041 (2009).
39. Hegerle, N. *et al.* Evolution of French Bordetella pertussis and Bordetella parapertussis isolates: increase of Bordetellae not expressing pertactin. *Clin. Microbiol. Infect.* **18**, E340–6 (2012).
40. Hegerle, N., Dore, G. & Guiso, N. Pertactin deficient Bordetella pertussis present a better fitness in mice immunized with an acellular pertussis vaccine. *Vaccine* **32**, 6597–6600 (2014).
41. Bouchez, V. *et al.* Genomic Sequencing of Bordetella pertussis for Epidemiology and Global Surveillance of Whooping Cough. *Emerg. Infect. Dis.* **24**, 988–994 (2018).
42. Katoh, K., Misawa, K., Kuma, K.-I. & Miyata, T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.* **30**, 3059–3066 (2002).
43. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17**, 10–12 (2011).
44. Andrews, S. & Others. FastQC: a quality control tool for high throughput sequence data. 2010. Preprint at (2017).
45. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv* (2013).
46. McKenna, A. *et al.* The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**, 1297–1303 (2010).
47. Casali, N. *et al.* Microevolution of extensively drug-resistant tuberculosis in Russia. *Genome Res.* **22**, 735–745 (2012).
48. Arndt, D. *et al.* PHASTER: a better, faster version of the PHAST phage search tool. *Nucleic Acids Res.* **44**, W16–21 (2016).
49. Croucher, N. J. *et al.* Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. *Nucleic Acids Res.* **43**, e15 (2015).
50. Nguyen, L.-T., Schmidt, H. A., von Haeseler, A. & Minh, B. Q. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.* **32**, 268–274 (2015).
51. Hoang, D. T., Chernomor, O., von Haeseler, A., Minh, B. Q. & Vinh, L. S. UFBoot2: Improving the Ultrafast Bootstrap Approximation. *Mol. Biol. Evol.* **35**, 518–522 (2018).
52. Suchard, M. A. *et al.* Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evol* **4**, vey016 (2018).
53. Drummond, A. J., Ho, S. Y. W., Phillips, M. J. & Rambaut, A. Relaxed phylogenetics and dating

with confidence. *PLoS Biol.* **4**, e88 (2006).
54. Minin, V. N., Bloomquist, E. W. & Suchard, M. A. Smooth skyride through a rough skyline: Bayesian coalescent-based inference of population dynamics. *Mol. Biol. Evol.* **25**, 1459–1471 (2008).
55. Ayres, D. L. *et al.* BEAGLE 3: Improved Performance, Scaling, and Usability for a High-Performance Computing Library for Statistical Phylogenetics. *Systematic Biology* vol. 68 1052–1061 Preprint at https://doi.org/10.1093/sysbio/syz020 (2019).
56. Rambaut, A., Drummond, A. J., Xie, D., Baele, G. & Suchard, M. A. Posterior Summarization in Bayesian Phylogenetics Using Tracer 1.7. *Syst. Biol.* **67**, 901–904 (2018).
57. Bos, K. I. *et al.* Pre-Columbian mycobacterial genomes reveal seals as a source of New World human tuberculosis. *Nature* **514**, 494–497 (2014).
58. Didelot, X., Croucher, N. J., Bentley, S. D., Harris, S. R. & Wilson, D. J. Bayesian inference of ancestral dates on bacterial phylogenetic trees. *Nucleic Acids Res.* **46**, e134 (2018).
59. Wood, S. N. *Generalized Additive Models: An Introduction with R, Second Edition*. (CRC Press, 2017).
60. Wood, S. & Wood, M. S. Package 'mgcv'. *R package version* **1**, 729 (2015).
61. Gabry, J. & Češnovar, R. cmdstanr: R Interface to'CmdStan'. *: https://mc-stan. org/cmdstanr, https://discourse. mc …* (2021).
62. Lefrancq, N. *noemielefrancq/Phylowave_Learning-Fitness-Dynamics-Pathogens-in-Phylogenies: V1*. (Zenodo, 2024). doi:10.5281/ZENODO.13952222.
63. Stadler, T. Simulating trees with a fixed number of extant species. *Syst. Biol.* **60**, 676–684 (2011).
64. Vaughan, T. G. ReMASTER: improved phylodynamic simulation for BEAST 2.7. *Bioinformatics* **40**, (2024).
65. Ly-Trong, N., Naser-Khdour, S., Lanfear, R. & Minh, B. Q. AliSim: A Fast and Versatile Phylogenetic Sequence Simulator for the Genomic Era. *Mol. Biol. Evol.* **39**, (2022).
66. Volz, E. M., Koelle, K. & Bedford, T. Viral phylodynamics. *PLoS Comput. Biol.* **9**, e1002947 (2013).

Acknowledgements: We thank Caitlin Collins, Megan O'Driscoll, Angkana T. Huang and Trevor Bedford, for discussions and feedback. We thank all the contributors to GISAID for sharing their data. This work was supported financially by the European Research Council (No. 804744 to HS). The National Reference Center for Whooping Cough and Other Bordetella Infections receives support from Institut Pasteur and Public Health France (Santé publique France, Saint Maurice, France).

Author contributions: Conceptualisation: N.L., J.P. and H.S. Method development and modelling analysis: N.L., supported by L.D., J.P. and H.S. Isolate and genomic data collection: N.L., S.B. and V.B. Supervision: J.P. and H.S. Writing – original draft: N.L. Writing – review and editing: N.L., L.D., V.B., S.B., J.P. and H.S. All authors provided input to the manuscript and reviewed the final version.

Competing interests: The authors declare no competing interests.

Additional Information: Supplementary Information is available for this paper. Correspondence and requests for materials should be addressed to Noémie Lefrancq.

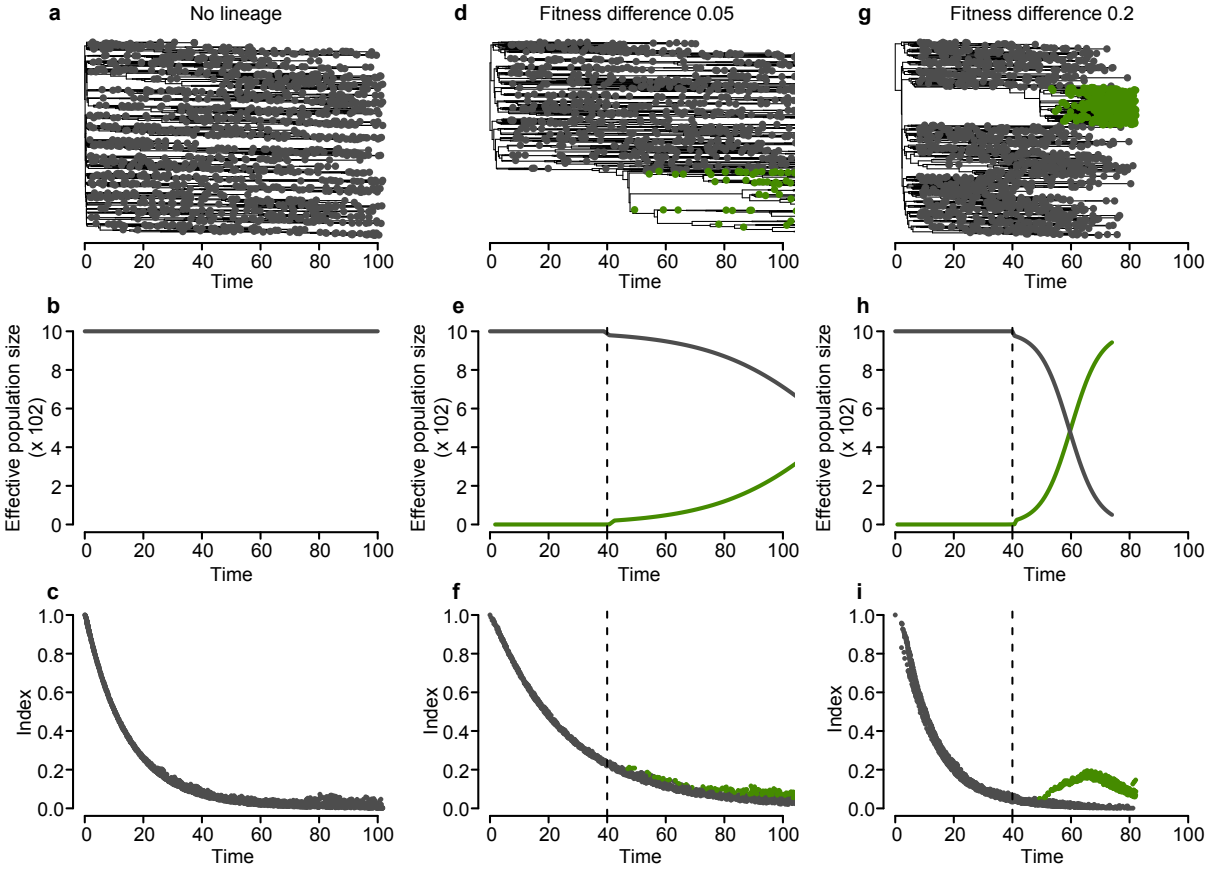
Data availability:

All *B. pertussis* sequences generated for this study were deposited in ENA, with accession numbers and metadata attached for each individual sequence available in Supplementary Table 10. All sequences and metadata used in this study, including the reference sequences, are listed in Supplementary Tables 10-13. All sequences are publicly available online on GenBank and ENA (*B. pertussis*, *M. tuberculosis*[20]) or GISAID (H3N2, SARS-CoV-2). The Supplementary Tables 10-13 are also available online in the repository: https://zenodo.org/records/13952222 [Ref: [62]].

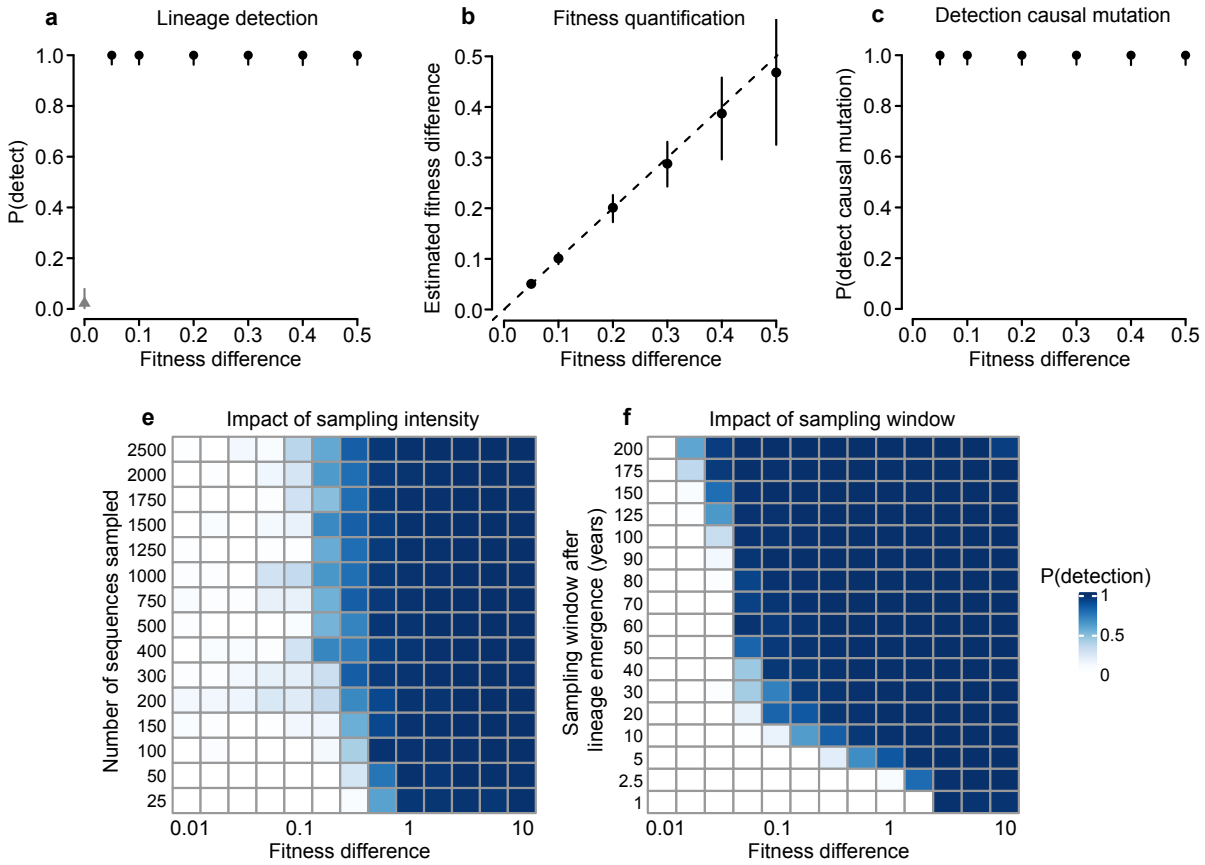
Code availability:

Code to replicate the main analyses of this paper is publicly available at https://zenodo.org/records/13952222 [Ref: [62]]. General guidelines to use *phylowave* and a step-by-step example are included in Supplementary Text 3-4.

Extended Data:

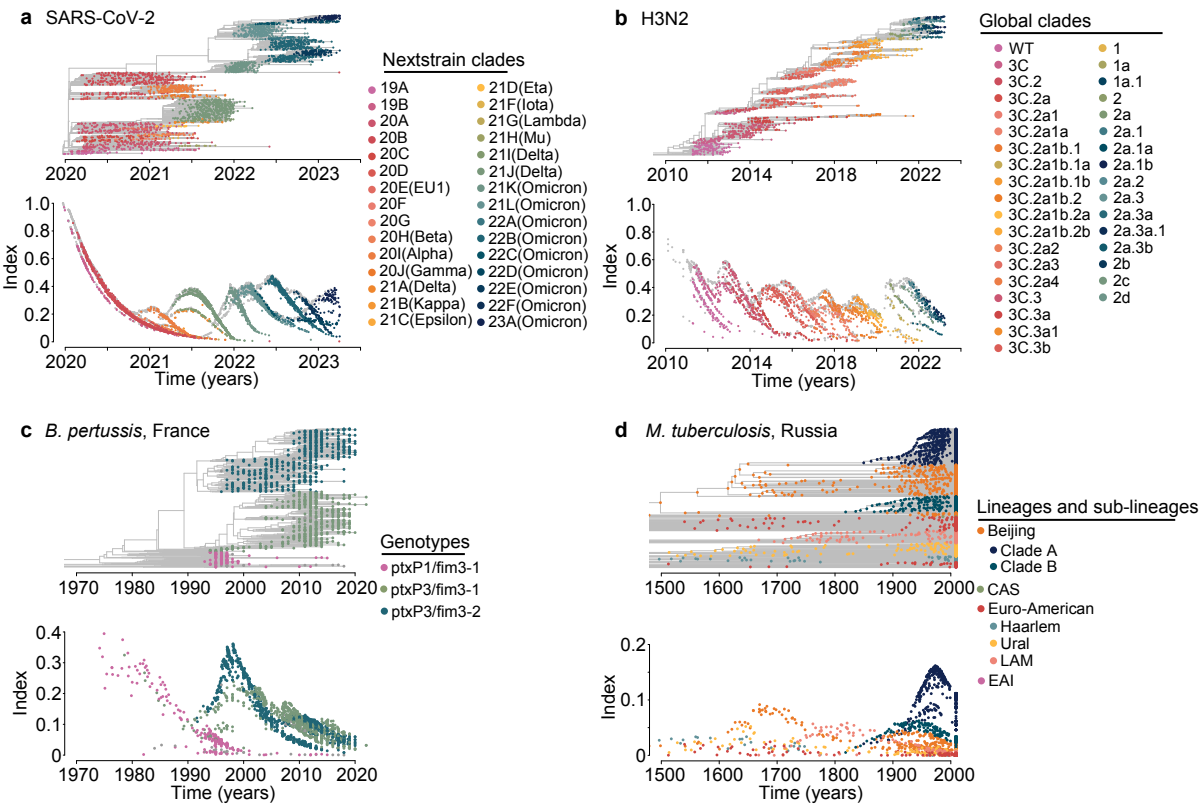


Extended Data Figure 1: Example of simulated dynamics for different fitness advantages.
We present examples of simulations in the case of no emerging lineages **(a-c)** and an emerging lineage with a fitness difference of 0.05 per time unit **(d-f)**, or 0.2 per time unit **(g-i)**. For each condition, we present a simulated tree, the effective population size for each lineage and the index of each node in the simulated tree. Colours denote each population: the background (grey) and the emerging lineage (green). The dashed line represents the time at which the emerging lineage was introduced (T = 40).



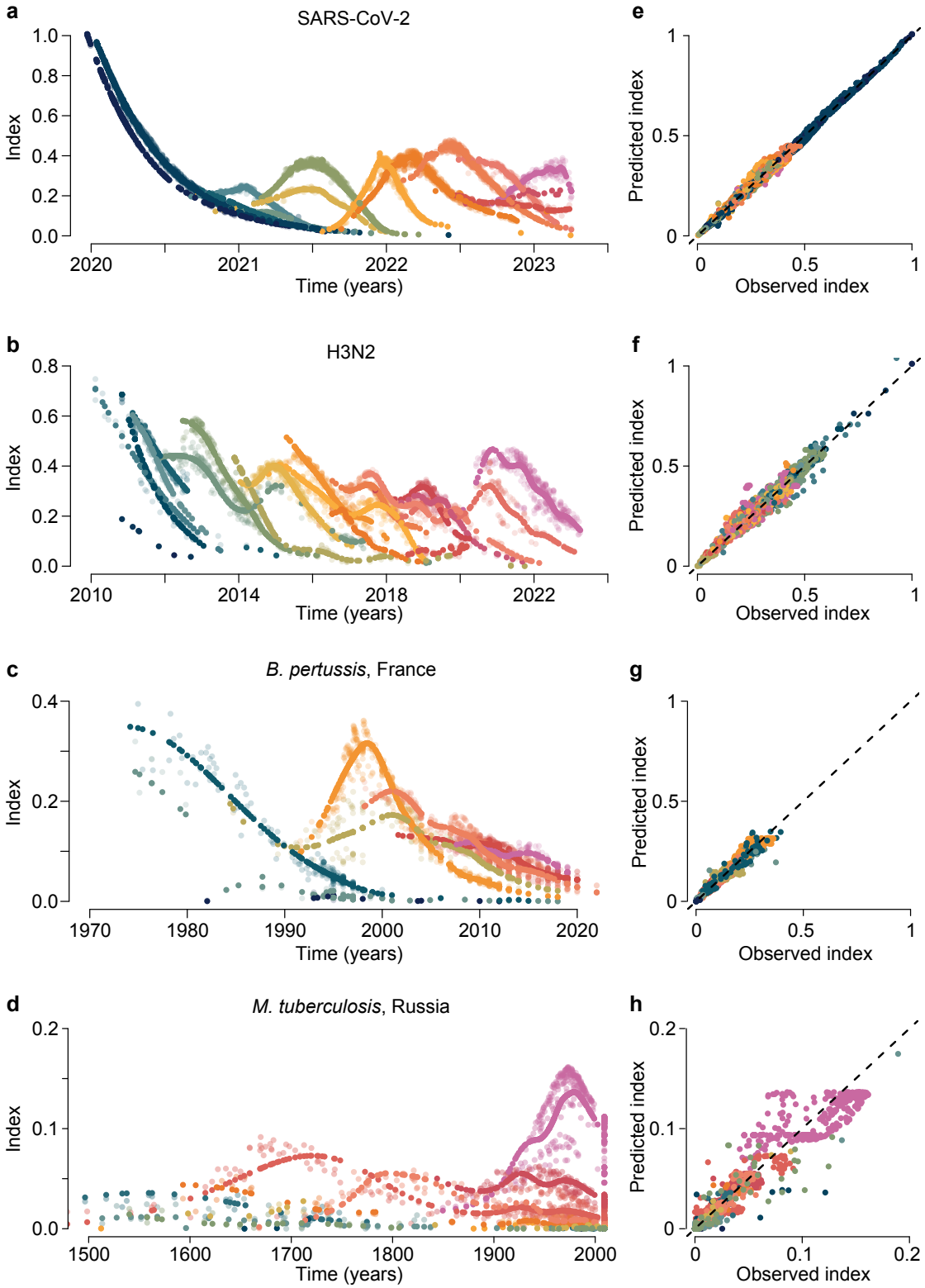
Extended Data Figure 2: Results of the simulation study to assess the ability of *phylowave* to detect emerging lineages.

(a-c) For each fitness value, we simulated a full lineage replacement (until the emerging lineage represents 95% of the population) and extracted 100 trees of 1500 tips each. We plot the proportion of times we detect the emerging lineage **(a)**, the estimated fitness difference, using our multinomial logistic model **(b)** and the proportion of times we detect the causal mutation **(c)**. Dots and bars in (a) and (c) denote the median and 95% binomial confidence intervals. The grey triangle in (a) denotes the case where no emerging lineages were simulated. Dots and bars in (b) denote the median and 95% credible interval of the model posterior. Details on the simulation studies can be found in Supplementary Text 2. **(d-e)** Ability of *phylowave* to detect an emerging lineage in datasets of different sizes (d) or sampling window post lineage emergence (e). In (d) we fix the sampling window after lineage emergence to 10 years. In (e) we fix the size of the datasets to 1500 sequences. For each scenario, 20 trees were simulated. Details on the simulation studies can be found in Supplementary Text S2.

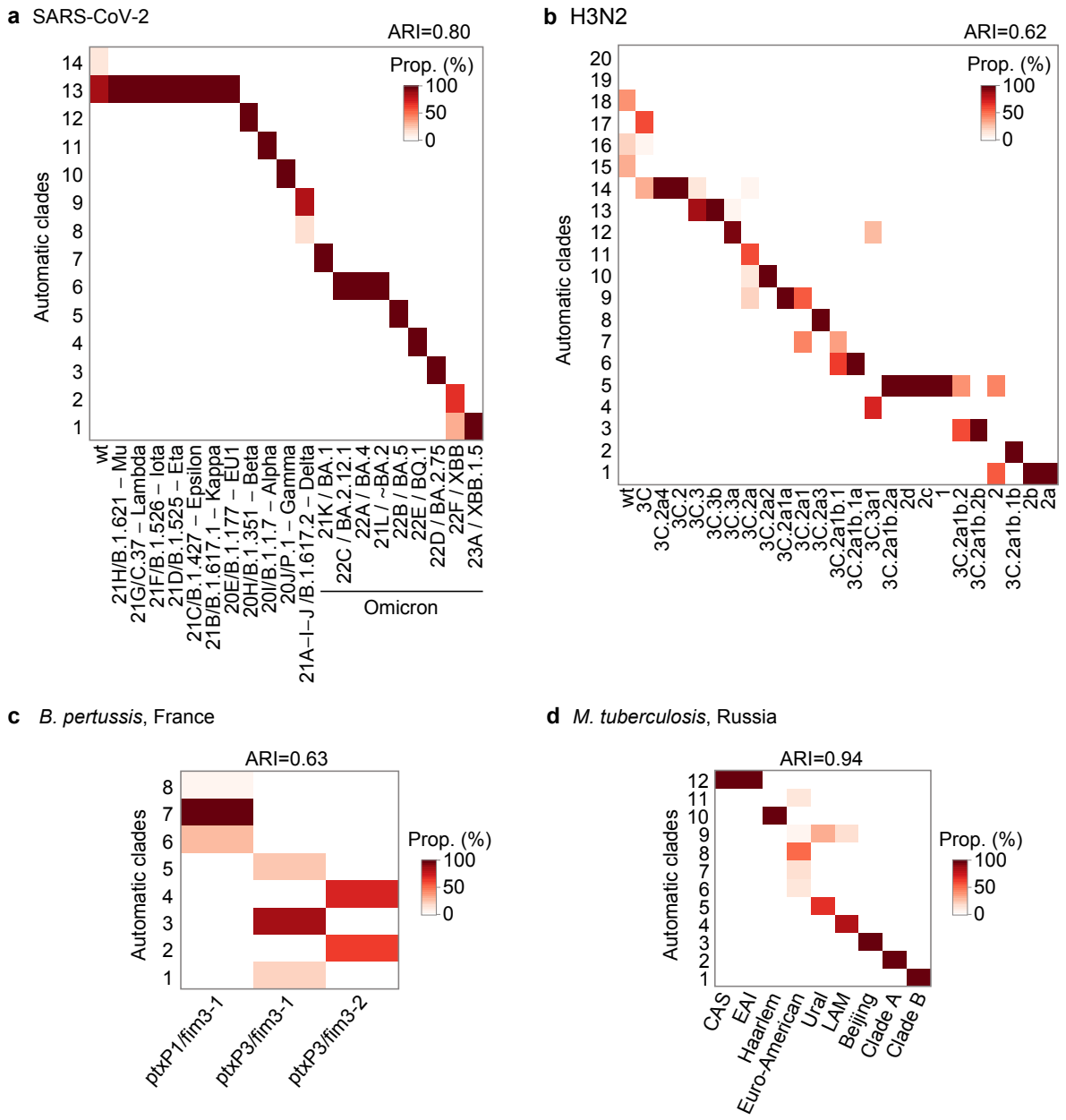


Extended Data Figure 3: Index dynamics coloured by known lineages

Similar to Fig. 1, for SARS-CoV-2 **(a)**, H3N2 **(b)**, *B. pertussis* **(c)** and *M. tuberculosis* **(d)**, we present the index dynamics computed at each node (terminal or internal). Here colours represent the different known clades, genotypes or lineages (see legend on the side). For *M. tuberculosis*, LAM denotes the Latin American-Mediterranean lineage, EAI denotes the East African Indian lineage and CAS denotes the Central Asian Strain lineage.

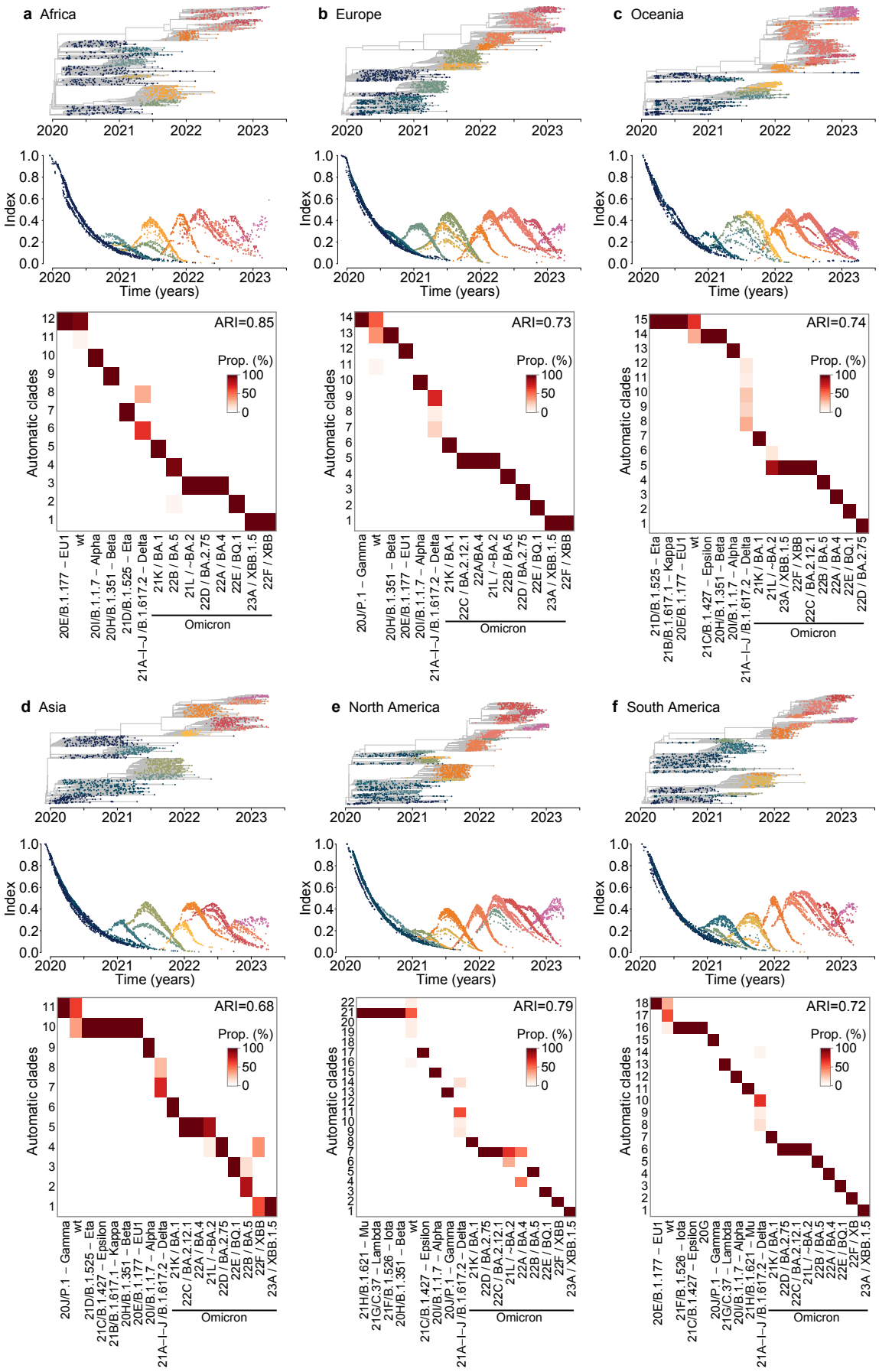


Extended Data Figure 4: Lineage detection based on index dynamics for each pathogen.
(a-d) For each pathogen we present model fits of the index dynamics using the best set of lineages. Solid dots represent the model prediction. Shaded dots represent the data. **(e-f)** Predicted versus observed index. The dashed lines denote identity lines. For each pathogen, colours represent the different lineages identified by their different index dynamics (same colours as in Fig. 1-4).



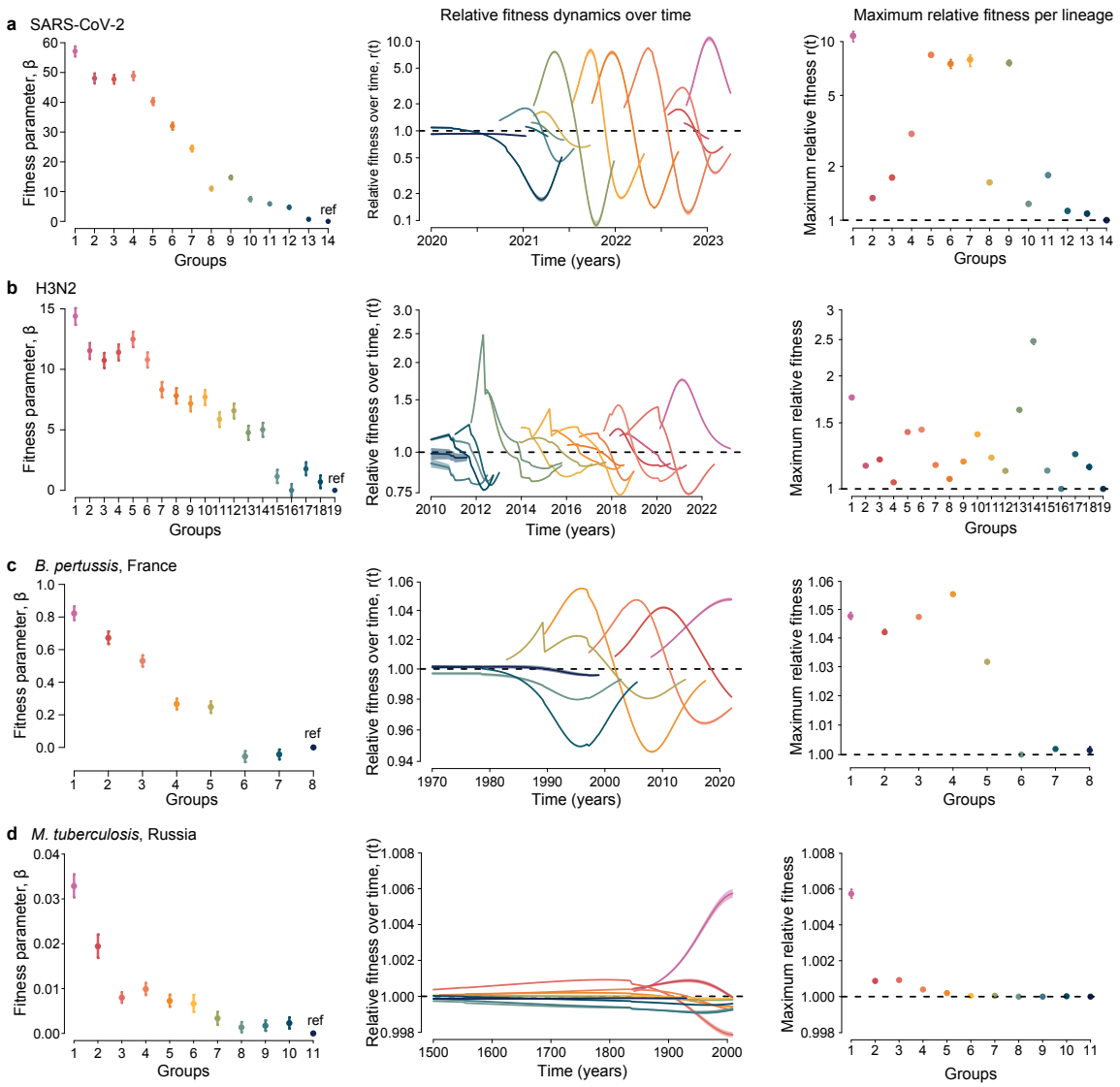
Extended Data Figure 5: Heatmaps comparing the identified lineages to the known population composition.

For SARS-CoV-2 **(a)**, H3N2 **(b)**, *B. pertussis* **(c)** and *M. tuberculosis* **(d)**, we present a heatmap comparing the known population structure (x-axis) to the automatic clades found by our *phylowave* (y-axis). Darker colours represent more agreement between both classifications. Contingency tables are presented in Tables S2-5.



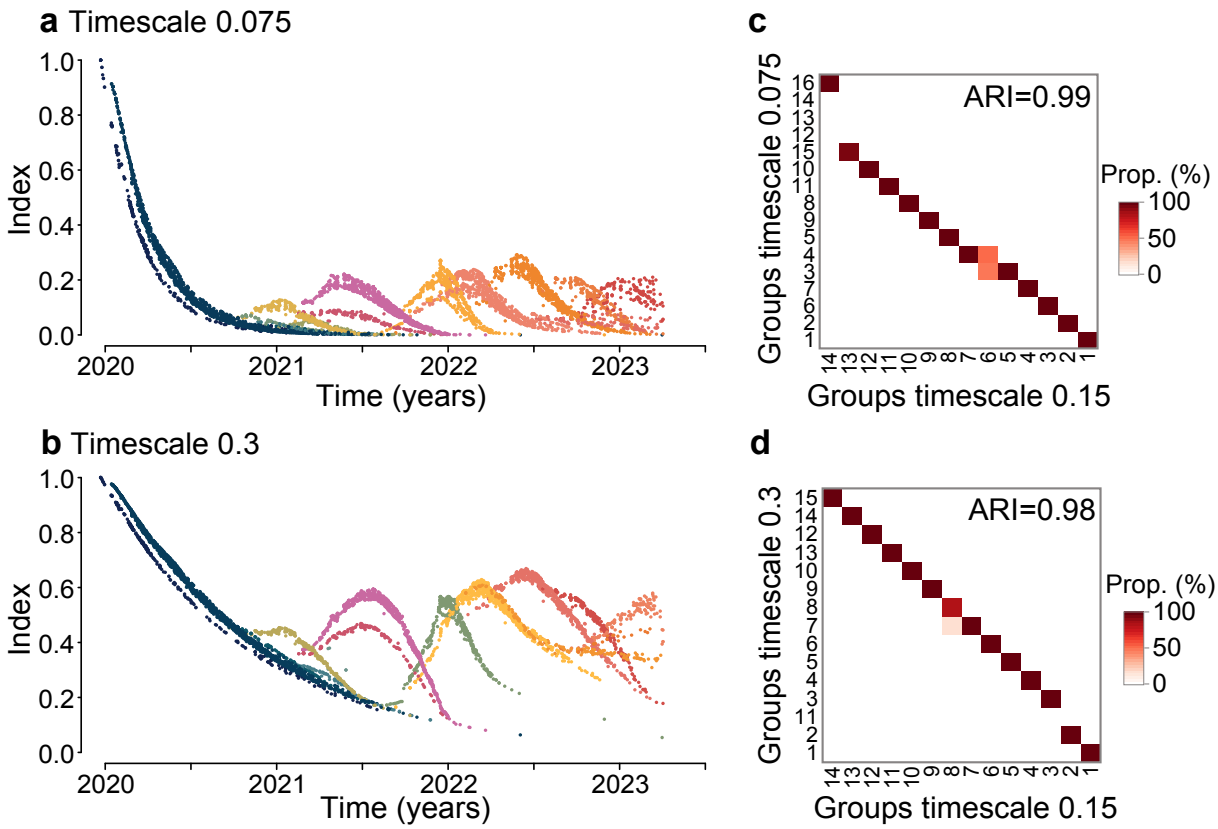
Extended Data Figure 6: SARS-CoV-2 index dynamics and lineages identified across continents.

We present the index dynamics computed at each node (terminal or internal) for datasets of SARS-CoV-2 isolated in Africa **(a)**, Europe **(b)**, Oceania **(c)**, Asia **(d)**, North America **(e)**, and South America **(f)**. The colours od the dots represent the different lineages identified by their different index dynamics. Timed-resolved phylogenies for each continent were obtained from NextStrain, accessed on 14 April 2023[37]. For each continent we also present a heatmap comparing the known clades identified by NextStrain (x-axis) to the automatic clades found by *phylowave* (y-axis). Darker colours represent more agreement between both namings.



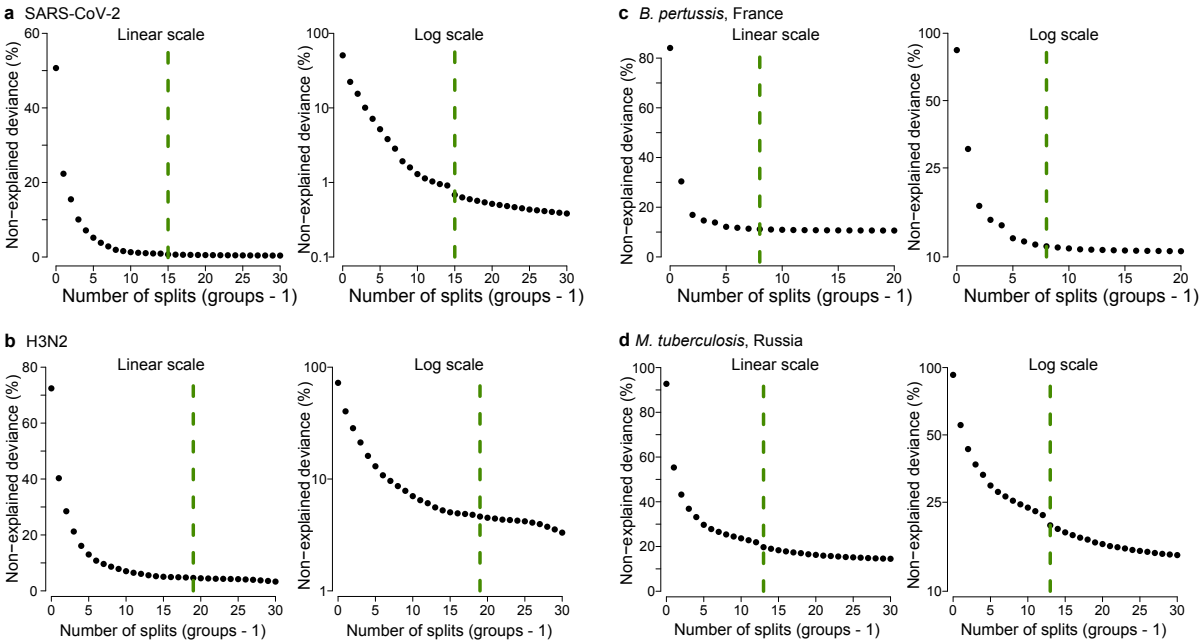
Extended Data Figure 7: Fitness estimates for all pathogen lineages

For SARS-CoV-2 **(a)**, H3N2 **(b)**, *B. pertussis* **(c)** and *M. tuberculosis* **(d)** we present the estimated fitness of each of their lineages. From left to right: Fitness parameter β for each lineage; Relative fitness dynamics overtime $r(t)$; maximum relative fitness per lineage. Dots represent median estimates for each lineage, bars denote 95% credible interval of the posterior. Lines and shaded areas represent the median and 95% credible interval of the posterior. Colours represent the different lineages identified for each pathogen.



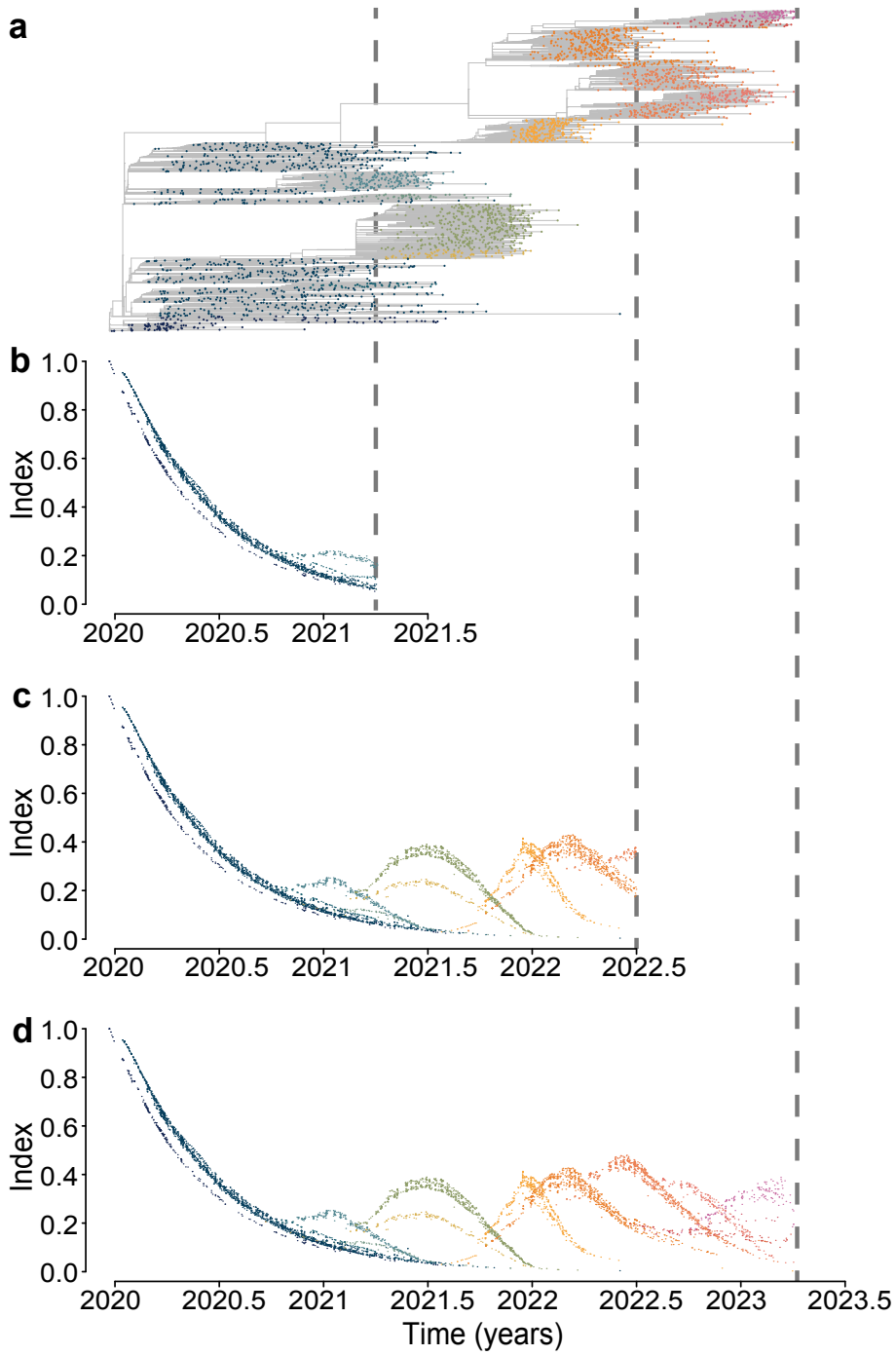
Extended Data Figure 8: Robustness of *phylowave* to the choice of timescale.

We show *phylowave* is robust to the choice of the timescale (governing the weight distribution used in the index computation). **(a-b)** Index dynamics computed on the global SARS-CoV-2 tree, with either a timescale of 0.075 (a) or 0.3 (b). The timescale used in the main analysis is 0.15. A smaller timescale focused more on recent population dynamics, a larger timescale focused more on the past evolution. Colours represent the lineages identified with our algorithm on those dynamics. **(c-d)** We compare the lineages identified with those timescales (y-axis) to the lineages presented throughout this study, with a timescale of 0.15 (x-axis). Darker colours represent more agreement between both namings. Overall, we find minimal differences in the lineages detected. Our results for SARS-CoV-2 are robust to the exact chosen timescale.



Extended Data Figure 9: Non-explained deviance as a function of the number of groups in the lineage detection algorithm.

For SARS-CoV-2 **(a)**, H3N2 **(b)**, *B. pertussis* **(c)** and *M. tuberculosis* **(d)**, we plot the proportion of non-explained deviance by the models with different numbers of groups. Dashed lines represent the number of groups chosen. We plot the proportion both on a linear scale (left) and log scale (right). The log scale enables a more precise appreciation of the number of groups at which the deviance explained does not increase substantially anymore.



Extended Data Figure 10: Example of index dynamics on time censored global SARS-CoV-2 datasets. **(a)** Global SARS-CoV-2 time-resolved phylogenetic tree, same as on Fig. 1-2. Dots denote terminal nodes only. **(b-c)** Index computed on censored datasets, on either 2021.26 (b) or 2022.5 (c). **(d)** Uncensored index dynamics. When censoring a dataset we prune all isolates not selected, effectively removing internal nodes as well as terminal nodes. This explains the slightly different dynamics observed near the censoring date. Colours represent the lineages automatically found by *phylowave* (same as Fig. 1-2).

Supplementary information guide

Manuscript: Learning the fitness dynamics of pathogens from phylogenies

Authors:
Noémie Lefrancq[1,2,3,*], Loréna Duret[1], Valérie Bouchez[3,4], Sylvain Brisse[3,4], Julian Parkhill[2,+], Henrik Salje[1,+]

Affiliations:

1. Department of Genetics, University of Cambridge, Cambridge, UK
2. Department of Veterinary Medicine, University of Cambridge, Cambridge, UK
3. Department of Biosystems Science and Engineering, ETH Zurich, 4009, Basel, Switzerland
4. Institut Pasteur, Université de Paris, Biodiversity and Epidemiology of Bacterial Pathogens, Paris, France
5. National Reference Center for Whooping Cough and Other Bordetella Infections, Paris, France

+ Joint senior authors
* Corresponding author: ncmjl2@cam.ac.uk

Table of content:

Supplementary tables

Supplementary information

Supplementary Text

Supplementary Text 1: Theoretical index behaviour

We provide here the analytical computation of the index behaviour in different populations. Let's recall Equation 1: we define the *Index* of each isolate i in its population at time t as:

$$Index(i) = \sum_{d=0}^{\infty} D_i(d,t) \cdot b^d$$

[Eq. 1]

With $D_i(d,t)$ the distance distribution - in number of mutations or evolutionary time (branch length) - from the isolate i to the rest of the population (internal and terminal nodes) at that time t (Fig. 1) and b^d, the kernel setting the weight of each distance d. b is the bandwidth, $b \in]0,1[$, which is a parameter to set, linked to the timescale (Table S1).

Using the probability $P_c(s = \frac{d}{\mu l}, t)$, for any pair of sequences sampled at time t, to coalesce some time $s = \frac{d}{\mu l}$ in the past, with μ being the rate at which the pathogen accumulates mutations per site and per unit of time, and l the length of its genome, we can write:

$$Index(t) = \int_0^{\mu l t} P_c(\frac{u}{\mu l}, t) \cdot b^u \ du$$

[Eq. 2]

Next, we write $P_c(\frac{u}{\mu l}, t)$ for different effective population sizes.

Expected behaviour of the index in a *constant effective population size*

In the simplest case of the structured coalescent process[17], if we consider two individuals from a population of constant size N_e, we can write their probability of coalescing some time s in the past as (Supplementary Fig. 18c):

$$P_c(s = \frac{d}{\mu l}, t) = \frac{1}{K} \frac{1}{N_e} exp(-\frac{s}{N_e}), \qquad if\ s \leq t \Leftrightarrow d \leq \mu l t$$
$$P_c(s = \frac{d}{\mu l}, t) = 0, \qquad if\ s > t \Leftrightarrow d > \mu l t$$

[Eq. 4]

With K the normalisation constant, so that $\int_0^{\infty} P_c(\frac{u}{\mu l}, t)\ du = 1$.

$$K = \mu l\ (1 - exp(-\frac{t}{\mu l\ N_e}))$$

We can plug Equation 3 in the index definition from Equation 2, making sure we take $s = \frac{d}{\mu l} \Leftrightarrow d = s\mu l$. After simplification it follows that:

$$Index(t) = \frac{(b \cdot exp(-\frac{1}{\mu l\ N_e}))^{\mu l t} - 1}{(\mu l\ N_e ln(b) - 1)\ (1 - exp(-\frac{t}{N_e}))}, \quad t > 0$$

[Eq. 5]

Which is the behaviour of the index as a function of time, in a constant population size.

Expected behaviour of the index in a *varying population size*

Following the work of Griffiths and Tavaré [18] on the coalescent process in varying population sizes, we can further derive the index in more complex population dynamics. We set the effective population size of our lineage to $N_e(t)$, which can vary through time. We can define the population-size intensity function $\boldsymbol{\Lambda}$ by [1819]:

$$\Lambda_t\,(s)\;=\;\int_0^s \frac{ds'}{N_e(t\;-\;s')},\quad\;\; t\;\geq s>0$$

We assume that $\Lambda(\infty)=\infty$, so that each pair of individuals may be traced back to a common ancestor with probability one [18]. The density λ of Λ is given by [18]:

$$\lambda_t(s)\;=\frac{1}{N_e(t-s)},\quad\;\; t\;\geq s>0$$

It follows that $P_c(s,t)$, i.e. the probability of waiting s time to have the first coalescent event is:

$$P_c(s,t)\;=\;\lambda_t(s)\;exp\;(-\Lambda_t(s)),\quad\;\; t\;\geq s>0$$

We can find back Equation 3, by taking $s=\frac{d}{\mu}$ and plugging in a constant population size $N_e(t)\;=\;N_e$:

$$\Lambda_t(s=\frac{d}{\mu l})\;=\;\int_0^{\frac{d}{\mu l}}\frac{ds'}{N_e}=\frac{d}{\mu l N_e}\;\;;\quad\lambda_t(s=\frac{d}{\mu l})\;=\frac{1}{N_e}$$

$$P_c(s=\tfrac{d}{\mu l},t)\;=\tfrac{1}{K}\,\tfrac{1}{N_e}exp\;(-\tfrac{d}{\mu l N_e}),\;\;t\;\geq\tfrac{d}{\mu l}>0$$

With K the normalisation constant. Next, we consider the case of exponentially varying population size.

Expected behaviour of the index in an *exponentially growing effective population size.*

We set: $N_e(t)\;=\;N_0\cdot e^{rt}$, with N_0 the initial population size and r the rate at which the population is growing (Supplementary Fig. 18f). We assume $r>0$. We can then define the new $\lambda_t(s)$ and $\Lambda_t(s)$:

$$\Lambda_t(s)\;=\;\frac{1}{N_0 r}e^{-rt}(e^{rs}-\;1)$$

And:

$$\lambda_t(s)\;=\frac{1}{N_0}e^{r(s-t)}$$

So that:

$$P_c(s\;=\frac{d}{\mu l},t)\;=\;\tfrac{1}{K}\tfrac{1}{N_0}e^{r(s-t)}exp\;(\tfrac{1}{N_0 r}e^{-rt}(1-e^{rs})),\qquad if\;\;t\;\geq s>0$$

$$P_c(s\;=\frac{d}{\mu l},t)\;=\;0,\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad if\;\;t\;<s$$

[Eq. 6]

With K the normalisation constant so that $\int_0^\infty P_c(\frac{u}{\mu l},t)\;du\;=\;1$.

Therefore, we can plug Equation 6 in the index definition from Equation 2, which leads to:

$$Index(t)=\frac{1}{K}\int_0^{\mu l t}\frac{1}{N_0}e^{r(\frac{u}{\mu l}-t)}exp\;(\frac{1}{N_0 r}e^{-rt}(1-e^{r\frac{u}{\mu l}}))\cdot b^u\;\;du\,,\quad\;\; t\;\geq\frac{d}{\mu l}>0$$

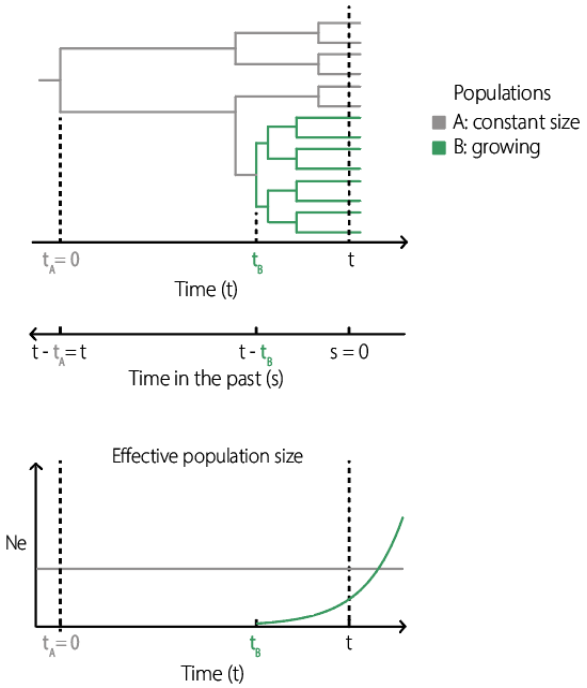
[Eq. 7]

This sum does not have a closed-form expression. However, it can be numerically approximated (Supplementary Fig. 18h).

Expected index behaviour for newly emerging lineage

We can note that in the case of a varying population size (e.g. exponentially varying), the index is dependent on r, the rate at which the population size is varying.

We can derive the index in structured populations that are more complex. For example, we consider here the case of a new lineage expanding in a population (schematic below). Let Pop_A be the ancestral population (schematic below, in grey), with constant effective population size N_A, and Pop_B, an offspring from Pop_A (schematic below, in green), which appeared at time t_B. At time t_B, the effective population size $N_B(t)$ of Pop_B is N_{B_0}. We assume that the Pop_B is growing exponentially ($N_B(t) = N_{B_0} exp(rt)$) through time with rate $r > 1$.



We now write the index of each population. We assume that the appearance of population B has a negligible impact on the index of the individuals sampled from population A. The effective size of population A is constant through time, therefore we can use Equation 5:

$$Index_{\,Indiv\ in\ Pop\ A}(t) = \frac{(b \cdot exp(-\frac{1}{\mu l\, N_A}))^{\mu l t} - 1}{(\mu l\, N_A ln(b) - 1)\,(1 - exp(-\frac{t}{N_A}))}, \quad t \geq 0$$

Population B is growing exponentially, within population A, therefore writing the index of individuals sampled from this population is more complex. Let's consider an individual sampled from population B. Its probability to coalesce with the rest of the population can be separated in two cases:

- It coalesces with an individual from population B, with probability $P_{c,B\to B}(s,t)$
- Or it coalesces with an individual from population A, with probability $P_{c,B\to A}(s,t)$

The total population through time is: $N_{tot}(t) = N_A + N_B(t)$.

Therefore, the probability of an individual sampled from population B to coalesce with another individual in the population is:

$$P_{c,B\to pop}(s,t) = \frac{N_B(t)}{N_{tot}(t)}\, P_{c,B\to B}(s,t) + \frac{N_A}{N_{tot}(t)}\, P_{c,B\to A}(s,t)$$

[Eq. 8]

We can note that $P_{c,BB}(s,t)$ exists only for $t > t_B$ (otherwise population B does not exist yet) and $t - t_B \geq s \geq 0$, and $P_{c,BA}(s,t)$ exists only for $s \geq t - t_B$.

First, let's write $P_{c,B\to B}(s,t)$. As population B is growing exponentially we can re-use Equation 7:

$$P_{c,B\to B}(s = \frac{d}{\mu l}, t) = \frac{1}{N_{B_0}} e^{r(s-t)} exp\left(\frac{1}{N_{B_0} r} e^{-rt}(1 - e^{rs})\right), \quad t \geq t_B \; and \; t - t_B \geq s \geq 0$$

[Eq. 9]

Second, let's write $P_{c,B\to A}(s,t)$. We note that this probability only exists for $s \geq t - t_B$, and the size of population A is constant. So we can rescale this probability:

$$P_{c,B\to A}(s,t) = P_{c,A}(s - t_B, t)$$

We can note that we already wrote this probability earlier in equation 4, so it follows that:

$$P_{c,B\to A}(s,t) = \frac{1}{N_A} exp(-\frac{s}{N_A}), \qquad if \; s - t_B > 0$$

[Eq. 10]

We can now plug Equations 9 and 10 into Equation 8, to obtain the index of individuals sampled from population B:

$$Index_{Indiv\ in\ PopB}(t) = \frac{1}{K}\left(\frac{N_B(t)}{N_{tot}(t)} \int_0^{\mu l(t-t_B)} \frac{1}{N_{B_0}} e^{r(\frac{u}{\mu l} - t)} exp\left(\frac{1}{N_{B_0} r} e^{-r(t-t}(1 - e^{r\frac{u}{\mu l}})\right) \cdot b^u \; du + \frac{N_A}{N_{tot}(t)} \int_{\mu l(t-t_B)}^{\mu l t} \frac{1}{N_A} exp(-\frac{1}{N_A} \cdot \frac{u}{\mu l}) \cdot b^u \; du \;\right), \quad t \geq t_B > 0$$

[Eq. 11]

With K the normalisation constant so that $\int_0^\infty P_{c,B}(\frac{u}{\mu l}, t)\, du = 1$.

Similarly to Equation 7, this Equation does not have a closed-form expression. However, it can be numerically approximated. Further, we can note that considering only two different populations already makes the index mathematically hard to track, at least without simplifying assumptions.

Supplementary Text 2: Details on simulation study

Simulations to verify the index dynamics

To verify the the expected index dynamics, we simulate trees for different population structures. We use the *sim2.bd.origin* function from the TreeSim package[63]. It simulates trees based on a birth-death model, with set rates of speciation (birth, λ) and extinction (death, μ). A constant effective population size can be simulated by $\lambda = \mu$. An exponentially growing effective population can be simulated by $\lambda > \mu$. To simulate a tree with an emerging lineage, we first simulate separately two trees, one with constant effective population size, and one with an exponentially growing effective population size. Then, we randomly select one tip from the first tree and use this tip as the root of the second tree. In Supplementary Fig. S18, we present those simulations, for three types of effective population sizes: constant, growing, and structured with an emerging lineage. We compare the simulation obtained with the formal expected dynamics (see derivations below). Overall, the simulations verify the formal expected dynamics. Parameters used: time window: 2 years, timescale: 1 year, substitution rate: 4 mutations per year.

We also reproduced sampling bias to check that our formal expected dynamics are correct even in that case. We sampled the sequences generated either taking 10% of the sequences from year 2-8 or only sequences from years 4-6 and 8-10 (and not years 1-3 or 6-8), mimicking common surveillance system biases. In Supplementary Fig. 19, we present those simulations, with 50 replicates each time. Overall, the simulations verify the validity of our approach. Parameters used: time window: 2 years, timescale: 1 year, substitution rate: 4 mutations per year.

Simulation study to test the ability of our approach to detect emerging lineages

To further test our approach to detect emerging lineages, we conducted a large simulation study. We repeatedly simulated timed phylogenetic trees where one lineage expands with a known fitness advantage compared to the background population. We used the package ReMASTER in BEAST2 to perform the simulations using a coalescent-based approach[64]. For each scenario, we simulated a tree by separately simulating the background tree and the emerging tree. Each tree was simulated in a coalescent-based fashion, by setting the effective size of each population. The background effective population size was set to 1000 for 40 time units, after which it decreased given a logistic growth model, reflecting the replacement of the background population by the emerging lineage. The emerging effective population size is set to 50 at T=40 (equivalent to a proportion of 5% in the population), and increases following a logistic growth model. The background tree and the emerging tree are then combined to form one unique tree. Trees are subsequently subsampled to a specific number of sequences. Example of simulated trees are presented in Extended Data Fig. 1.

We employ three different simulation strategies to test our model:

'Full lineage replacement' (Extended Data Fig. 2a-c). We simulate trees until the emerging lineage reaches 95% of the population, which is ensuring the emerging is seen replacing the background population. Each tree contains 1500 tips. We tested six different fitness differences, spanning 0.05/year to 0.5/year. These example values cover a broad range of dynamics ranging from slow to fast lineage replacement, consistent with what we observed in our case study pathogens. We also considered the case of a homogeneous population where no emerging lineage is present. In total, we simulated 700 trees (100 per scenario). In all the scenarios, we find that our method is able to identify

the emerging lineages with near-perfect precision (Extended Data Fig. 2a). *phylowave* is also specific: when no emerging lineage is present, our method does not falsely identify a lineage, in contrast to standard clustering tools, which still define clusters, even in the absence of a lineage with any selective advantage. Further, we quantified the fitness of each lineage by using our logistic model and correctly recovered their fitness (Extended Data Fig. 2b).

'Changing sampling intensity' (Extended Data Fig. 2d). We simulated trees until 10 years post lineage emergence and varied the number of sequences sampled (from 25 to 2500 sequences). We tested 13 different fitness differences, spanning 0.01/year to 10/year. These simulations were done using the same framework as above. For each scenario, we simulated 20 trees. In total, we simulated 3900 trees. Overall we found that the sampling intensity did not impact very much the lineage detection ability. All the lineages with a fitness >1/year were consistently detected, even in trees that were very sparsely sampled.

'Changing sampling window post lineage emergence' (Extended Data Fig. 2e). Lastly, we investigated the impact of the time between lineage emergence and the dates of sequences. As above, we simulated trees with 1500 tips each, however here we varied the time at which the simulation stopped (between 1 and 200 years). For each scenario, we simulated 20 trees. In total, we simulated 4420 trees. We found that lineages with only small fitness advantages require sequences covering longer time periods to be detected. Therefore, a lineage with a fitness advantage as low as 0.02/year, which would take >300 years to become the majority in the population, will not be detected with datasets that do not span that long. However, for larger fitness difference, *phylowave* was able to consistently identify the emerging lineage. We note that the time periods considered can be covered by internal or terminal nodes, not necessarily by sequences (terminal nodes) alone.

Additionally, to investigate how *phylowave* compares to existing approaches, we tested both fastbaps[14] and treestructure[15] on the simulated '*optimal signal*' datasets described above. To test fastbaps, we simulated sequence alignment for each simulated tree using the AliSim tool[65] from IQ-TREE[50]. We simulated an alignment of 3000 positions for each tree, with a substitution rate of 0.001 mutations per site per unit of time, an equal proportion of bases and a JC69 substitution model. We used default parameters to optimise the prior and find the best baps partition. To test treestructure, we ran the *trestruct* function on each simulated phylogeny with a minCladeSize of 30, a significance level of 0.005 and other parameters set to default. For each scenario and for each tree, we then compared the results obtained with each method (Supplementary Fig. 2). An example of the different methods is presented in (Supplementary Fig. 1). We found that our method was consistently better at recovering the population structure (i.e. one emerging lineage and a background population.

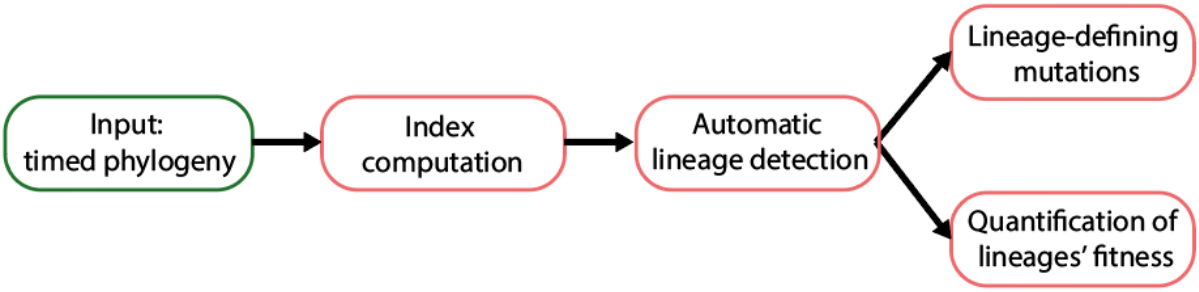
Supplementary Text 3: General guidance to use *phylowave* on a dataset.

This general guidance is also available in the README at https://zenodo.org/records/13952222 [Ref: [62]]*, together with the codes.*

phylowave is an approach that summarises changes in population composition in phylogenetic trees of pathogens, allowing for the automatic detection of lineages based on shared fitness and evolutionary relationships. It is currently written in R, with the exception of the fitness model which is written in Stan. In principle, *phylowave* is applicable to any pathogen (e.g. from viruses and bacteria) provided a timed-phylogeny is available and the sampling is representative of the diversity.

Here we describe the general organisation of the pipeline, which is detailed in the following sections:

- Input
- Index computation
- Lineage detection
- Quantification of lineages' fitness
- Lineage-defining mutations



Input

The minimal inputs to use the pipeline are:

- a **time-resolved phylogenetic tree**. The tree must be of class *phylo* and *binary*. The functions *ape::read.tree* and *ape::read.nexus* will enable you to load most trees (of newick or nexus formats, respectively) as objects of class *phylo*. You can test if you tree is binary with the function *ape::is.binary*. If it isn't binary you can always use the function *ape::multi2di* to resolve the polytomies artificially.
- the **sampling times of all tips**. This must be a vector of numerical values, of the same length as the number of tips.

Index computation

To compute the index of all nodes, use the function compute.index, you will need different inputs:

1. Data:
    - the *timed_tree*
    - *metadata* dataframe (see below the details of the *dataset_with_nodes*)
    - *distance matrix*: can be computed from the timed tree with the function *dist.nodes.with.names*

2. Information on the pathogen genome (these are pathogen specific):
    - The genome length of the pathogen considered: *genome_length* (in bp).
    - The mutation rate of the pathogen considered: *mutation_rate* (in bp/genome/year), an average is fine.
3. Index parameters (these are both pathogen-specific and dataset-specific):
    - The *timescale*: timescale (in years), this will be used to compute the bandwidth, see more details below.
    - Window of time on which to search for samples in the population: *wind*, see more details below.

The function outputs a vector containing the index of each node (internal and terminal).

More details on the input of the function:

dataset_with_nodes: To store the data throughout the analysis, we use a main metadata dataframe called dataset_with_nodes which looks like this:

ID	name_seq	time	is.node	Known clade classification	Index
1	name1	t1	'no'		
2	name2	t2	'no'		
3	name3	t3	'no'		
…	…	…	…		
n	namen	tn	'no'		
n+1	n+1	tn+1	'yes'		
…	…	…	…		
2n−1	2n−1	t2n−1	'yes'		

Where n is the number of tips (terminal nodes) in the tree. The column 'Known clade classification' is optional, but is useful to compare the results to existing sequence classifications. The index of each node (internal and terminal) is stored in the column 'Index'.

timescale: The timescale determines the kernel which enables to track lineage emergence dynamically, focusing on short distances between nodes (containing information about recent population dynamics) rather than long distances (containing information about past evolution). The timescale is tailored to the specific pathogen studied and its choice depends on the molecular signal,

as well as the transmission rate. In the study, we used timescales ranging from months (typical of RNA viruses) to years (typical of bacteria). To determine a timescale suitable for your dataset, we recommend thinking about the generation time of the pathogen considered, its mutation rate, and the amount of diversity already accumulated. For example, at the time of the analysis, SARS-CoV-2 was a new pathogen, spreading quickly and accumulating diversity at a rate of ~2 mutations per month. Therefore, a small timescale of less than a year chosen (0.15 years). On the contrary, *Mycobacterium tuberculosis* is an older and relatively slowly spreading pathogen, which accumulates mutations at a rate of ~0.2 mutation per year. A much larger timescale was then chosen (30 years), to reflect this. Ultimately, the best timescale is one that maximises the visualisation of population dynamics. We recommend trying different values.

wind: The choice of wind will depend on the sampling intensity of the dataset. It defines the window of time around each node on which to search for samples in the population. Ultimately it smooths the index dynamics. As a mean of example, for SARS-CoV-2, we set wind to 15 days, as the dataset was intensely sampled. But for *Bordetella pertussis,* which is more sparsely sampled, we chose a wind of 1 year. If wind is too large, then all the nodes are considered to be part of the same time window. If wind is too small, then only the nodes in direct proximity of the node of interest will be considered in the time window, which can result in noisy index dynamics. We recommend choosing a wind value that enables to span multiple sampling times, for example if you have samples and nodes every week, you may choose a wind of ~1-2 months. If you have samples and nodes every year, you may choose a wind of ~2 years.

For an example, see the SARS-CoV-2 code in Supplementary Text 4.

Lineage detection

To run the lineage detection algorithm, use the function *find.groups.by.index.dynamics*, you will need different inputs:

1. Data: *timed_tree* and *metadata* (*dataset_with_nodes*)
2. Lineage detection parameters:
    - *min_descendants_per_tested_node*: to start the analysis, start from nodes that have this minimum number of sequences
    - *min_group_size*: minimum group size, when creating a new potential split
    - *node_support*: numeric value of support of each node (e.g. mutations on the branch leading to the node, or bootstrap support)
    - *threshold_node_support*: threshold on the node support for the nodes to be considered in the detection algorithm
    - *weight_by_time*: size of the window of time on which to compute the weights (NULL or numeric, in years)
    - *weighting_transformation*: type of weighting to use (NULL, inv_freq, inv_sqrt, or inv_log)
    - *max_groups_found*: maximum number of groups to find (Integer)
3. Technical parameters: they do not necessarily need to be updated (see the function documentation for details): *p_value_smooth*, *stepwise_deviance_explained_threshold*,

stepwise_AIC_threshold, *k_smooth*, *parallelize_code*, *number_cores*, *plot_screening*, *keep_track* and *log_y*.

The function outputs multiple elements in a list:

- potential_splits: vector of the nodes included in most complex model tested
- best_dev_explained: vector of the deviance explained by the best models for each number of groups
- first_dev: the null deviance of the initial model (when no lineage is present)
- best_AIC : vector of the AIC of the best models for each number of groups
- best_BIC: vector of the BIC of the best models for each number of groups
- best_summary: list of the summaries of the best models for each number of groups
- best_mod: list of the best models for each number of groups
- best_groups: list of the groups used in the best models for each number of groups
- best_nodes_names: list of the nodes included in the best models for each number of groups

Typically, one chooses a value of *max_groups_found* greater than the expected number of lineages. The algorithm then runs until it finds all those groups, or until it cannot find any significant split anymore. The user can then check the deviance explained by all the models with increasing complexity and choose an adequate number of groups.

Once the split nodes have been defined, the user can then extract the group ID for each node using the function *merge.groups*. One can choose to refine these groups if needed, by setting a minimum number of nodes per group (*group_count_threshold*) or a minimum frequency of each group (*group_freq_threshold*).

For an example, see the SARS-CoV-2 code in Supplementary Text 4.

Post-hoc analyses

Two main post-hoc analyse can be done and are briefly described below.

Quantification of lineages' fitness

To quantify the fitness of each lineage, we developed a multinomial logistic model to fit the proportion of tips and nodes that belong to each lineage through time. This is done by using the function *estimate_rel_fitness_groups_with_branches*, which takes in entry:

- the *dataset_with_nodes* dataframe, with a column 'groups' which gives the group ID of each node in the dataset
- the timed tree
- *min_year*, the starting year at which to start fitting the model
- the window size (*window*), or number of windows (*N*), to divide the time series (from *min_year* to the last time point), compute proportions and fit the model.

For an example, see the SARS-CoV-2 code in Supplementary Text 4.

Lineage-defining mutations

Use the function *association_scores_per_group* to compute the association score of each mutation to each group. The function takes in entry:

- *dataset_with_nodes*
- *dataset_with_inferred_reconstruction*: a dataframe with the same first columns as *dataset_with_nodes* and then one column per snp its ancestral reconstruction along all nodes
- *tree*: timed tree
- *possible_snps*: the list of mutations that need to be considered
- *upstream_window*: for each group, how far upstream to consider mutations
- *downstream_window*: for each group, consider nodes from the MRCA up to this time

The codes to perform this analysis on the SARS-CoV-2 data is available in the file *2_4_Lineage_Defining_mutations.R*, in the folder *2_Functions*. As the SARS-CoV-2 genetic data is restricted, we cannot provide the raw data, but we provide all the sequence accession number to download them from GISAID.

Supplementary Text 4: Step-by-step guidance using the SARS-CoV-2 dataset as an example dataset.

This example, together with the codes, are also available in the README at https://zenodo.org/records/13952222 [Ref: [62]].

We provide here a working example on the SARS-CoV-2 timed phylogeny. You can go to specific sections of interest, however we recommend reading through all of this example.
Sections of this example:

- Load codes and SARS-CoV-2 data
- Compute the SARS-CoV-2 index dynamics
- Find SARS-CoV-2 clades based on index dynamics
- Quantify the fitness of detected SARS-CoV-2 lineage

Load codes and SARS-CoV-2 data

Load index functions

First, source all the necessary functions:

```
source(file = '2_Functions/2_1_Index_computation_20240909.R')
source(file = '2_Functions/2_2_Lineage_detection_20240909.R')
source(file = '2_Functions/2_3_Lineage_fitness_20240909.R')
```

Load necessary packages

```
library(ape, quiet = T); library(phytools, quiet = T); library(stringr, quiet = T)
library(MetBrewer, quiet = T); library(parallel, quiet = T); library(mgcv, quiet = T)
library(cowplot, quiet = T); library(ggplot2, quiet = T); library(ggtree, quiet = T);
library(cmdstanr, quiet = T); library(binom, quiet = T)
```

Versions:
Packages: ape v5.7-1, phytools v1.9-16, stringr v1.5.0, MetBrewer v0.2.0, parallel v4.1.2, mgcv v1.8-42, cowplot v1.1.1, ggplot2 v3.4.3, ggtree v3.2.1, cmdstanr v0.5.2, binom v1.1
R: 4.1.2

Load data

Load the NexStrain SARS-CoV-2 tree, in which all the tip name include: collection time, location and Pango lineage

```
tree_sars_cov2 =
read.nexus('1_Data/1_1_SARS_CoV_2/Tree_SARSCoV2_global_alltime_nextstrain_20230414.nexus')
## Make sure the tree is binary, and ladderized
tree_sars_cov2 = collapse.singles(ladderize(multi2di(tree_sars_cov2, random = F), right = F))
## Names all sequences
names_seqs = tree_sars_cov2$tip.label
n_seq = length(names_seqs)
## Collection times of all sequences
```

```
times_seqs = as.numeric(sapply(names_seqs, function(x)tail(str_split(x, pattern = '/')[[1]],2)[1]))
## Nextstrain clades of all sequences
clades_seqs = sapply(names_seqs, function(x)tail(str_split(x, pattern = '/')[[1]],1))
```

Compute the SARS-CoV-2 index dynamics

Index parameters

Set the index parameters.

```
## Length genome
genome_length = 29903 # reference nextstrain https://www.ncbi.nlm.nih.gov/nuccore/MN908947
## Mutation rate
mutation_rate = 8.1e-4 # mutation rate used by nextstrain https://github.com/nextstrain/ncov
## Parameters for the index
timescale = 0.15 ## Timescale
## Window of time on which to search for samples in the population
wind = 15 #days
wind = wind/365
```

Compute pairwise distance matrix

Compute distance between each pair of sequences and internal nodes in the tree

```
genetic_distance_mat = dist.nodes.with.names(tree_sars_cov2)
```

Get the time of each internal node

```
nroot = length(tree_sars_cov2$tip.label) + 1 ## Root number
distance_to_root = genetic_distance_mat[nroot,]
root_height = times_seqs[which(names_seqs == names(distance_to_root[1]))] - distance_to_root[1]
nodes_height = root_height + distance_to_root[n_seq+(1:(n_seq-1))]
```

Preparation data tips and nodes

Prepare the main dataframe, where the index and lineages of all nodes (internal and terminal) are going to be stored.

```
# Meta-data with all nodes
dataset_with_nodes = data.frame('ID' = c(1:n_seq, n_seq+(1:(n_seq-1))),
                                'name_seq' = c(names_seqs, n_seq+(1:(n_seq-1))),
                                'time' = c(times_seqs, nodes_height),
                                'is.node' = c(rep('no', n_seq), rep('yes', (n_seq-1))),
                                'Nextstrain_clade' = c(clades_seqs, rep(NA, n_seq-1)))
```

Compute index of every tip and node

```
dataset_with_nodes$index = compute.index(time_distance_mat = genetic_distance_mat,
                                         timed_tree = tree_sars_cov2,
                                         time_window = wind,
                                         metadata = dataset_with_nodes,
```

```
                    mutation_rate = mutation_rate,
                    timescale = timescale,
                    genome_length = genome_length)
```

Plot tree & index below, with colors from NextStrain clades

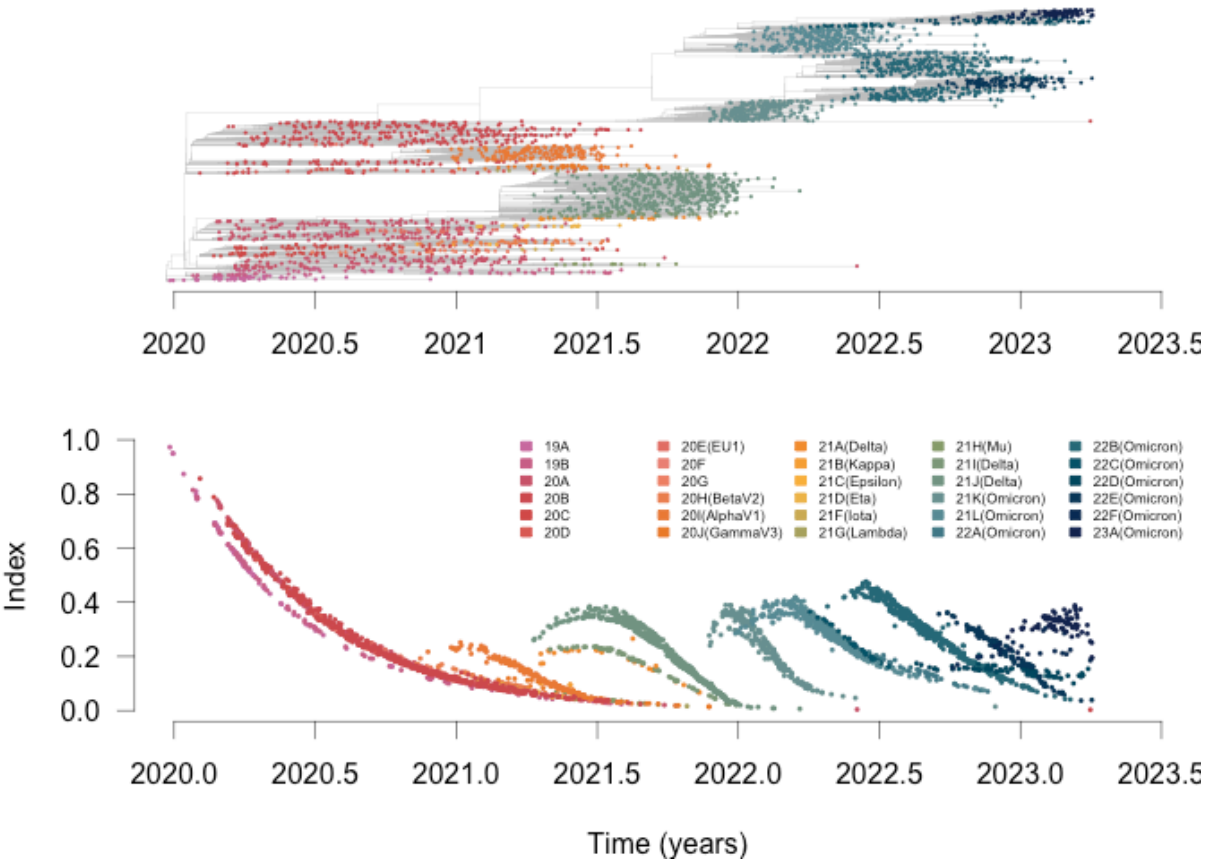
First, generate the color key, based on the Nextstrain clade of each sequence.

```
## Color key for Nextstrain clades
colors_clade = met.brewer(name="Cross",
n=length(levels(as.factor(dataset_with_nodes$Nextstrain_clade))), type="continuous")

## Color of each node, based on the key
dataset_with_nodes$Nextstrain_clade_color = as.factor(dataset_with_nodes$Nextstrain_clade)
clade_labels = levels(dataset_with_nodes$Nextstrain_clade_color)
levels(dataset_with_nodes$Nextstrain_clade_color) = colors_clade
dataset_with_nodes$Nextstrain_clade_color =
as.character(dataset_with_nodes$Nextstrain_clade_color)
```

Then plot the tree and index:

```
par(mfrow = c(2,1), oma = c(0,0,0,0), mar = c(4,4,0,0))

min_year = 2020
max_year = 2023.5

## Tree
plot(tree_sars_cov2, show.tip.label = FALSE,
   edge.color = 'grey', edge.width = 0.25,
   x.lim = c(min_year, max_year)-root_height)
tiplabels(pch = 16, col = dataset_with_nodes$Nextstrain_clade_color, cex = 0.3)
axisPhylo_NL(side = 1, root.time = root_height, backward = F,
         at_axis = seq(min_year, max_year, 0.5)-root_height,
         lab_axis = seq(min_year, max_year, 0.5), lwd = 0.5)
## Index
plot(dataset_with_nodes$time,
   dataset_with_nodes$index,
   col = adjustcolor(dataset_with_nodes$Nextstrain_clade_color, alpha.f = 1),
   bty = 'n', xlim = c(min_year, max_year), cex = 0.4,
   pch = 16, bty = 'n', ylim = c(0, 1),
   main = paste0(''),
   ylab = 'Index', xlab = 'Time (years)', xaxt = 'n', yaxt = 'n')
axis(2, las = 2, lwd = 0.5)
axis(1, lwd = 0.5)

# Color key
legend('topright',
```

```
legend = clade_labels,
fill = colors_clade, border = colors_clade,
cex = 0.5, bty = 'n', ncol = 5)
```



Find SARS-CoV-2 clades based on index dynamics

Run the lineage detection algorithm on SARS-CoV-2 data

Parameters for the detection:

```
time_window_initial = 2030;
time_window_increment = 100;
p_value_smooth = 0.05
weight_by_time = 0.1
k_smooth = -1
plot_screening = F
min_descendants_per_tested_node = 30
min_group_size = 30
weighting_transformation = c('inv_sqrt')

parallelize_code = T
number_cores = 2

max_stepwise_deviance_explained_threshold = 0
max_groups_found = 13
```

```
stepwise_AIC_threshold = 0

keep_track = T
```

Run the detection function (this steps takes approximately <10 min on 2 cores):

```
start_time = Sys.time()
potential_splits = find.groups.by.index.dynamics(timed_tree = tree_sars_cov2,
                                metadata = dataset_with_nodes,
                                node_support = tree_sars_cov2$edge.length[match((n_seq+1):(2*n_seq-
1), tree_sars_cov2$edge[,2])],
                                threshold_node_support = 1/(29903*0.00081),
                                time_window_initial = time_window_initial,
                                time_window_increment = time_window_increment,
                                min_descendants_per_tested_node =
min_descendants_per_tested_node,
                                min_group_size = min_group_size,
                                p_value_smooth = p_value_smooth,
                                stepwise_deviance_explained_threshold =
max_stepwise_deviance_explained_threshold,
                                stepwise_AIC_threshold = stepwise_AIC_threshold,
                                weight_by_time = weight_by_time,
                                weighting_transformation = weighting_transformation,
                                k_smooth = k_smooth,
                                parallelize_code = parallelize_code,
                                number_cores = number_cores,
                                plot_screening = plot_screening,
                                max_groups_found = max_groups_found,
                                keep_track = keep_track)
end_time = Sys.time()
print(end_time - start_time)
```

Instead, you may wish to load the results:

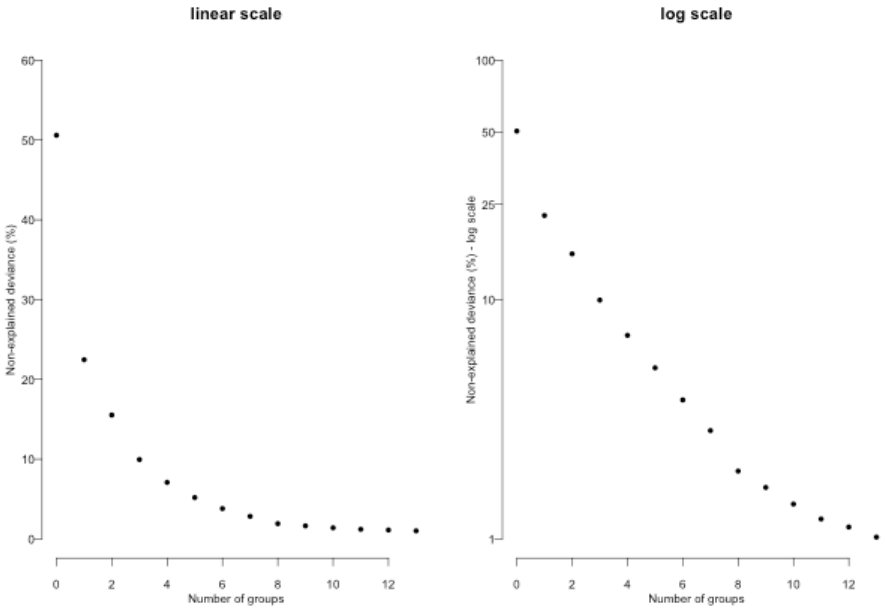
```
potential_splits = readRDS('README_files/potential_splits.rds')
```

Look at the deviance explained by the models with different number of groups.

Here for simplicity we directly chose max_groups_found = 13, which is the number of groups used in the original analysis. To decide on 13 groups, we initially ran the algorithm up to 30 groups.

```
df_explained_dev = data.frame('N_groups' = 0:length(potential_splits$best_dev_explained),
                    'Non_explained_deviance' = (1-c(potential_splits$first_dev,
potential_splits$best_dev_explained)),
                    'Non_explained_deviance_log' = log(1-c(potential_splits$first_dev,
potential_splits$best_dev_explained)))
```

```
df_explained_dev$Non_explained_deviance_log = df_explained_dev$Non_explained_deviance_log-
min(df_explained_dev$Non_explained_deviance_log)

par(mfrow = c(1,2), oma = c(2,2,1,1), mar = c(2,2,2,0.5), mgp = c(0.75,0.25,0), cex.axis=0.5,
cex.lab=0.5, cex.main=0.7, cex.sub=0.5)
plot(df_explained_dev$N_groups,
    df_explained_dev$Non_explained_deviance,
    bty = 'n', ylim = c(0, ceiling(10*max(df_explained_dev$Non_explained_deviance))/10),
    xaxt = 'n', yaxt = 'n', pch = 16, main = 'linear scale', cex = 0.5,
    ylab = 'Non-explained deviance (%)', xlab = 'Number of groups')
axis(1, lwd = 0.5, tck=-0.02)
axis(2, las = 2, at = seq(0,ceiling(10*max(df_explained_dev$Non_explained_deviance))/10,0.1),
    labels = seq(0, ceiling(10*max(df_explained_dev$Non_explained_deviance))/10,0.1)*100, lwd =
0.5, tck=-0.02)

plot(df_explained_dev$N_groups,
    (df_explained_dev$Non_explained_deviance),
    log = 'y',
    ylim = c(0.01, 1),
    bty = 'n',
    xaxt = 'n', yaxt = 'n', pch = 16, main = 'log scale', cex = 0.5,
    ylab = 'Non-explained deviance (%) - log scale', xlab = 'Number of groups')
axis(1, lwd = 0.5, tck=-0.02)
axis(2, las = 2, at = c(0.01, 0.1, 0.25, 0.5, 1),
    labels = c(0.01, 0.1, 0.25, 0.5, 1)*100, lwd = 0.5, tck=-0.02)
```



Optimize the number of groups: set the minimum number of sequences per group to 30, with a minimum frequency of 1%.

```
split = merge.groups(timed_tree = tree_sars_cov2, metadata = dataset_with_nodes,
                     initial_splits = potential_splits$potential_splits,
                     group_count_threshold = 30, group_freq_threshold = 0.01)
```

Label sequences with these new groups, and assign a color to each of them.

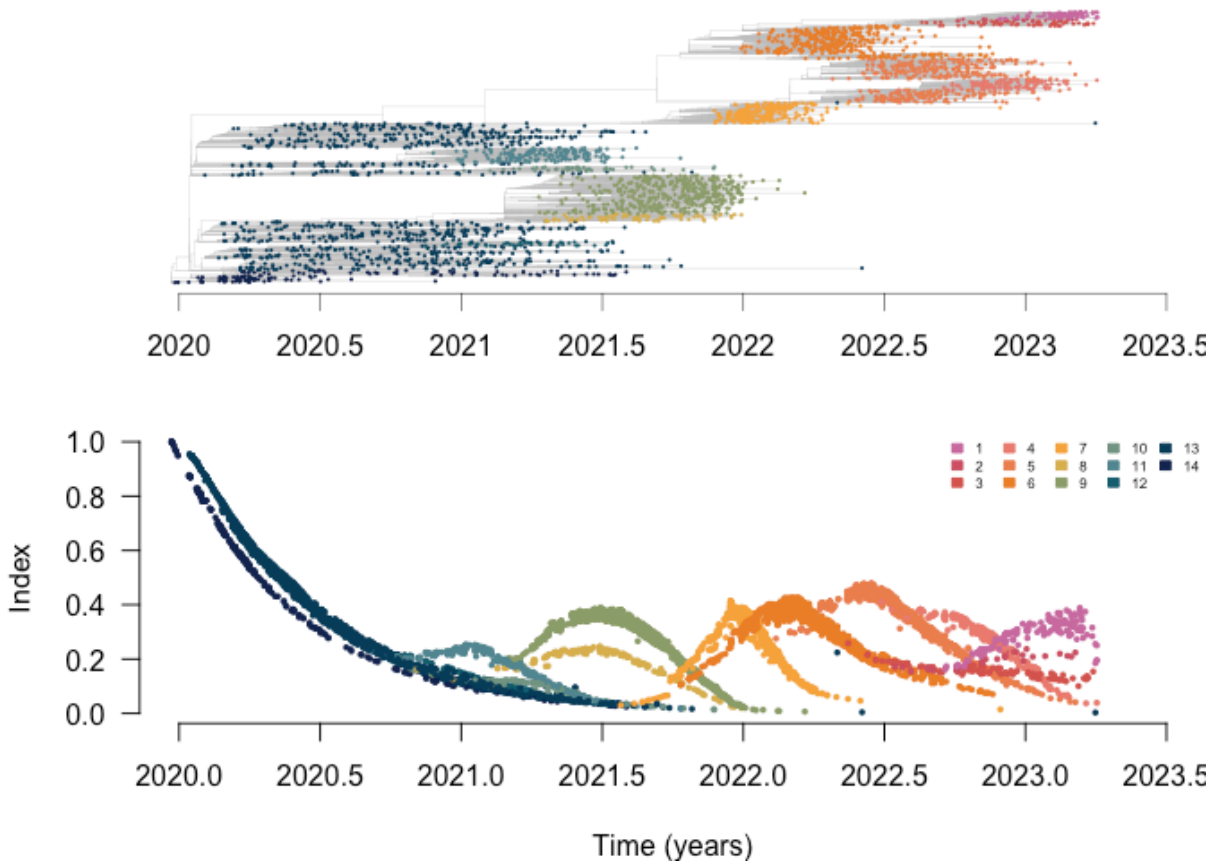
```
## Label sequences with new groups
dataset_with_nodes$groups = as.factor(split$groups)
## Reorder labels by time of emergence
name_groups = levels(dataset_with_nodes$groups)
time_groups_world = NULL
for(i in 1:length(name_groups)){
  time_groups_world = c(time_groups_world,
min(dataset_with_nodes$time[which(dataset_with_nodes$groups == name_groups[i] &
                                                    dataset_with_nodes$is.node == 'no')]))
}
levels(dataset_with_nodes$groups) = match(name_groups, order(time_groups_world, decreasing =
T))
dataset_with_nodes$groups = as.numeric(as.character(dataset_with_nodes$groups))
dataset_with_nodes$groups = as.factor(dataset_with_nodes$groups)
## Update names in split list
split$tip_and_nodes_groups = match(split$tip_and_nodes_groups, order(time_groups_world,
decreasing = T))
names(split$tip_and_nodes_groups) = 1:length(split$tip_and_nodes_groups)
split$groups = as.factor(split$groups)
levels(split$groups) = match(name_groups, order(time_groups_world, decreasing = T))
split$groups = as.numeric(as.character(split$groups))
## Choose color palette
n_groups <- length(name_groups)
colors_groups = (met.brewer(name="Cross", n=n_groups, type="continuous"))
## Color each group
dataset_with_nodes$group_color = dataset_with_nodes$groups
levels(dataset_with_nodes$group_color) = colors_groups
dataset_with_nodes$group_color = as.character(dataset_with_nodes$group_color)
```

Plot tree & index below, with colors from index-defined groups

Plot the tree and index colored with the new groups:

```
par(mfrow = c(2,1), oma = c(0,0,0,0), mar = c(4,4,0,0))

## Tree
plot(tree_sars_cov2, show.tip.label = FALSE,
    edge.color = 'grey', edge.width = 0.25,
```

```
    x.lim = c(min_year, max_year)-root_height)
tiplabels(pch = 16, col = dataset_with_nodes$group_color, cex = 0.3)
axisPhylo_NL(side = 1, root.time = root_height, backward = F,
             at_axis = seq(min_year, max_year, 0.5)-root_height,
             lab_axis = seq(min_year, max_year, 0.5), lwd = 0.5)

## Index colored by group
plot(dataset_with_nodes$time,
    dataset_with_nodes$index,
    col = adjustcolor(dataset_with_nodes$group_color, alpha.f = 1),
    bty = 'n', xlim = c(min_year, max_year), cex = 0.5,
    pch = 16, bty = 'n', #ylim = c(0, 1),
    main = paste0(''), #log = 'y',
    ylab = 'Index', xlab = 'Time (years)', yaxt = 'n')
axis(2, las = 2)
# Color key
legend('topright',
      legend = name_groups,
      fill = colors_groups, border = colors_groups,
      cex = 0.5, bty = 'n', ncol = 5)
```



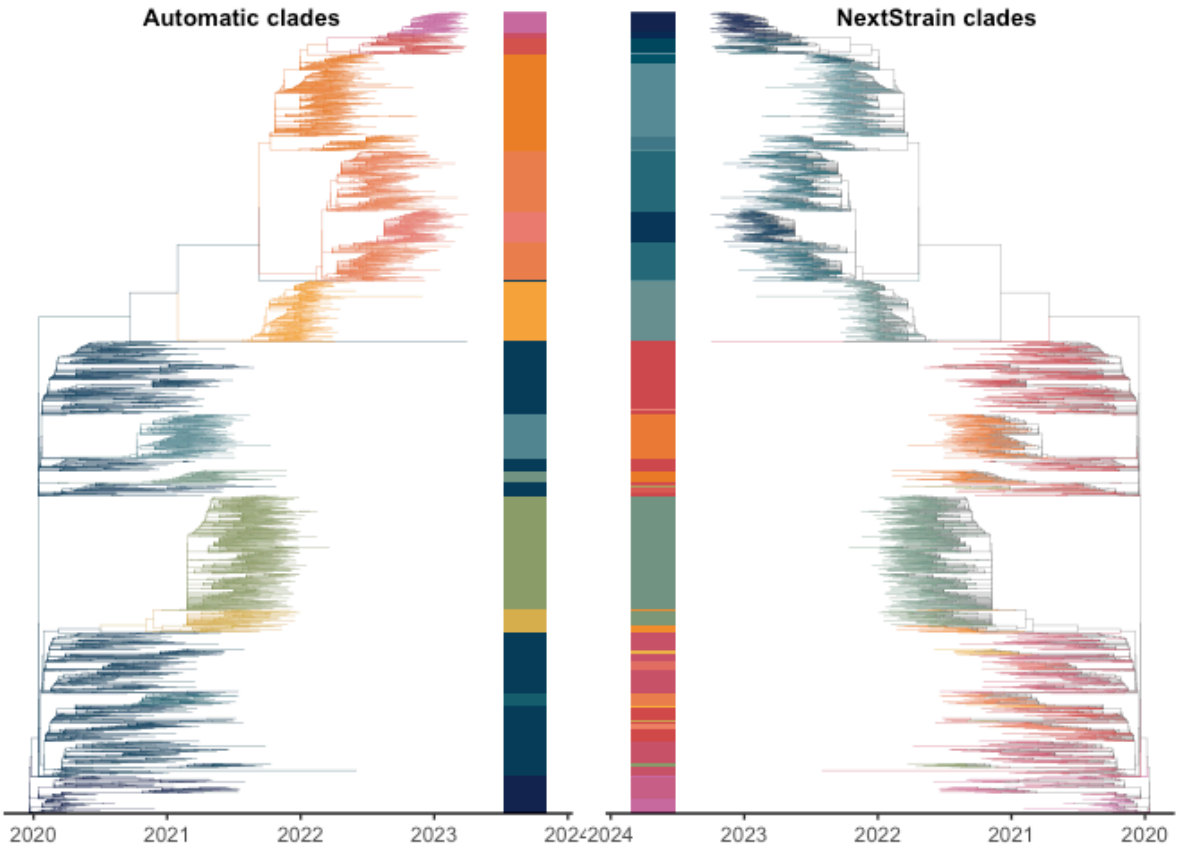
Compare NextStrain groups and groups called with the index

Generate SARS-CoV-2 trees coloured with each set of groups next to each other:

```
## Tree with index-defined groups
groups = matrix(dataset_with_nodes$groups[which(dataset_with_nodes$is.node == 'no')], ncol = 1)
colnames(groups) = 'groups'
rownames(groups) = dataset_with_nodes$name_seq[which(dataset_with_nodes$is.node == 'no')]
cols = as.character(colors_groups)
names(cols) = as.character(1:max(as.numeric(name_groups)))
plot_tree_sars_world_groups <- ggtree(tree_sars_cov2,
mrsd=lubridate::date_decimal(max(times_seqs)), size = 0.10,
                aes(color = as.character(dataset_with_nodes$groups))) +
  scale_color_manual(values = cols)+theme_tree2()
plot_tree_sars_world_groups = gheatmap(plot_tree_sars_world_groups, groups, offset=0.1,
width=0.10,
                 colnames=FALSE, legend_title="Group", color=NA) +
  scale_fill_manual(values = (cols))+scale_y_continuous(expand=c(0, 0.3))+theme(legend.position =
'none')

## Tree with NextStrain clades
Nextstrain = matrix(dataset_with_nodes$Nextstrain_clade[which(dataset_with_nodes$is.node ==
'no')], ncol = 1)
colnames(Nextstrain) = 'groups'
rownames(Nextstrain) = dataset_with_nodes$name_seq[which(dataset_with_nodes$is.node ==
'no')]
cols_NextStrain = as.character(colors_clade)
names(cols_NextStrain) = clade_labels
plot_tree_sars_world_Nextstrain <- ggtree(tree_sars_cov2,
mrsd=lubridate::date_decimal(max(times_seqs)), size = 0.10,
                                aes(color = as.character(dataset_with_nodes$Nextstrain_clade))) +
  scale_color_manual(values = cols_NextStrain)+
  theme_tree2(legend = 'none')
plot_tree_sars_world_Nextstrain = gheatmap(plot_tree_sars_world_Nextstrain, Nextstrain,
offset=0.1, width=0.10,
                  colnames=FALSE, legend_title="Group", color=NA) +
  scale_fill_manual(values = cols_NextStrain, na.value = 'white')+
  scale_x_reverse() +
  scale_y_continuous(expand=c(0, 0.3))+
  theme(legend.position = 'none')
```

Plot the generated SARS-CoV-2 trees:

```
plot_grid(plot_tree_sars_world_groups, plot_tree_sars_world_Nextstrain,
        rel_widths = c(1, 1), labels = c('Automatic clades', 'NextStrain clades'), label_size = 10, label_x =
c(0.1, 0.25), ncol = 2)
```



Quantify the fitness of detected SARS-CoV-2 lineages

Run the fitness model

Quantify the fitness of each group you can run the code (this steps takes approximately <5 min on 3 cores):

```
start_time = Sys.time()
## Load and compile stan code (this can take a few minutes)
model_compiled <- cmdstan_model(stan_file =
'2_Functions/Model_multinomial_logistic_birthdeath_lineage_fitness_20231220.stan')
## Run model on SARS-CoV-2 groups
res_fitness = estimate_rel_fitness_groups_with_branches(dataset_with_nodes =
dataset_with_nodes,
                                                        tree = tree_sars_cov2,
                                                        min_year = 2020,
                                                        window = 30/365,
                                                        model_compiled = model_compiled,
                                                        iter_warmup = 250, iter_sampling = 500, refresh = 50, seed = 1)
end_time = Sys.time()
print(end_time - start_time)
```

You might encounter a warning saying that '*alpha_true_GA*' has a missing init value - this is normal as those groups (ancestral groups that are not present at the start of the time series) do not always exist and therefore there is no default initial value. This is does not impact the model run. The seed has been set to 1 so allow for reproducible results.

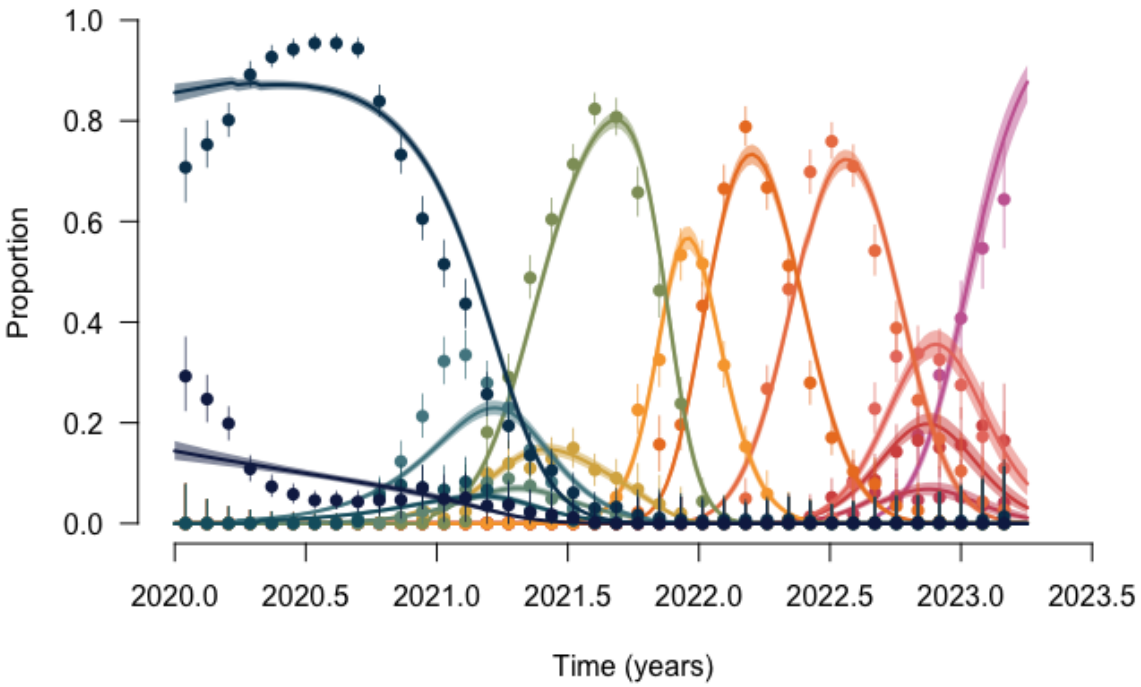
To save some time, you may wish to load the results:

```
res_fitness = readRDS('README_files/res_fitness.rds')
```

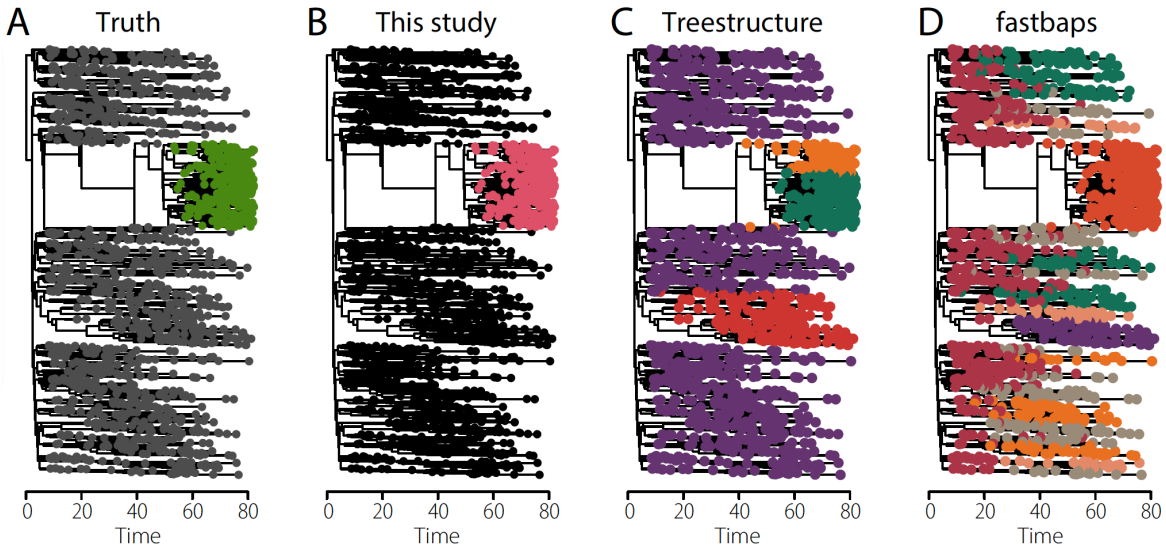
Plot the fits and estimated parameters

Plot the fits:

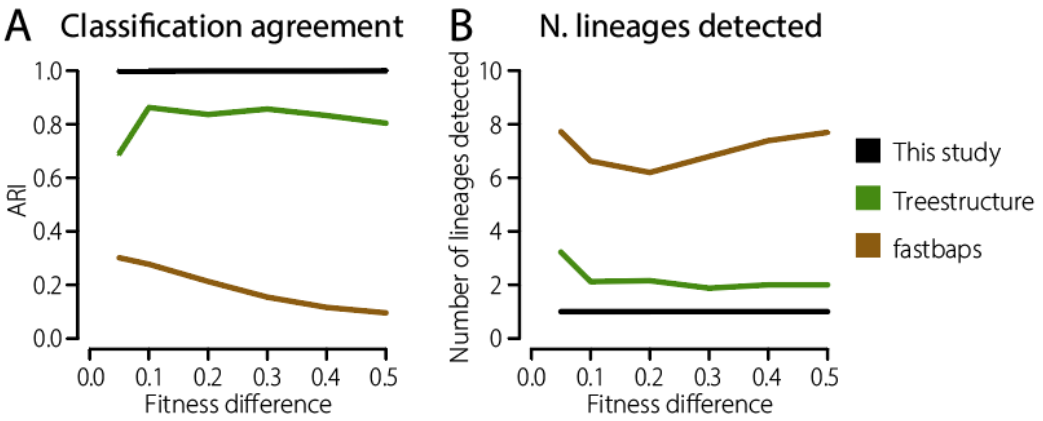
```
order_colors = order(as.numeric(split$tip_and_nodes_groups))
colour_lineage = colors_groups[match(split$tip_and_nodes_groups[order_colors], name_groups)]

plot_fit_data_new(data = res_fitness$data,
                  Chains = res_fitness$chains,
                  colour_lineage = colour_lineage,
                  xmin = 2020, xmax = 2023.5)
```



Supplementary figures

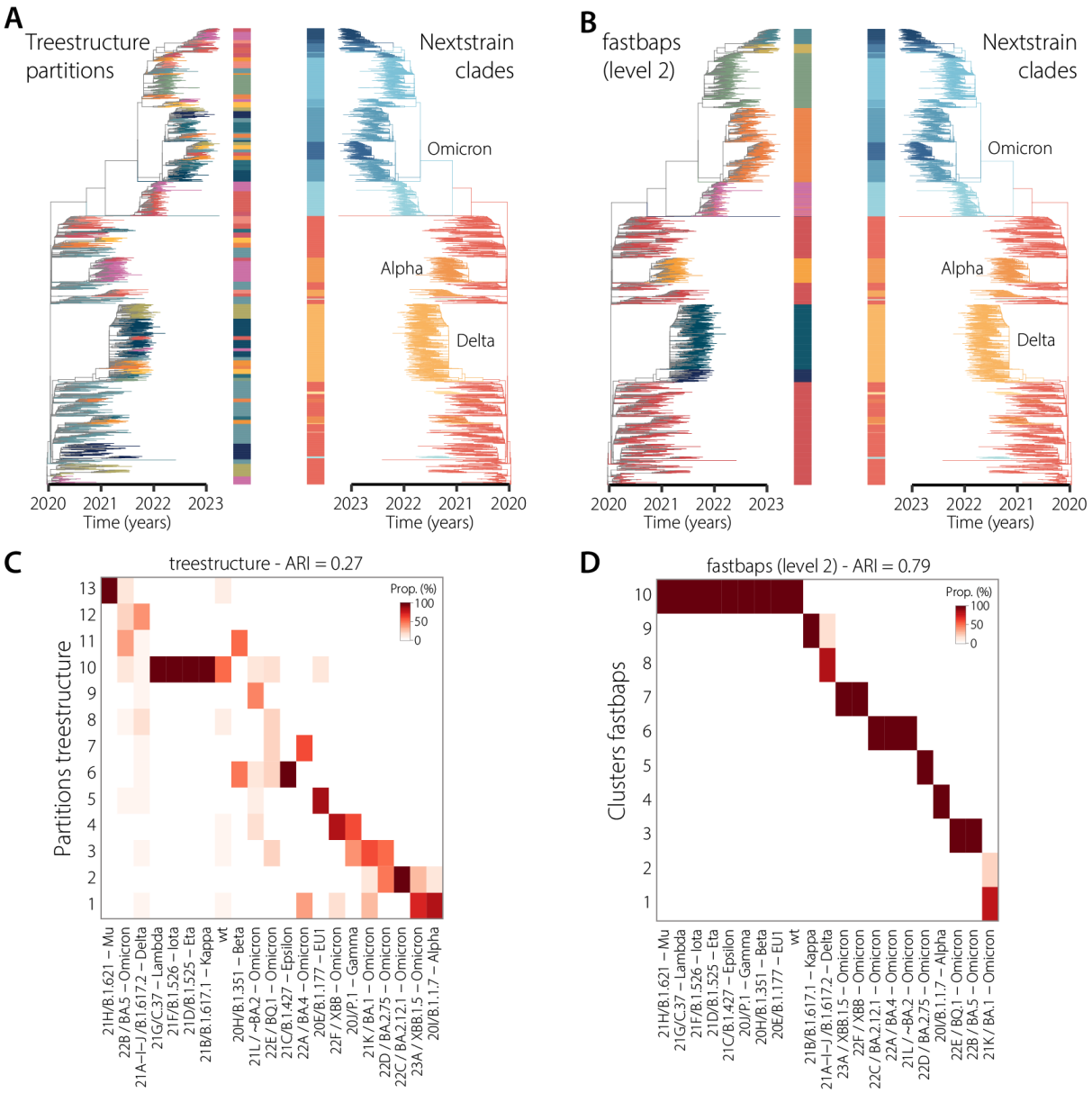


Supplementary Figure 1: Example of classification with our method, treestructure and fastbaps. We present an example of results on one simulated tree consisting of a background population (grey) and one emerging lineage (green) with a fitness advantage of 0.2 per time unit **(A)**. The results with our method are presented in **(B),** treestructure[15] in **(C)**, and fastbaps[14] in **(D).** For each method, the colours indicate the different lineages identified.



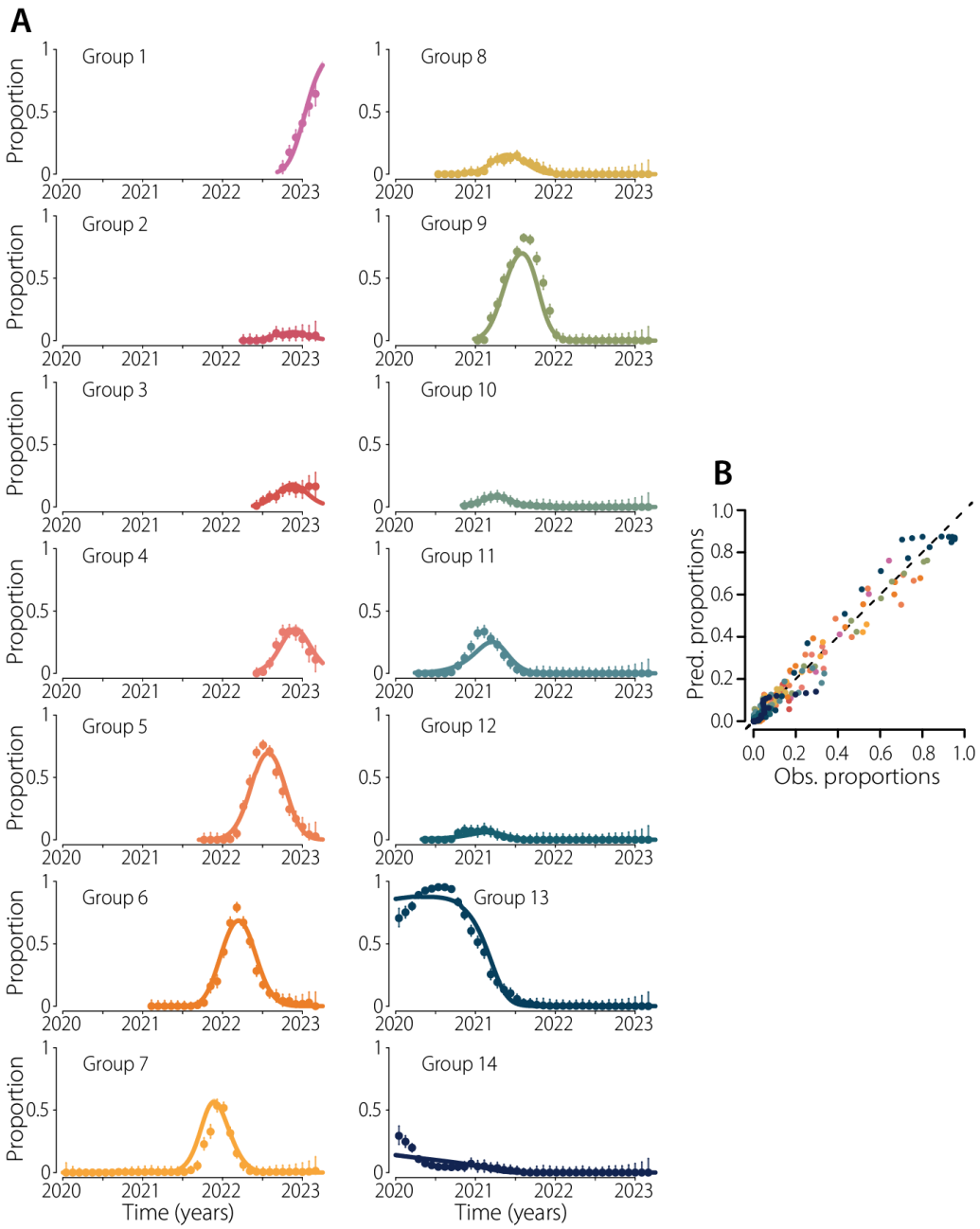
Supplementary Figure 2: Comparison of the lineages found by our method, treestructure and fastbaps on simulated datasets.

For each scenario and each method, we plot the classification agreement (Adjusted Rand Index [21]) **(A)**, and the number of lineages detected **(B)**. Colours denote the different methods.



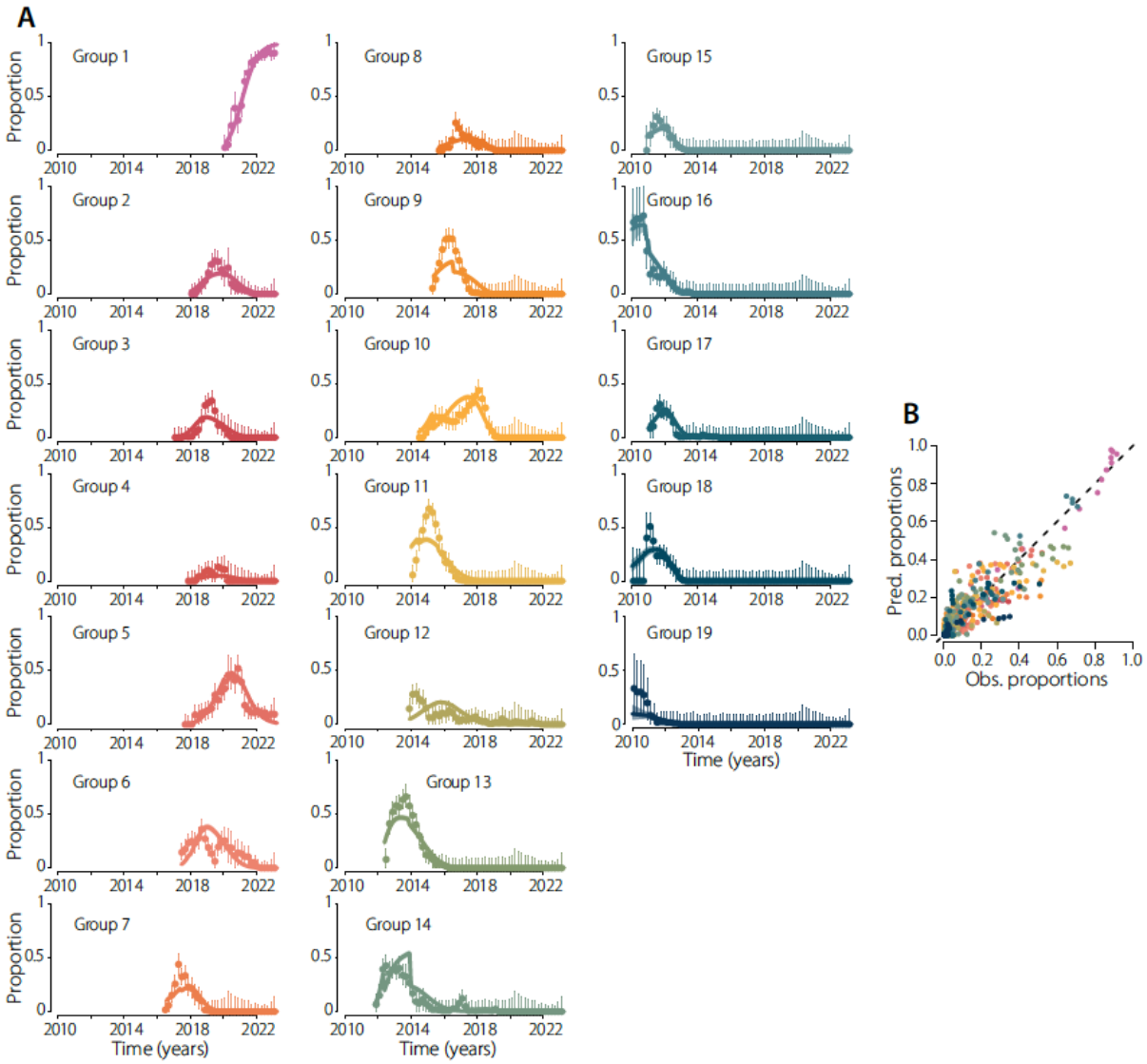
Supplementary Figure 3: SARS-CoV-2 lineages identified with treestructure and fastbaps

(A-B) Global SARS-CoV-2 trees coloured by the lineages identified with treestructure (A), or fastbaps (B). **(C-D)** We compare the lineages identified with either algorithm (y-axis) to the NextStrain clades (x-axis). Darker colours represent more agreement between both namings.



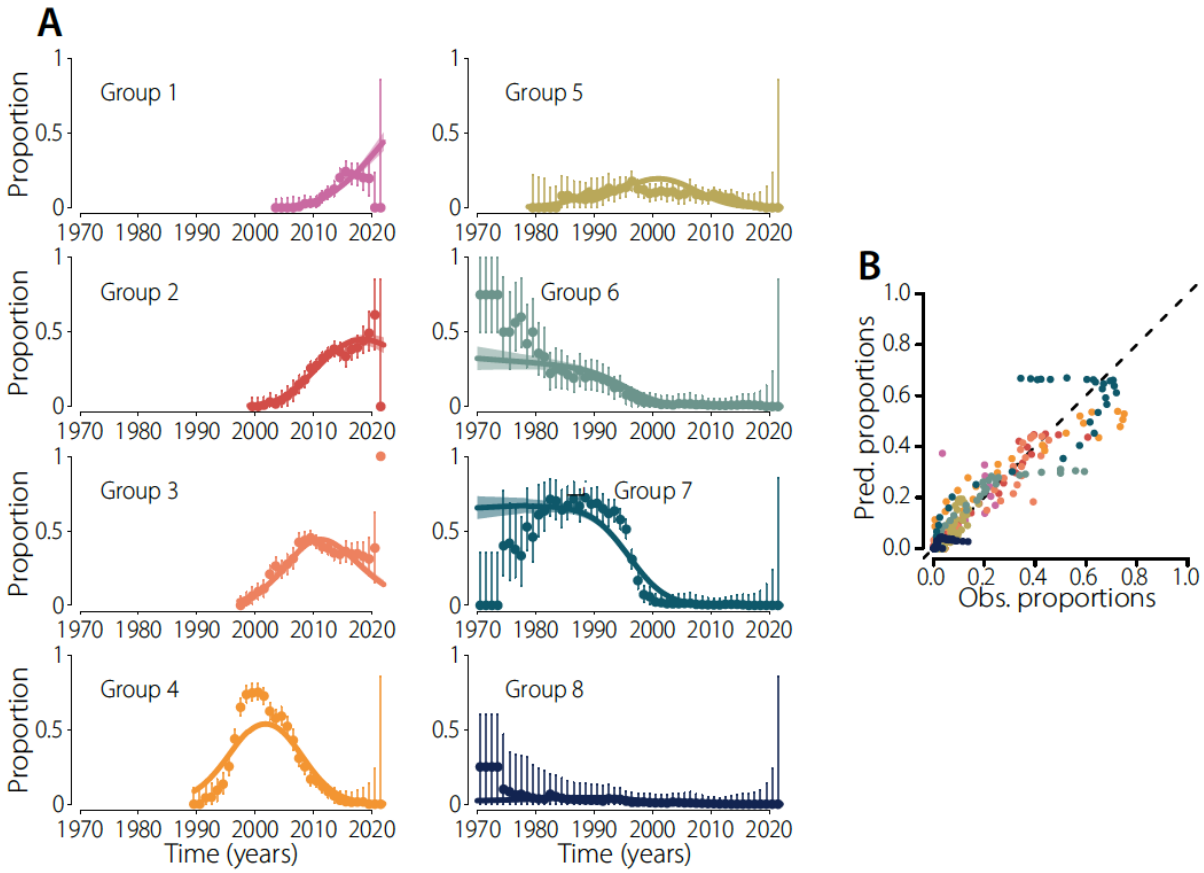
Supplementary Figure 4: Fitness model fits for all lineages of SARS-CoV-2

(A) Fits of the proportion of all the SARS-CoV-2 lineages. Coloured dots represent data, bars denote 95% confidence intervals. Coloured lines and shaded areas represent the median and 95% credible interval of the posterior. **(B)** Predicted versus observed proportions.



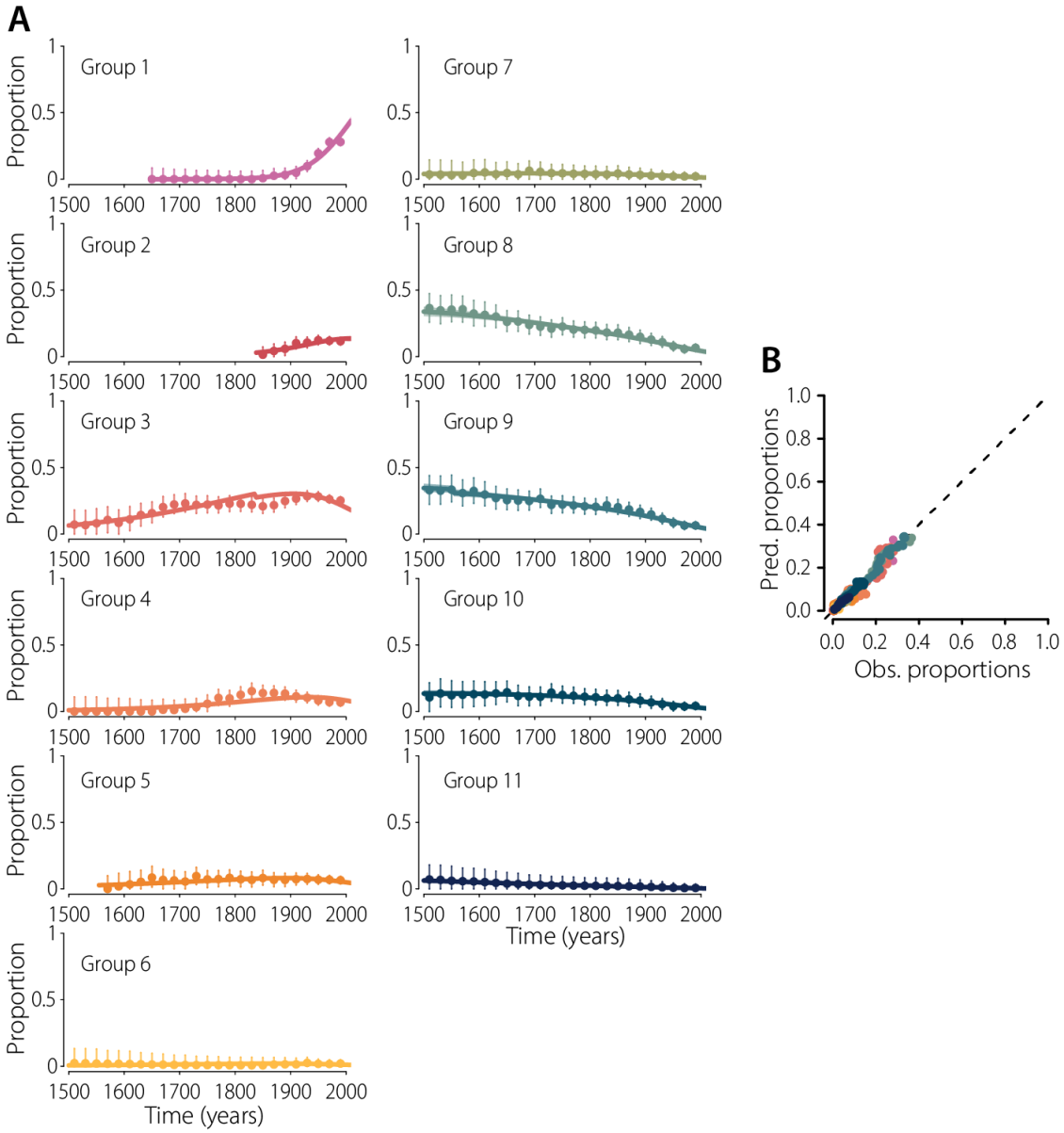
Supplementary Figure 5: Fitness model fits for all lineages of H3N2

(A) Fits of the proportion of all the H3N2 lineages. Coloured dots represent data, bars denote 95% confidence intervals. Coloured lines and shaded areas represent the median and 95% credible interval of the posterior. **(B)** Predicted versus observed proportions.



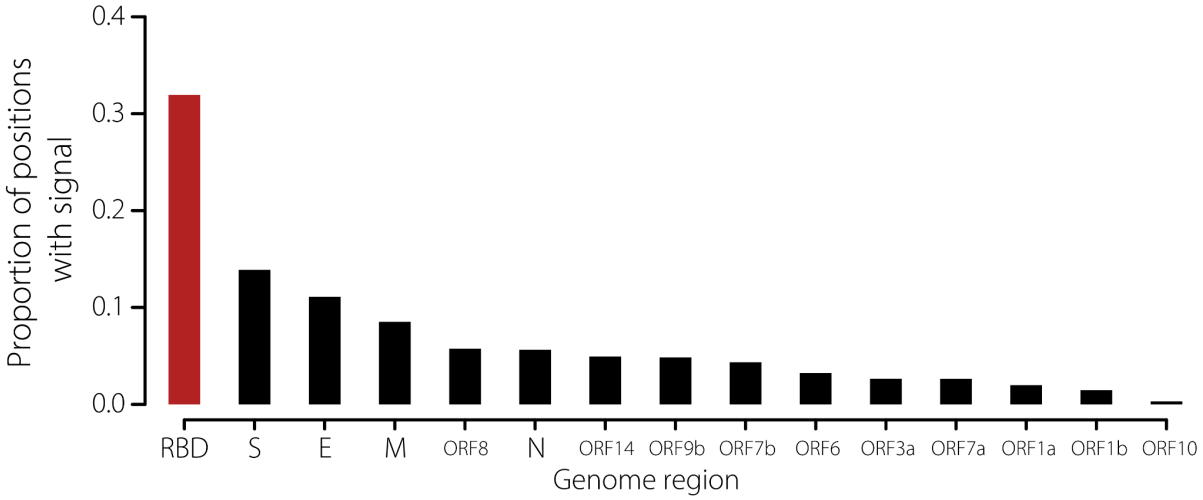
Supplementary Figure 6: Fitness model fits for all lineages of *B. pertussis*

(A) Fits of the proportion of all the *B. pertussis* lineages. Coloured dots represent data, bars denote 95% confidence intervals. Coloured lines and shaded areas represent the median and 95% credible interval of the posterior. **(B)** Predicted versus observed proportions.



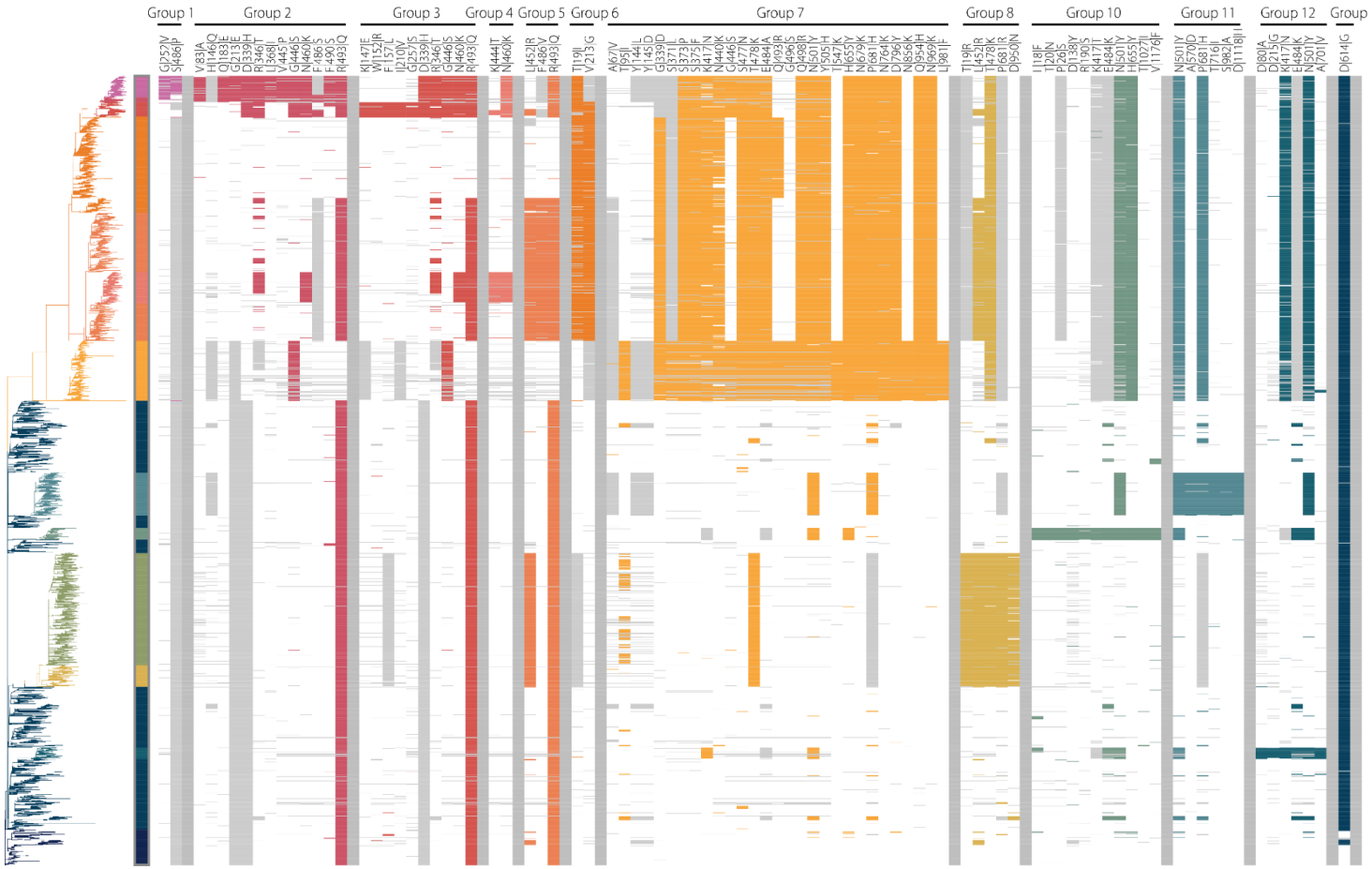
Supplementary Figure 7: Fitness model fits for all lineages of *M. tuberculosis.*

(A) Fits of the proportion of all the *M. tuberculosis* lineages. Coloured dots represent data, bars denote 95% confidence intervals. Coloured lines and shaded areas represent the median and 95% credible interval of the posterior. **(B)** Predicted versus observed proportion



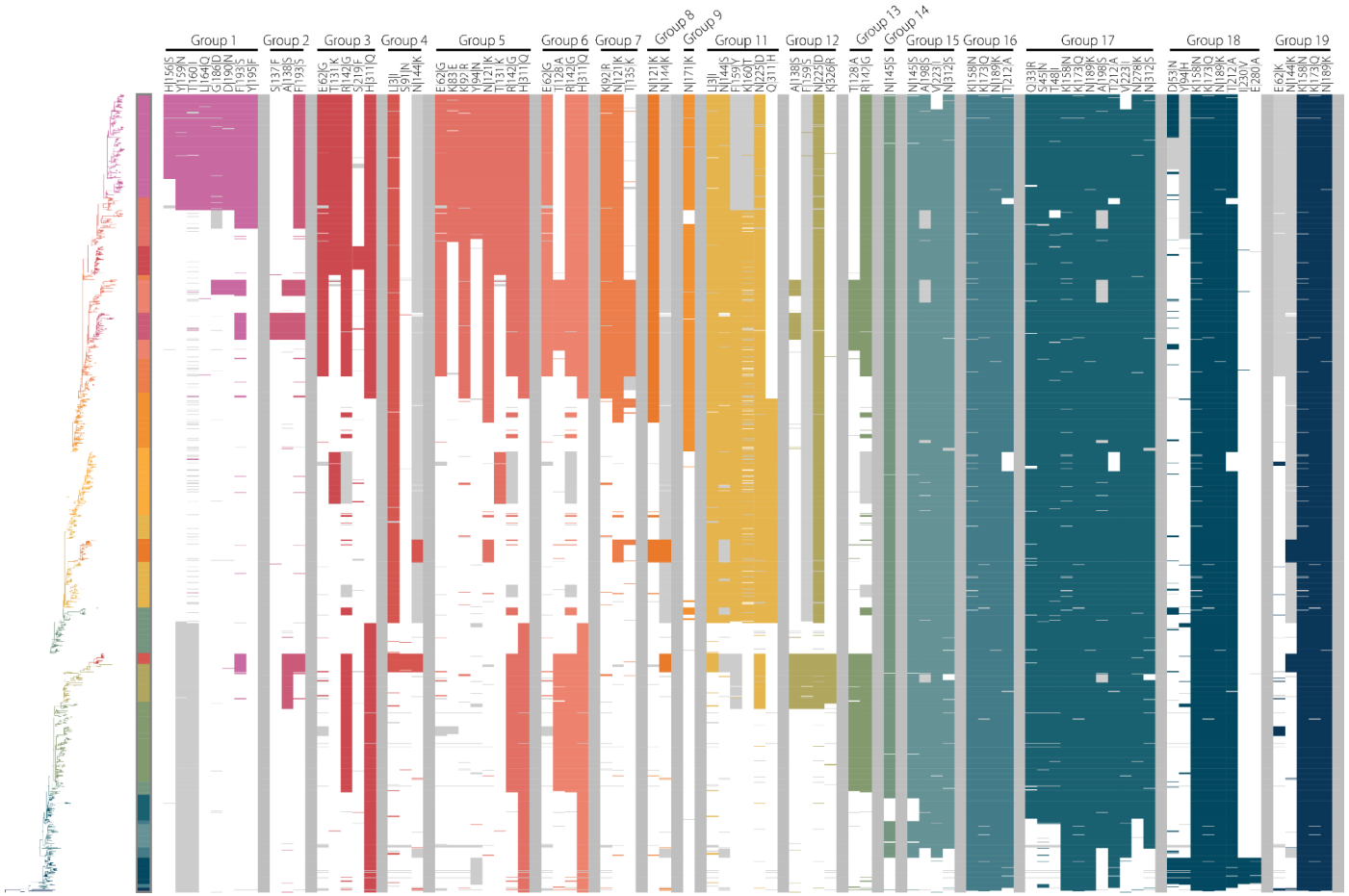
Supplementary Figure 8: Proportion of mutations that are defining the lineages of SARS-CoV-2 worldwide, by ORFs

Additionally to Figure 4E, we plot the proportion of amino acid substitutions that are lineage-defining within SARS-CoV-2 ORFs, and the Receptor Binding Domain (RBD).



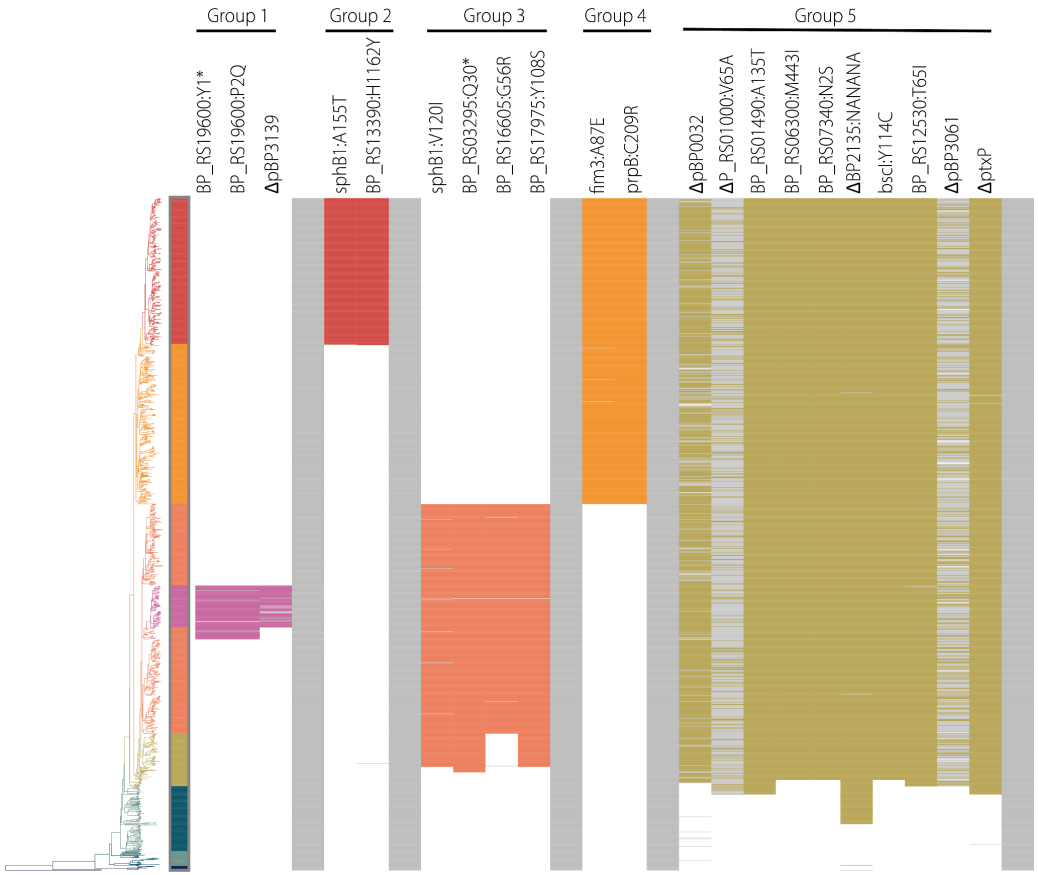
Supplementary Figure 9: Phylogenetic tree and mutations in the spike protein that are defining lineages in the global SARS-CoV-2 dataset

We present the SARS-CoV-2 time-resolved tree (left), together with the mutations that we found to be defining its lineages (right). Colours represent the different lineages. Each column on the right displays one mutation, with its name at the top. Colours denote isolates that are carrying the labelled mutation, white denotes the absence of that mutation (although isolates could have other mutations at this position), grey denotes an unknown amino acid. Some mutations (e.g., T478K or N501Y) are defining multiple lineages and are therefore plotted twice. The list of lineage-defining mutations can be found in Data File S5.



Supplementary Figure 10: Phylogenetic tree and mutations in the HA1 subunit that are defining lineages in the global H3N2 dataset

We present the H3N2 time-resolved tree (left), together with the mutations that we found to be defining its lineages (right). Colours represent the different lineages. Each column on the right displays one mutation, with its name at the top. Colours denote isolates that are carrying the labelled mutation, white denotes the absence of that mutation (although isolates could have other mutations at this position), grey denotes an unknown amino acid. Some mutations (e.g., N144K or F193S) are defining multiple lineages and are therefore plotted twice. The list of lineage-defining mutations can be found in Data File S6.

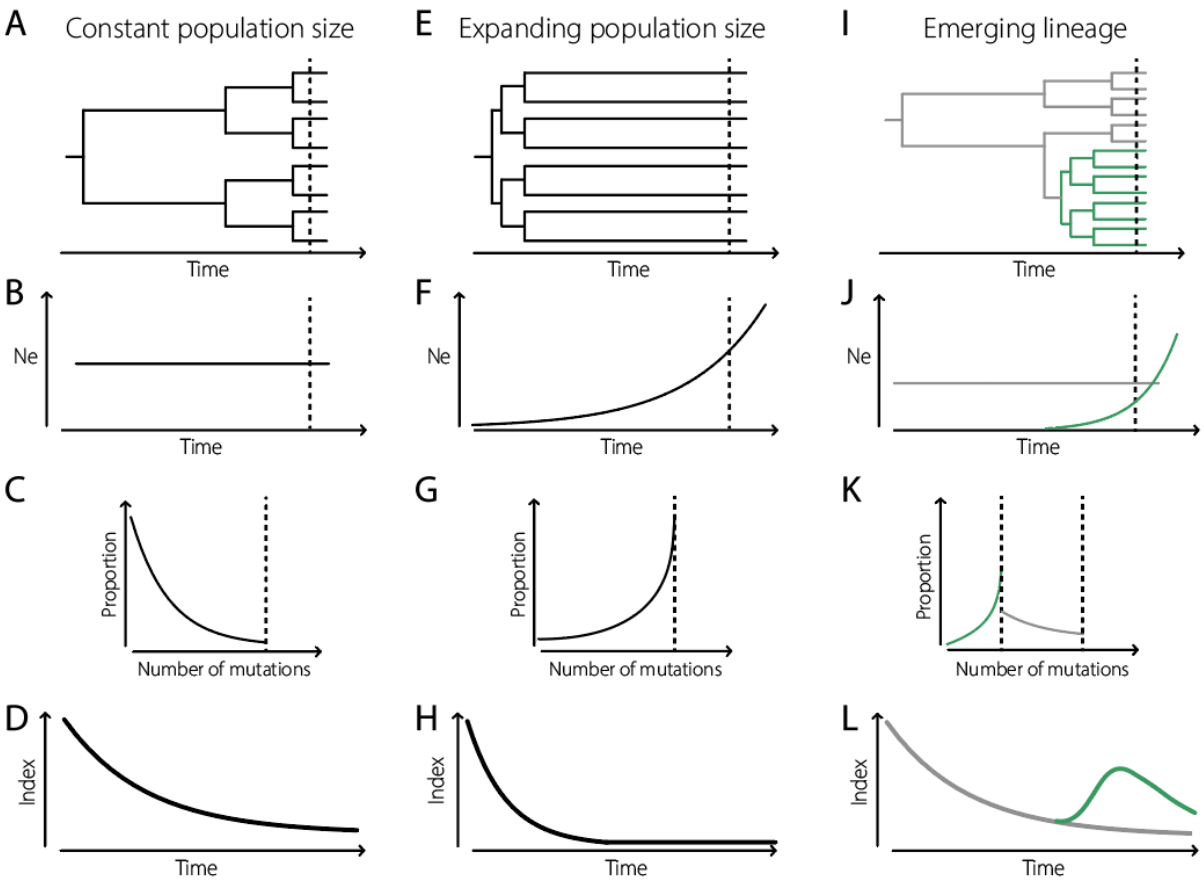


Supplementary Figure 11: Phylogenetic tree and mutations defining lineages in the *B. pertussis* dataset from in France

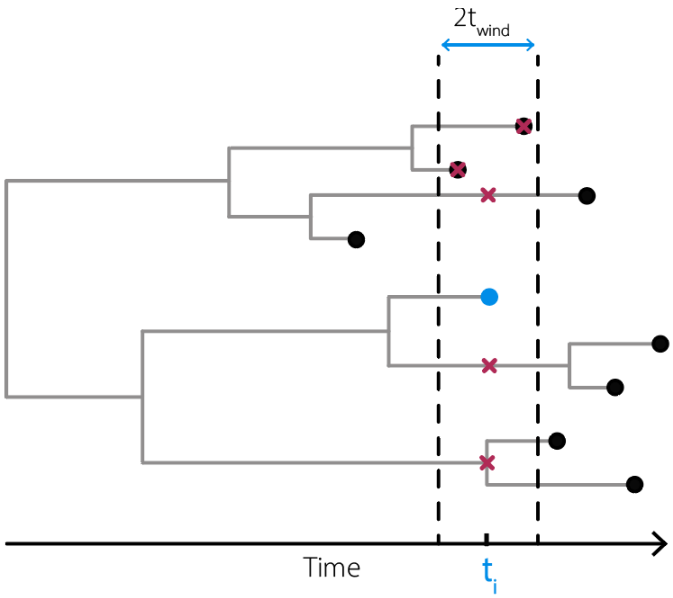
We present the *B. pertussis* time-resolved tree (left), together with the mutations that we found to be defining its lineages (right). Colours represent the different lineages. Each column on the right displays one mutation, with its name at the top. Colours denote isolates that are carrying the labelled mutation, white denotes the absence of that mutation (although isolates could have other mutations at this position), grey denotes an unknown nucleotide. The list of lineage-defining mutations can be found in Data File S7.

Supplementary Figure 12: Phylogenetic tree and mutations defining lineages 1 and 2 in the *M. tuberculosis* dataset from in Samara, Russia

We present the *M. tuberculosis* time-resolved tree (left), together with the mutations that we found to be defining the lineages 1 and 2 (right). Colours represent the different lineages. Each column on the right displays one mutation, with its name at the top. Colours denote isolates that are carrying the labelled mutation, white denotes the absence of that mutation (although isolates could have other mutations at this position). Some mutations (e.g., rpoB:S450L or katG:S315T) are defining both lineages and are therefore plotted twice. The list of lineage-defining mutations can be found in Data File S8.

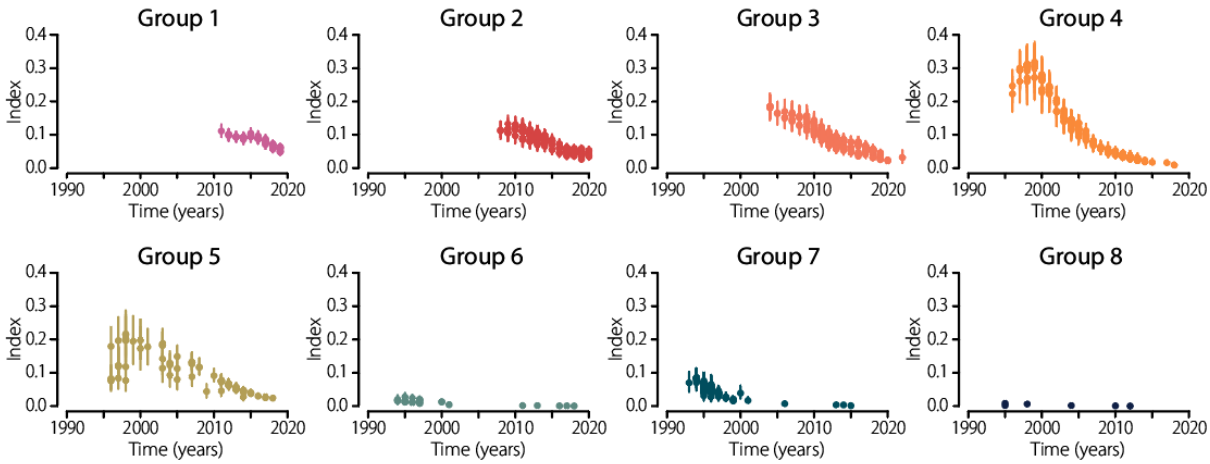


Supplementary Figure 13: Population history, pairwise distance distribution and index dynamics. **(A-D)** Constant effective population size. **(E-H)** Exponential population size. (A and B are inspired by Volz and colleagues, 2013[66]) **(I-L)** Case of an emerging, exponentially growing, lineage in a population of constant effective size.



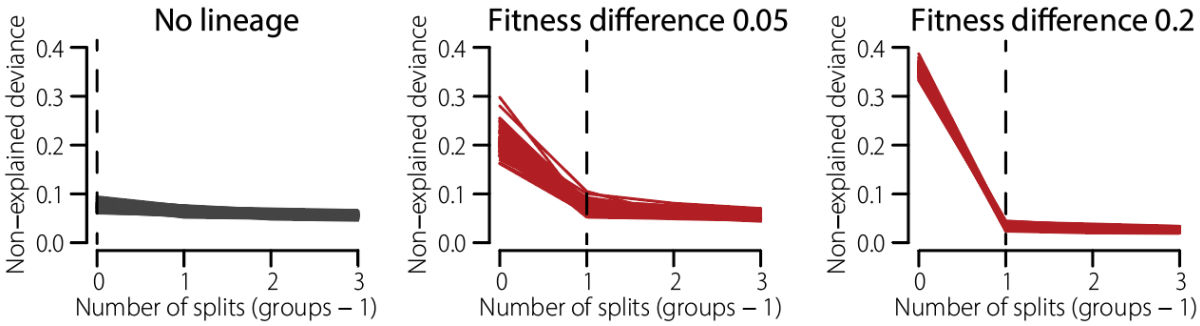
Supplementary Figure 14: Schematic of the notations used to compute the index on a timed-tree with sequences sampled through time.

The blue dot denotes the sequence of interest i, with t_i its sampling time. The dashed lines represent the window $[t_i - t_{wind};\ t_i + t_{wind}]$. All the nodes that fall within this window are considered to be circulating at the same time as i. These nodes are denotes by red crosses on the figure.



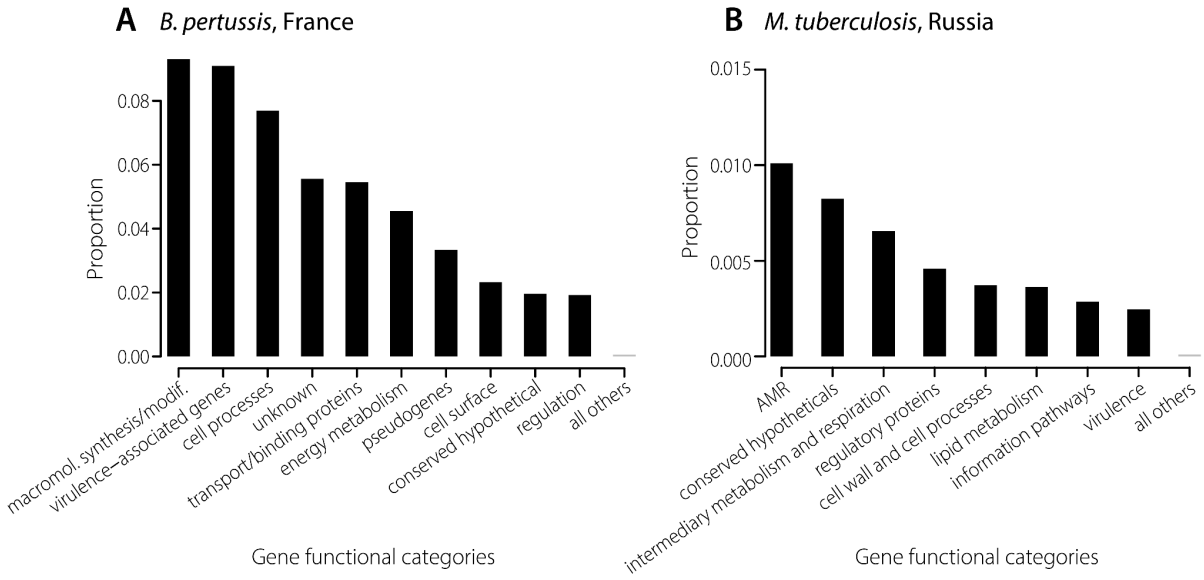
Supplementary Figure 15: Sensitivity analysis: *B. pertussis Index* dynamics over the posterior density of trees

We present the *Index* dynamics computed over 3000 trees from the BEAST posterior of *B. pertussis*. Dots and bars denote the median and 95% credible interval of the *Index* values. We plot only the *Index* of tips as we can summarise their values over the whole posterior.



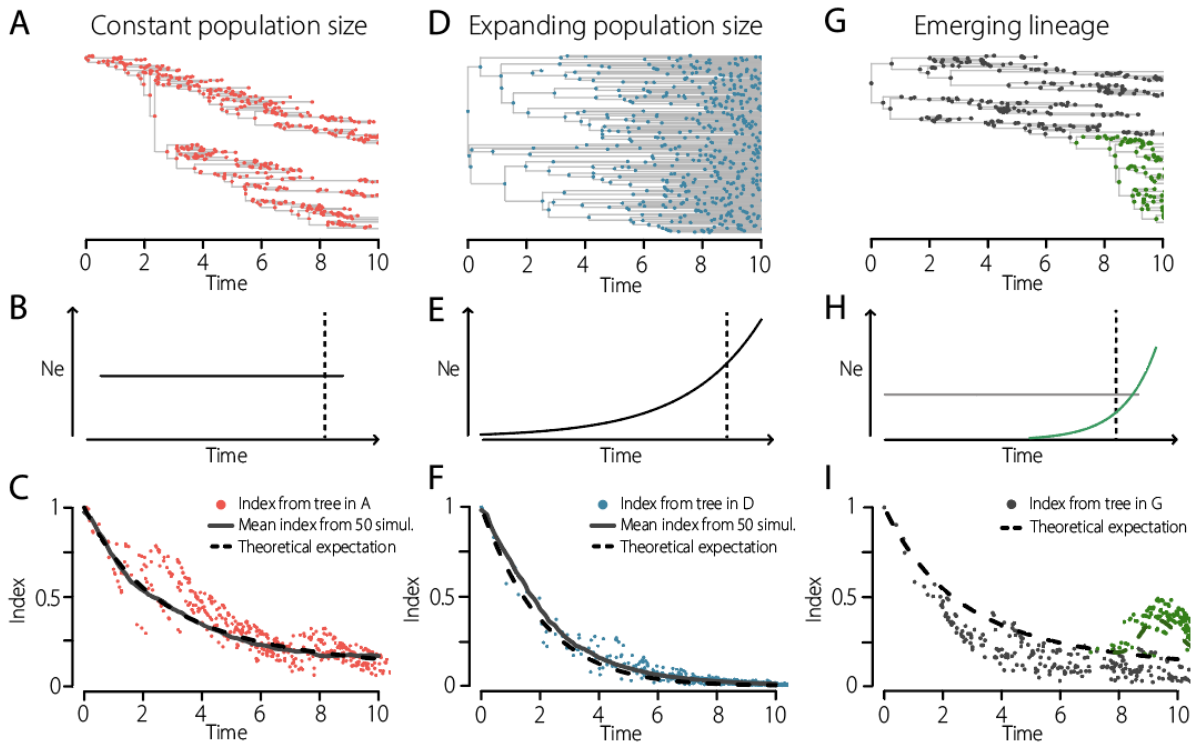
Supplementary Figure 16: Simulation study: non-explained deviance as a function of the number of groups in the lineage detection algorithm.

We simulated trees with or without an emerging lineage - from left to right: no lineage, fitness difference of 0.05/time unit, and fitness difference of 0.2/time unit. We present here the proportion of non-explained deviance by the models with different numbers of groups. Each simulation is plotted as a line (100 lines per scenario). The colour of the line indicates if a lineage was detected (grey: no lineage, red: one lineage detected). Lineages are detected if the non-explained deviance continues to decrease. Dashed lines denote the number of groups chosen.

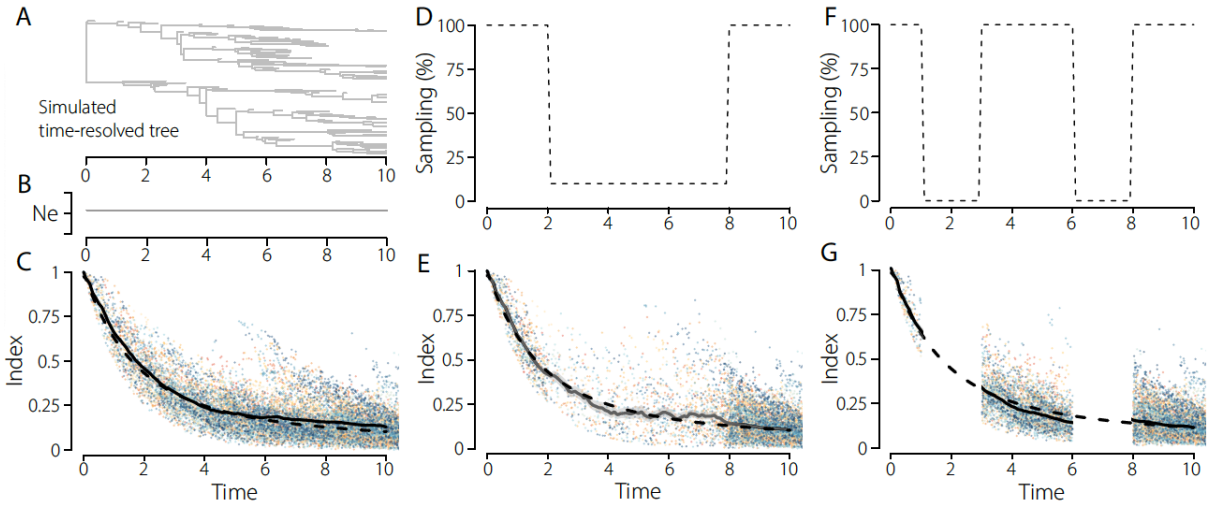


Supplementary Figure 17: Proportion of synonymous mutations that are lineage-defining, by gene functional categories, for *B. pertussis* and *M. tuberculosis*

Similarly to Figure 4K-L, we plot the proportion of synonymous mutations that are lineage-defining within each functional category, for (A) *B. pertussis* and (B) *M. tuberculosis* [35,36]. For *M. tuberculosis*, we only considered the most recent lineages 1 and 2. As expected, we find no statistical differences, as opposed to Figure 4K-L.



Supplementary Figure 18: Illustration of the index behaviour in different population histories. Similarly to Figure S21, we illustrate here the behaviour of the index. In each case, we simulate trees and compute the index on them. **(A-C)** Constant population size. Simulated time-resolved tree, under a birth-death model with equal probability of birth and death, i.e, constant population size on average. (B) Effective population size used in the simulation. (C) Index through time. **(D-F)** Exponential population size. **(G-I)** Case of an emerging, exponentially growing, lineage in a population of constant effective size. Colours denote each simulation. Dashed lines: expected dynamics given equations in the Methods. Solid lines: mean over the 50 simulations.



Supplementary Figure 19: Robustness to sampling schemes, from simulation study.
We assess the robustness of the index computation to sampling intensity. **(A-C)** Simulations with no sampling bias. 50 simulations were performed. The tree in A represents one simulation. B represents the effective population size trend: constant. **(D-E)** For each simulation, only 10% of the sequences from year 2-8 were used to compute the index. **(F-G)** No sequences from years 1-3 or 6-8 were used to compute the index. In C, E and G, colours denote each simulation. Dashed lines: expected dynamics given equations in the Methods. Solid lines: mean over the 50 simulations, for the different sampling biases.

Supplementary Tables

Pathogen	SARS-CoV-2	Influenza H3N2	*B. pertussis*	*M. tuberculosis*
Genome length	29903	1701	4086189	4411532
Substitution rate (substitution per site per year)	$8.1\ 10^{-4}$	$3.82\ 10^{-3}$	$2.5\ 10^{-7}$	$4.6\ 10^{-8}$
Time scale (years)	0.15	0.4	2	30
Kernel bandwidth *b*	0.91	0.87	0.84	0.94

Supplementary Table 1: Genome lengths, substitution rates, timescales and bandwidths used in this study.

We present the list of the different parameters that we use to compute the index. The function "*index.bandwidth()*" in the GitHub library allows to compute the kernel bandwidth given a genome length, a substitution rate and a timescale. The kernel bandwidth is dimensionless.

VOCs			Mu	Lambda	Iota	Eta	Epsilon	Kappa	EU1	Beta	Alpha	Gamma	Delta	Omicron								
Nextclade			21H	21G	21F	21D	21C	21B	20E	20H	20I	20J	21A-I-J	21K	22C	22A	21L	22B	22E	22D	22F	23A
Pango		wt	B.1.621	C.37	B.1.526	B.1.525	B.1.427	B.1.617.1	B.1.177	B.1.351	B.1.1.7	P.1	B.1.617.2	BA.1	BA.2.12.1	BA.4	~BA.2	BA.5	BQ.1	BA.2.75	XBB	XBB.1.5
Automatic clades	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	149
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	243	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	764	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	63	103	600	0	0	0	0	0
	7	3	0	0	0	0	0	0	0	0	0	0	0	471	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	172	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	889	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	339	0	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0
	13	1661	27	13	5	35	11	5	57	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	288	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Supplementary Table 2: SARS-CoV-2 contingency table comparing the automatic lineages to those previously identified.

Numbers indicate the counts of tips and internal nodes belonging to each category.

Global clade		wt	3C	3C.2a4	3C.2	3C.3	3C.3b	3C.3a	3C.2a	3C.2a2	3C.2a1a	3C.2a1	3C.2a3	3C.2a1b.1	3C.2a1b.1a	3C.3a1	3C.2a1b.2a	2d	2c	1	3C.2a1b.2	3C.2a1b.2b	2	3C.2a1b.1b	2b	2a
Automatic clades	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	51	311
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	63	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	11	21	67	28	0	14	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	139	55	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	60	0	83	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	94	0	31	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	59	189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	257	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	124	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	261	27	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0	57	25	76	36	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	53	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	17	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	19	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Supplementary Table 3: H3N2 contingency table comparing the automatic lineages to those previously identified.

Numbers indicate the counts of tips and internal nodes belonging to each category.

Genotypes		ptxP3/fim3-2	ptxP3/fim3-1	ptxP1/fim3-1
Automatic clades	**1**	0	155	0
	2	529	0	0
	3	0	682	0
	4	584	0	0
	5	0	197	0
	6	0	0	56
	7	0	30	207
	8	0	0	15

Supplementary Table 4: *B. pertussis* contingency table comparing the automatic lineages to those previously identified.
Numbers indicate the counts of tips and internal nodes belonging to each category.

Lineages		CAS	EAI	Haarlem	E-A	Ural	LAM	Beijing	Clade A	Clade B
Automatic clades	**1**	0	0	0	0	0	0	0	0	515
	2	0	0	0	0	0	0	0	227	0
	3	0	0	0	0	0	0	513	4	0
	4	0	0	0	0	0	143	0	0	0
	5	0	0	0	0	131	0	0	0	0
	6	0	0	0	37	0	0	0	0	0
	7	0	0	0	45	0	0	0	0	0
	8	0	0	0	140	0	0	0	0	0
	9	0	0	0	12	62	32	0	0	0
	10	0	0	81	0	0	0	0	0	0
	11	0	0	0	36	0	0	0	0	0
	12	6	7	0	2	0	0	0	0	0

Supplementary Table 5: *M. tuberculosis* contingency table comparing the automatic lineages to those previously identified.
Numbers indicate the counts of tips and internal nodes belonging to each category. LAM denotes the Latin American-Mediterranean lineage, E-A the Euro-American lineage, EAI the East African Indian lineage and CAS the Central Asian Strain lineage.

Supplementary Table 6: SARS-CoV-2 lineage-defining mutations.
For each lineage, the set of defining amino-acid substitutions is given. Mutations are separated in three groups based on their locations in the genome, for convenience: "Spike", "E, N and M"and "ORFs". This table is provided as a csv file.

Supplementary Table 7: H3N2 lineage-defining mutations.
For each lineage, the set of defining amino-acid substitutions in the HA protein is given. Mutations are separated in three groups based on their locations in the protein, for convenience: "Sig", "HA1"and "HA2". This table is provided as a csv file.

Supplementary Table 8: *Bordetella pertussis* lineage-defining mutations in France.
This table lists all the *Bordetella pertussis* lineage-defining non-synonymous substitutions and mutations in the promoter regions. Information on the locus tag, gene functional category, gene product and gene name is included when available. This table is provided as a csv file.

Supplementary Table 9: *Mycobacterium tuberculosis* lineage-defining mutations in Samara, Russia.
This table lists all the *Mycobacterium tuberculosis* lineage-defining non-synonymous substitutions and mutations in the promoter regions. Information on the gene id, gene functional category, gene product and gene name is included when available. This table is provided as a csv file.

Supplementary Table 10: Isolates of SARS-CoV-2
This table lists all the 3129 whole genome SARS-CoV-2 sequences used in this study. For each sequence, we list its identifier, GISAID accession number, collection date, country of isolation and Nextstrain clade. This table is provided as a csv file.

Supplementary Table 11: Isolates of H3N2
This table lists all the 1476 H3N2 Hemagglutinin sequences used in this study. For each sequence, we list its identifier, GISAID accession number, collection date, location of isolation and global clade. This table is provided as a csv file.

Supplementary Table 12: Isolates of *Bordetella pertussis*
This table lists all the 1248 whole genome *B. pertussis* sequences from France[3,38–41] and the Tohama I reference genome. For each sequence, we list its identifier, accession number, collection year, country of isolation and genotype. This table is provided as a csv file.

Supplementary Table 13: Isolates of *Mycobacterium tuberculosis*
This table lists all the 997 whole genome *M. tuberculosis* sequences from Samara, Russia[20] and the H37Rv reference genome. For each sequence, we list its accession number, collection year, country of isolation and clade. This table is provided as a csv file.