Molecular Biology and Evolution, 2025, **42**, 1–13
https://doi.org/10.1093/molbev/msaf029
Advance access publication 3 February 2025
Methods

MBE OXFORD

What Do We Gain When Tolerating Loss? The Information Bottleneck Wrings Out Recombination

Apurva Narechania [1,2,*] **Dean Bobo** [1,3] **Rob DeSalle** [1] **Barun Mathema** [4] **Barry Kreiswirth** [5] **Paul J. Planet** [1,6,7,*]

[1]Institute for Comparative Genomics, American Museum of Natural History, New York, NY, USA
[2]Section for Hologenomics, The Globe Institute, University of Copenhagen, Copenhagen, Denmark
[3]Department of Ecology, Evolution, and Environmental Biology, Columbia University, New York, NY, USA
[4]Department of Epidemiology, Mailman School of Public Health, Columbia University, New York, NY, USA
[5]Center for Discovery and Innovation, Hackensack Meridian Health, Nutley, NJ, USA
[6]Division of Infectious Diseases, Children's Hospital of Philadelphia, Philadelphia, PA, USA
[7]Department of Pediatrics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA

***Corresponding authors:** E-mails: anarechania@amnh.org; planetp@chop.edu.
Associate editor: Daniel Falush

Abstract

Most microbes have the capacity to acquire genetic material from their environment. Recombination of foreign DNA yields genomes that are, at least in part, incongruent with the vertical history of their species. Dominant approaches for detecting these transfers are phylogenetic, requiring a painstaking series of analyses including alignment and tree reconstruction. But these methods do not scale. Here, we propose an unsupervised, alignment-free, and tree-free technique based on the sequential information bottleneck, an optimization procedure designed to extract some portion of relevant information from 1 random variable conditioned on another. In our case, this joint probability distribution tabulates occurrence counts of *k*-mers against their genomes of origin with the expectation that recombination will create a strong signal that unifies certain sets of co-occurring *k*-mers. We conceptualize the technique as a rate–distortion problem, measuring distortion in the relevance information as *k*-mers are compressed into clusters based on their co-occurrence in the source genomes. The result is fast, model-free, lossy compression of *k*-mers into learned groups of shared genome sequence, differentiating recombined elements from the vertically inherited core. We show that the technique yields a new recombination measure based purely on information, divorced from any biases and limitations inherent to alignment and phylogeny.

Keywords: microbial evolution, recombination, information theory

Significance

The information bottleneck (IB), a lossy compression technique borrowed from the information theoretic and natural language processing literature, is well suited to detecting evolutionary patterns in sets of co-occurring *k*-mers. Here, we show that we can detect simulated and real recombination events while highlighting a core set of *k*-mers that comprise the vertically inherited portion of any set of genomes. Moreover, genome compressibility offers a new way to compare clades across the microbial tree of life. In our application, the bottleneck is informed by genome origin, our relevance variable. But the technique is general. The IB can be used for any biological contingency matrix where the goal is to learn groups from unstructured data.

Introduction

Microbial genomes have accumulated at an unprecedented rate (Kyrpides et al. 2014; Nayfach et al. 2021). Most work on their molecular evolution is grounded in sequence alignment and phylogenetic tree reconstruction, computationally expensive techniques ill-equipped to handle sequence volume in a read streaming era (Erlich 2015). The evolution of microbes is particularly challenging because recombined elements contribute signal unrelated to vertical descent (Fraser et al. 2007). Current techniques require alignment of reads across a reference genome (Croucher et al. 2015; Didelot and Wilson 2015), whole genome alignment (Darling et al. 2004), and/or phylogenomic methods based on orthology (Chiu et al. 2006; Zhao et al. 2012; Page et al. 2015). Each requires careful curation and deliberate sampling to limit data to reasonable scales. Reference genomes are a precursor for many such analyses, but references can bias results by confining conclusions to a single well-understood sequence. For larger, unbiased data sets that include as much natural variation as possible, these approaches are ineffective. We need tools that can tolerate information loss without sacrificing knowledge of key evolutionary events.

Lossy compression, where an individual or algorithm makes decisions about which data are important (or relevant) from a

Received: November 30, 2023. **Revised:** December 3, 2024. **Accepted:** January 14, 2025

large body of information, may offer a solution (Marzen and DeDeo 2017). To do this in a principled way, the relevance of a given data set can be measured as information retained about some other correlated variable. For example, in unsupervised natural language processing (NLP), large corpora of texts are distilled to a few topics that reflect overall themes by comparing patterns of co-occurring words in the source texts. In topic modeling of this sort, the texts themselves are the relevance variable. The goal is to cluster the word distribution with respect to the documents from which they arise. This idea was first described by Tishby et al. (2000) as the information bottleneck (IB). It was premised on rate distortion, Shannon's theory of lossy compression which yoked signal distortion to the rate at which that signal can be encoded (Shannon 1948). The IB's primary innovation was the use of a relevance variable to quantify this distortion and instill it with meaning. Topic modeling was one of this technique's first applications.

Topic modeling has become an important part of the NLP literature with several wider applications to unsupervised machine learning. The dominant technique in the field is latent Dirichlet allocation (LDA) (Blei 2003), a probabilistic method that, like the IB, considers each document a mixture of topics. Some authors have applied this idea to whole genomes (Chen et al. 2010; La Rosa et al. 2015; Liu et al. 2016), and since the publication of STRUCTURE, LDA has become foundational in the genetics literature (Pritchard et al. 2000). Despite LDA's popularity and success, a number of authors have shown that unbalanced sampling can lead to erroneous or missed population assignments (Wang 2017). LDA also makes a number of statistical assumptions including the assignment of hyperparameters and a Dirichlet prior (Wallach et al. 2009). In contrast, the IB is model free and unaffected by size sample bias.

Genomes are living documents that can be sliced into words of arbitrary size (Sims et al. 2009; Cong et al. 2016). In a genomic context, where words are *k*-mers and documents are their genomes of origin, we predict that IB-derived topics will represent compressed, co-occurring groups of *k*-mers driven by shared ancestry. *K*-mer clusters might describe co-linear blocks distributed across the genome as fragments inherited simultaneously, or may be common to all genomes, providing a simple, operational definition of a genomic core. The process requires only 2 key parameters: the *k*-mer size and the number of clusters expected. In the NLP topic modeling analogy, the core genome of a species could be considered the set of concepts common to every document in a library, while recombined regions are like themes or ideas restricted only to certain shelves.

Here, we apply the IB to tens of thousands of microbial genomes. Remarkably, our approach identifies recombination tracts without resorting to reference-based analyses. We make no attempt to model evolution, annotate genes, build alignments, or reconstruct trees. Further, squeezing *k*-mers through the bottleneck quantifies distortion which we repurpose to measure horizontal signal.

Theory and Implementation

We show a schematic of our procedure in Fig. 1. Consider a set of genomes (G) chopped into *k*-mers (K). The IB is designed to compress K under the constraint of G into a set number of clusters, C (Tishby et al. 2000). This compression is lossy (Fig. 1a). The IB extends Shannon's theory of rate distortion by guiding it with an additional, orienting variable. In our case, this variable is genomes of origin, G. This concept of a relevance variable assigns value to distortion resulting in principled lossy compression.

In Fig. 1b, we outline the idea as a thought experiment. Consider 3 microbial genomes, G1 to G3. G3 has a large recombination region unique to its lineage. We digest these genomes in silico and tabulate *k*-mers (K) with respect to their genomes of origin (G) as a joint probability distribution. Our goal is to reduce the dimensionality of K by grouping the *k*-mers into meaningful clusters, C. In this case, we define just 2 clusters. In a rate distortion sense, C can be considered the channel capacity. For the technique to work, the 2 variables in our joint distribution $p(k,g)$ must be nonindependent or, more precisely, must have positive mutual information, $I(K,G)$:

$$I(K, G) = \sum_K \sum_G p(k, g) \log \frac{p(g|k)}{p(g)}$$

where $p(k,g)$ is the joint probability of K and G, $p(g|k)$ is the conditional probability of G given K, and $p(g)$ is the marginal probability of G. C is now a meaningful compression of the data, maximizing the mutual information between the clusters and genomes, I(C;G), while minimizing the mutual information between the *k*-mers and the clusters, I(C;K). The IB is a classic optimization problem.

In the scenario shown in Fig. 1b, *k*-mers from the core genome sort into cluster 1 (C1), while *k*-mers from the recombined region populate cluster 2 (C2). When these *k*-mers are mapped back onto the source genomes, we confirm that the clusters reflect that the position and length of the recombined region has been learned through compression.

This formulation balances the compactness of K, with the erosion of information about G. The choice of C determines the optimization landscape. If C equals 1, *k*-mers are clumped into just 1 cluster, the ultimate compression. If C equals K, every *k*-mer is its own cluster, preserving all relevant information. Of course, collapsing all *k*-mers into 1 cluster is overly reductive, and assigning each *k*-mer to its own cluster is meaningless. The IB negotiates these 2 extremes. In NLP, the result is a set of clusters that coalesce into topics over a body of literature (Pereira et al. 1993). In genomics, the process identifies co-occurring and/or spatially co-located *k*-mers with distinct biological and/or evolutionary meaning.

Our optimization function has an exact, optimal solution. The most surprising outcome of this solution is that the relative entropy, or Kullback–Liebler divergence (Kullback and Leibler 1951), emerges as the distortion measure for the IB. The relative entropy is a fundamental quantity in information theory, and in our IB context, it measures the distortion of compressed *k*-mers:

$$D_{KL} = \sum_G p(g|k) \log \frac{P(g|k)}{P(g|c)}$$

where $p(g|k)$ is the conditional probability of G given K, and $p(g|c)$ is the conditional probability of G given C. We set the number of clusters and implement a sequential procedure. From an initial random distribution of all *k*-mers across our clusters, we draw 1 *k*-mer out and represent it as a singleton. Using greedy optimization, we merge this singleton into one of the existing bulk clusters. Slonim's sequential IB (SIB) (Slonim et al. 2002) employs the Jensen–Shannon divergence (Lin 1991) in the cost of merging a *k*-mer, k, into a cluster, c:

$$d(k, c) = (p(k) + p(c)) * D_{JS}(p(g|k), p(g|c))$$

where the sum of the marginal distributions $p(k)$ and $p(c)$ weights the Jensen-Shannon divergence between the conditional distributions $p(g|k)$ and $p(g|c)$. A *k*-mer will join a new cluster only if its new address reduces the total distortion. Otherwise, it remains in its existing cluster. With respect to our initial

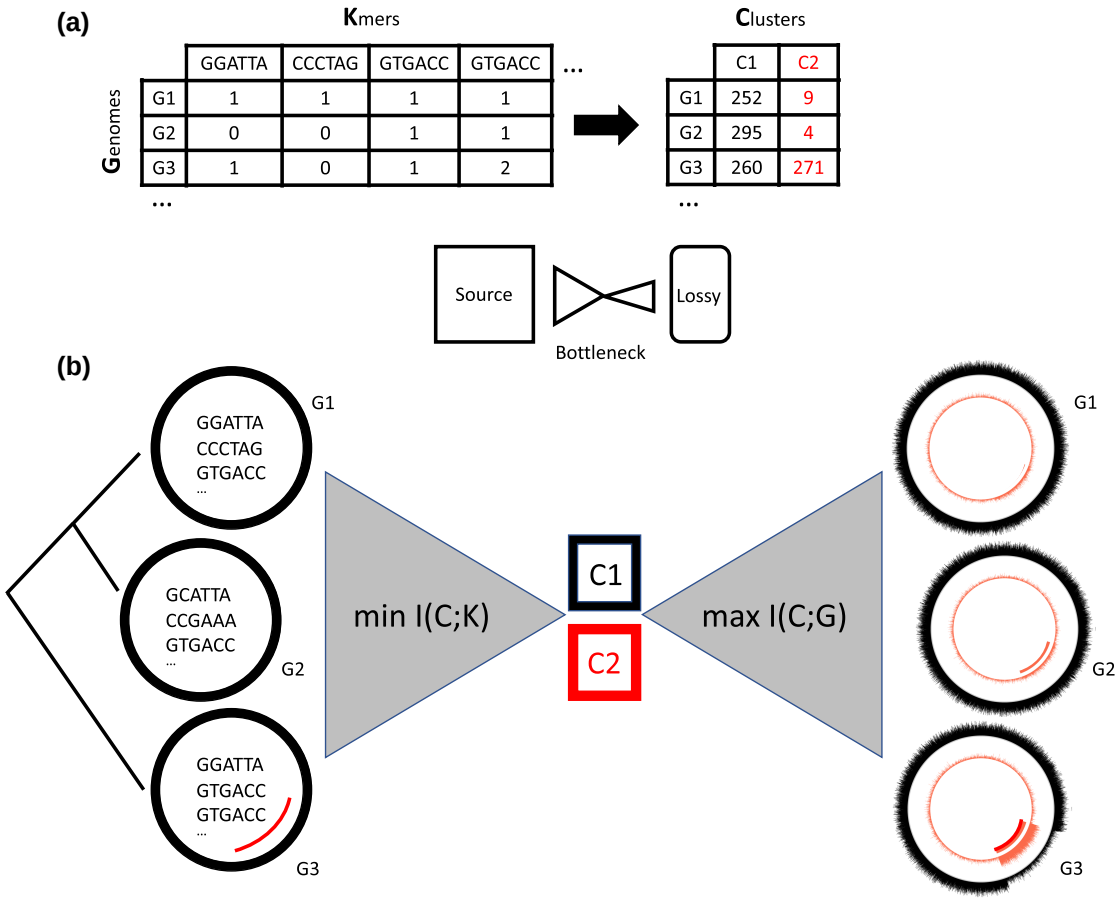

Fig. 1. The IB. In the IB, a *k*-mer distribution, *K*, is compressed into *C* while retaining as much information as possible about a correlated relevance variable, *G*. The joint distribution, *p(k,g)*, has positive mutual information, and the goal of the IB is to capture the most relevant information at interpretive scale. Though the technique is lossy, a) it puts recombination events into sharp relief b). We show 3 input genomes, one of which contains a substantial recombination region. The numbers in the table are hypothetical. As the mutual information between *C* and *K* is minimized, the mutual information between *C* and *G* is maximized. At optimality, *C* is presumed to be a lossy but adequate model of *K*. This model identifies a set of core *k*-mers in C1 plotted here as the outer track on all 3 genomes, and the recombined region in C2, plotted as the inner track. The location of the recombination event is shown as precisely overlapping with the alignments of C2 *k*-mers on the G3 genome.

random conditions, this algorithm is guaranteed to converge to a local optimum. We mitigate the risk of getting trapped in local optima by testing several random initializations, and we reduce the overall computation by implementing a *k*-mer sketch (Ondov et al. 2016; Narechania et al. 2024).

Once the clusters stabilize, we quantify the information captured by calculating the normalized mutual information, $NMI = I(C;G)/I(K;G)$. Trivially, the NMI is equal to 1 when each *k*-mer occupies its own cluster and is 0 when all *k*-mers are mixed into 1. The curve traced between $C = 1$ and $C = x$ is called the relevance–compression curve (Still and Bialek 2004). The choice of C modulates the Shannon channel capacity, and the shape of this curve describes the compressibility of the data. We also use the shape of the curve to guide the optimum number of clusters to model. But the most important aspect of the SIB, and the reason we chose it for this work, is that its distortion, as measured by NMI, is a proxy for horizontal signal.

Results and Discussion

The Bottleneck in Test: Simulated Recombination Events

We simulated 3 recombination rates (0.000005, 0.00001, and 0.000015) across 3 sets of 100 1 Mb genomes in SimBac (Brown et al. 2016). SimBac models homologous recombination. These events appear in black on the innermost track of the 3 circos plots in Fig. 2. We sketched 19-mers and applied the IB, modeling a range of different cluster sizes as shown in the relevance–compression curves. We use these curves to determine an appropriate number of clusters for a given data set. In each case shown here, the NMI rises quickly between cluster sizes of 2 and 15, elbowing thereafter. For these simulations, 15 clusters are therefore the bare minimum required to capture the easily modeled portion of the lossy compression. But the curves indicate gains even after this point. We settled on 50 clusters to describe these data sets. Though these curves are not definitive, they are a useful tool to choose a suitable number of clusters.

The number of genomes in a data set has a profound effect on the shape of the relevance–compression curves. In supplementary fig. S1, Supplementary Material online, we show data from 100 simulated sets each of 4, 10, 100, and 1000 1 Mb genomes. Regardless of the number of genomes in each experiment, the NMI increases with increased modeling (1 example for each shown in supplementary fig. S1a, Supplementary Material online). As expected, this increase is rapid and stark in the smaller data sets and slow and somewhat halting in the larger ones. Complex scenarios with more genomes and more evolutionary paths require a broader channel (more clusters) to transmit

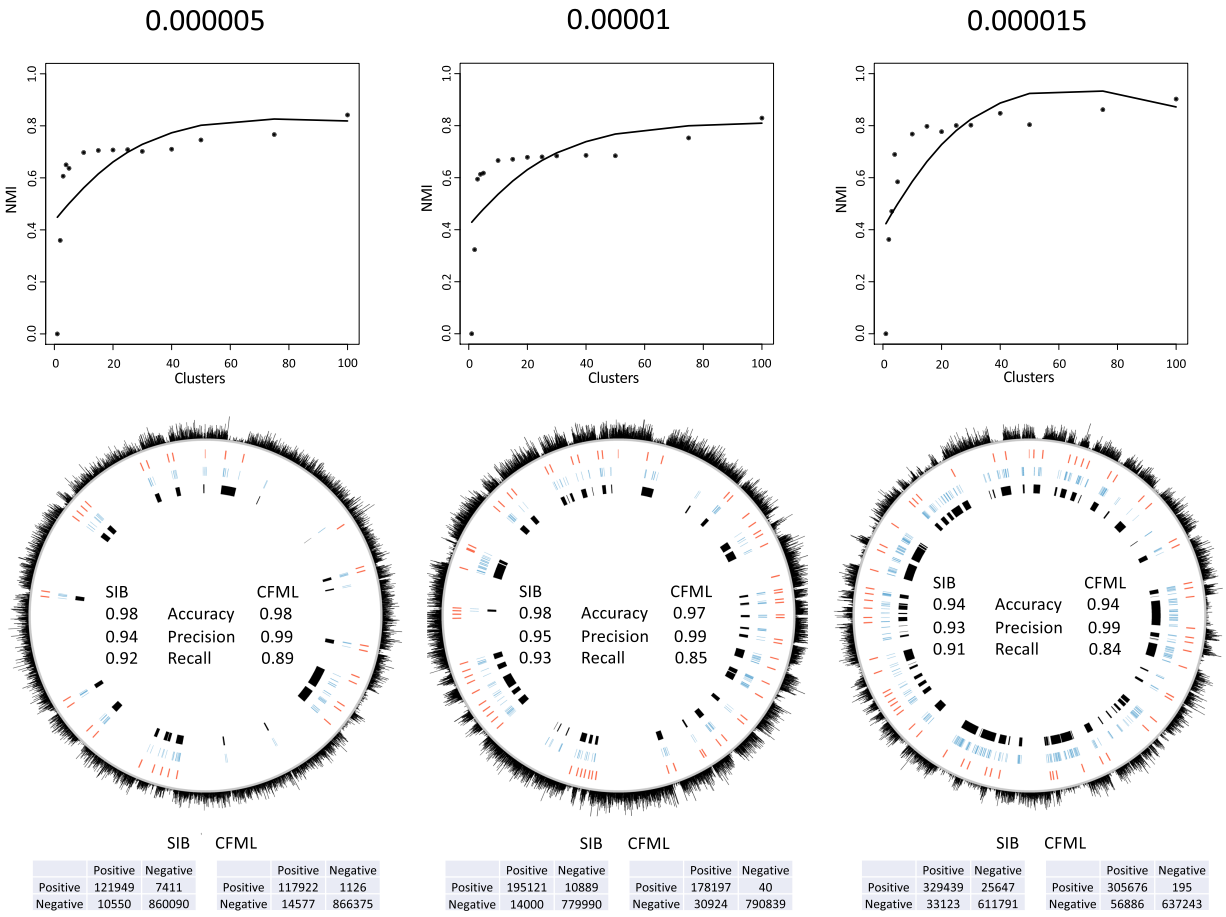

Fig. 2. Simulations. We simulated recombination events for rates of 0.000005, 0.00001, and 0.000015 across 100 1 Mb genomes. We calculated relevance–compression curves across a range of modeled clusters and fit a line to approximate the optimal number of clusters for each simulation. The recombination events are shown as blocks on the first, innermost track. IB-based change point calls for breakpoints in the 50-cluster run are shown on the third track, and ClonalFrameML-based intervals are shown in the second track. Confusion matrices and learning stats comparing the 2 techniques are shown embedded.

signal. More clusters reduce channel noise. The curve describing this process defines the limits of our compression. Theoretically, the space above the curve is unachievable, forming an upper bound. The relevance–compression curve therefore defines absolute limits on the quantity and quality of evolutionary information communicated as we sweep through a dilating channel. In supplementary fig. S1b and c, Supplementary Material online, we show how information loss in more complex compression problems is driven by the narrowing gap between *k*-mer frequency outside versus inside the simulated recombination events. With 4 genomes, discrimination is easy. With 1000, noise limits our resolution.

In the circos plots shown in both Fig. 1 and supplementary fig. S1, Supplementary Material online, the outermost track is a frequency plot of *k*-mers from each data set's core cluster aligned to an arbitrary genome from the set. The core emerges as a dense block of shared genome sequence interrupted by our simulated recombination events. *K*-mers that would otherwise occupy these gaps are sorted into other clusters. If a single ancestor sustains multiple transfer events, all *k*-mers from these events merge into a single cluster shared by the same subset of descendants. We show a simple example as a matrix of circos plots in supplementary fig. S2, Supplementary Material online. Here, 4 genomes experience 2 recombination events at different evolutionary points. The core appears as cluster 5. The other 4 clusters form 2 pairs, with each pair describing the donor and acceptor of each homologous event. In every case, plots of the bottleneck-defined core function almost as a photographic negative, highlighting the blank spaces as regions scrambled by horizontal signal.

In the examples shown, the pattern of events is evident by eye, but with more genomes and increased modeling (more clusters), automated inspection is key. We use a signal processing method based on change point detection to programmatically detect changes in *k*-mer frequency. We specifically employ the PELT algorithm (Killick et al. 2012) to model probabilities of change along the genome backbone. As shown in the track of red hashes (Fig. 1; supplementary fig. S1, Supplementary Material online), change point detection largely delimits the boundaries of our simulated recombination events. PELT's detection sensitivity is modulated by a penalty parameter that can vastly change the analysis. The Bayesian information criterion can be used to optimize this parameter. But in practice, change point optimization is a difficult problem often requiring a parameter sweep followed by manual inspection (Killick and Eckley 2014). A sweep through values for this parameter for the 0.00001 rate simulation is shown in supplementary fig. S3, Supplementary Material online. While this parameter can alter the change points derived from our primary frequency data, the *k*-mer frequency data itself almost always shows gaps in the core's continuity at recombination events. This is particularly evident in simulations with high recombination rates (0.000015 in Fig. 1) or simulations with many genomes (1000 on supplementary fig. S1, Supplementary Material online). In the 0.00001 rate example shown in Fig. 2,

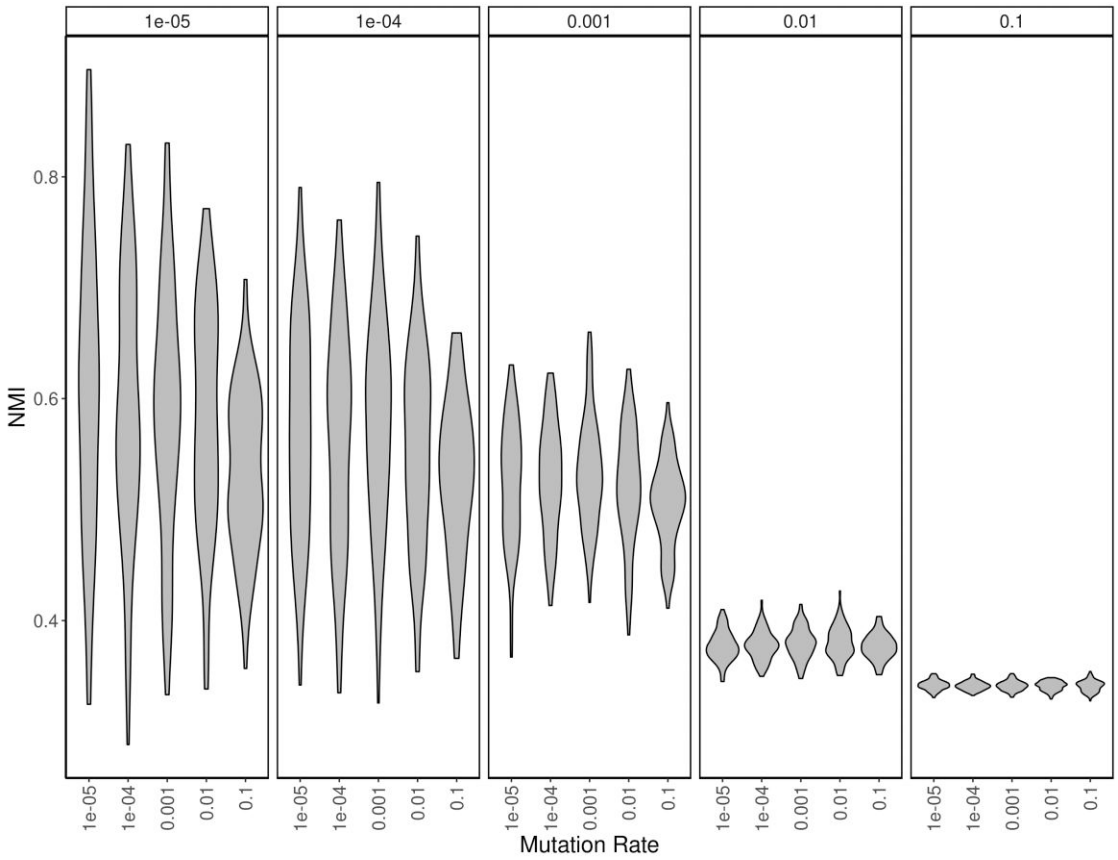

Fig. 3. Simulation across various rates of recombination and background mutation. For every recombination rate and mutation rate pair, we simulated 100 sets of 100 1 Mb genomes. NMI values are grouped by recombination rate (across the top) and refined by mutation rate (within each recombination rate box). As the recombination rate increases, the NMI decreases. The effect of background mutation rate is swamped by recombination, becoming a factor only when recombination rates are very low.

48 of the 50 recombination events have *k*-mer counts significantly lower than the rest of the core's background (Wilcoxon, $P < 0.05$). Our primary result, the frequency of *k*-mers in various clusters therefore contains very clear information on recombination events sometimes missed by PELT. Future work incorporating other signal processing techniques may improve automated detection.

We compare our approach to ClonalFrameML (Didelot and Wilson 2015), one of the leading recombination detection programs in microbial genomics. Unlike the IB, ClonalFrameML requires a reference, alignment of all remaining 99 genomes to that reference, tree reconstruction from the resulting character matrix, and recombination detection using a likelihood-based sliding window approach. ClonalFrameML calls are shown in the track of light blue hashes (Fig. 2). The program successfully identifies regions of recombination, sometimes picking up smaller regions that IB fails to detect at the shown change point sensitivity. But ClonalFrameML seems to subdivide events into smaller ones while the bottleneck aggregates them. Analysis of the absolute number of called events is therefore impractical, and we instead compare the 2 methods by the amount of recombined sequence *covered* by predictions. Both methods excel, performing substantially better across genomes with a lower recombination rate. But because ClonalFrameML overpredicts recombination breakpoints, the bottleneck's recall on total recombined bases is substantially better (see confusion matrices in Fig. 2). Both techniques are equally accurate, but ClonalFrameML is marginally more precise because of its ability to pick up the smallest recombination events.

The relevance–compression curve's responsiveness to evolutionary complexity opens the door to reconceptualizing recombination. Given a set of genomes, most researchers calculate R/theta (Guttman and Dykhuizen 1994), the ratio of the recombination rate over the mutation rate, to quantify a species' tendency to recombine. But R/theta's calculation is grueling, requiring alignment (either reference based or across whole genomes), tree reconstruction, and algorithms to isolate the clonal frame (Croucher et al. 2015; Didelot and Wilson 2015). The IB clearly pinpoints recombination events in space without resorting to any of these traditional procedures.

NMI is a corollary to this compression, but can NMI itself be an informative recombination metric? In Fig. 3, we show that for sets of 100 simulated genomes compressed into a fixed set of 5 clusters, NMI decreases with increased recombination rate. We also show that even very high background mutation rates have a negligible effect on NMI for all sets except those subject to vanishingly low recombination. Recombination seems to dominate distortion indicating that evolutionary complexity—not the noise of mutation—is most tied to information loss. At high rates of recombination, every base of a 1 Mb genome is likely scrambled. Under such flux, some sites recombine several times, eroding the core genome itself. This erosion is reflected in the steep decline in NMI with increasing recombination rate. The NMI's responsiveness to both recombination rates and mutation rates makes it an attractive alternative to arduous—and in some cases—impossible R/theta calculations. As expected, we capture more information if we allow more clusters, but over-modeling in

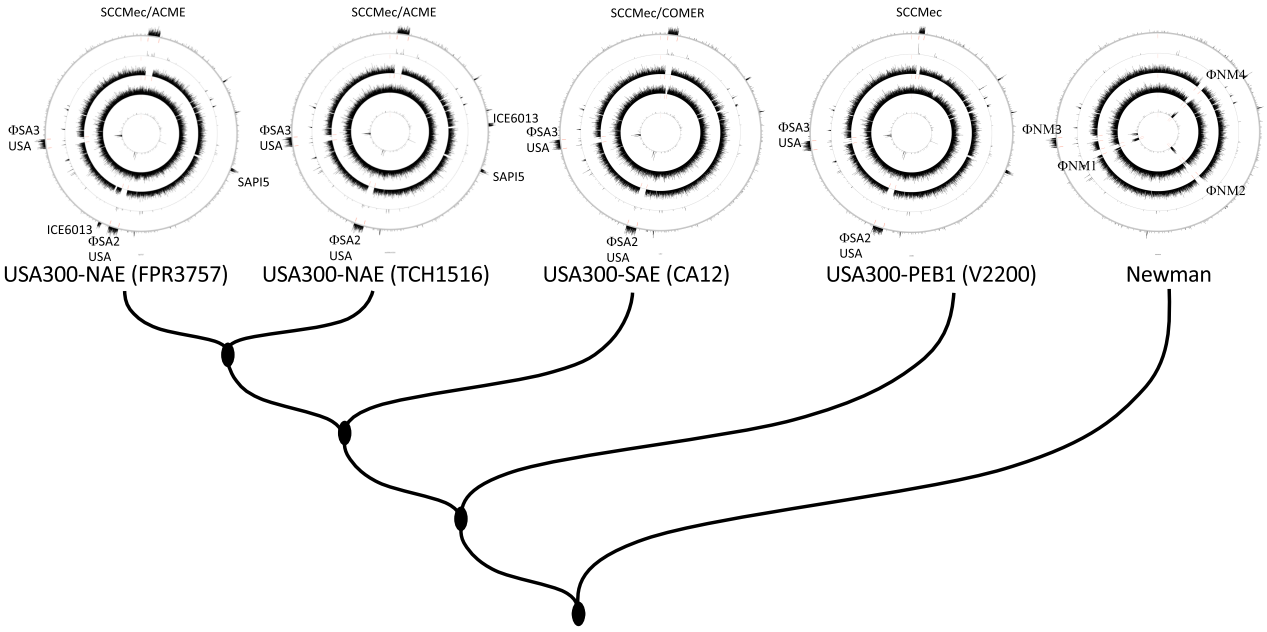

Fig. 4. Nonhomologous recombination. We show the evolution of mobile elements in the Clonal Complex 8. We map *k*-mers clustered into 5 groups back to each of the genomes shown and highlight notable mobile elements including the Staphylococcal Chromosomal Cassette carrying the methicillin resistance gene *mecA* (SSCmecA), the Arginine Catabolic Mobile Element (ACME), the Copper and Mercury Resistance mobile element (COMER), *S. aureus* Pathogenicity Island 5 (SaPI5), phage 2 from USA300 (ϕSA2usa), the integrative conjugative element 6013 (ICE6013), and phage 3 from USA300 (ϕSA2usa) and phages from strain Neman (ϕNM1-4). In most cases, *k*-mers from mobile elements have been sorted into the outer track, while there are absences in other tracks. The tree drawing represents well-established relationships between these strains. Representatives of the NAE, the SAE, and the pre-epidemic early branching clade 1 are shown.

simpler evolutionary scenarios erodes the information captured (supplementary fig. S4, Supplementary Material online).

The Bottleneck in Action: Real-World Recombination

DNA exchange between genomes can be either (i) homologous, involving corresponding genome fragments in the donor and acceptor, or (2) nonhomologous, involving importation of a new sequence into an existing lineage. The latter is commonly referred to as horizontal transfer and often involves mobile elements. We used 2 systems of *Staphylococcus aureus* genomes to show that the IB can detect both phenomena.

For a test of nonhomologous recombination in real genomes, we chose a set of *S. aureus* sequences with well-characterized community-associated methicillin resistance (supplementary table S1, Supplementary Material online) and clearly defined mobile genetic elements (Diep et al. 2006; Highlander et al. 2007; Planet et al. 2015). We assigned *k*-mers to 5 clusters and mapped them back to each of the genomes (Fig. 4). The SIB identified the Staphylococcal Chromosomal Cassette carrying the methicillin resistance gene *mecA* (SSCmecA), the Arginine Catabolic Mobile Element (ACME) that uniquely identifies the North American Epidemic (NAE) clade, and the Copper and Mercury Resistance mobile element (COMER) that uniquely identifies the South American Epidemic (SAE) clade. It also found the *S. aureus* Pathogenicity Island 5 (SaPI5), phage 2 from USA300 (ϕSA2usa), the integrative conjugative element 6013 (ICE6013), and phage 3 from USA300 (ϕSA3usa). Notably, the very widespread ICE6013 (Smyth and Robinson 2009; Sansevere and Robinson 2017) incorporates at variable locations in different USA300 genomes as shown in the examples given (Sabirova et al. 2014; Jamrozy et al. 2016). The lack of SaPI5 in some strains of USA300-SAE has been previously reported (Bianco et al. 2023). Clearly, the evolution of pathogenicity in these genomes including the order of mobile element integrations over time is congruent with the tree-based view of these strains.

To test our ability to detect homologous recombination, we used genomes from ST239 *S. aureus* to corroborate a known, large-scale recombination event found in nature. The ST239 strain is a hybrid: a segment from a CC30 (clonal complex 30) donor replaced nearly 20% of the homologous region in a CC8 strain (Robinson and Enright 2004). The evolutionary histories of genes across these segments are incongruent. Previous studies compared the histories of thousands of genes to reach this conclusion (Narechania et al. 2016). Here, we attempt to localize this same phenomenon using the co-occurrence pattern of *k*-mers alone. We chose 10 genomes (supplementary table S1, Supplementary Material online), sampled from both the donor clade (CC30), the recipient clade (CC8), and genomes outside of the evolutionary event.

Figure 5a highlights 2 of these 10 genomes and 3 of the 10 clusters we modeled for this analysis. Both *S. aureus* COL (CC8) and *S. aureus* T0131 (ST239) share a large, congruent core. The gap in this core characterizes the dimensions of the recombination event, whose *k*-mers are split into 2 other clusters, one from the donor and the other from the acceptor, shown here as the second and third tracks. The bottleneck learns the structural evolution of the clade as tracts of co-occurring sequence. The clusters themselves comprise our evolutionary model for the structural event and highlight the core. The hallmark of homologous recombination is replacement of shared genomic sequence. Earlier, we confirmed that we can capture each side of a simulated homologous event as separate clusters of *k*-mers (supplementary fig. S2, Supplementary Material online). In Fig. 5a, we show that we can do the same for real data. Genomes that exhibit nonhomologous recombination lack this kind of donor/acceptor pairing among clusters. The importation events characteristic of mobile

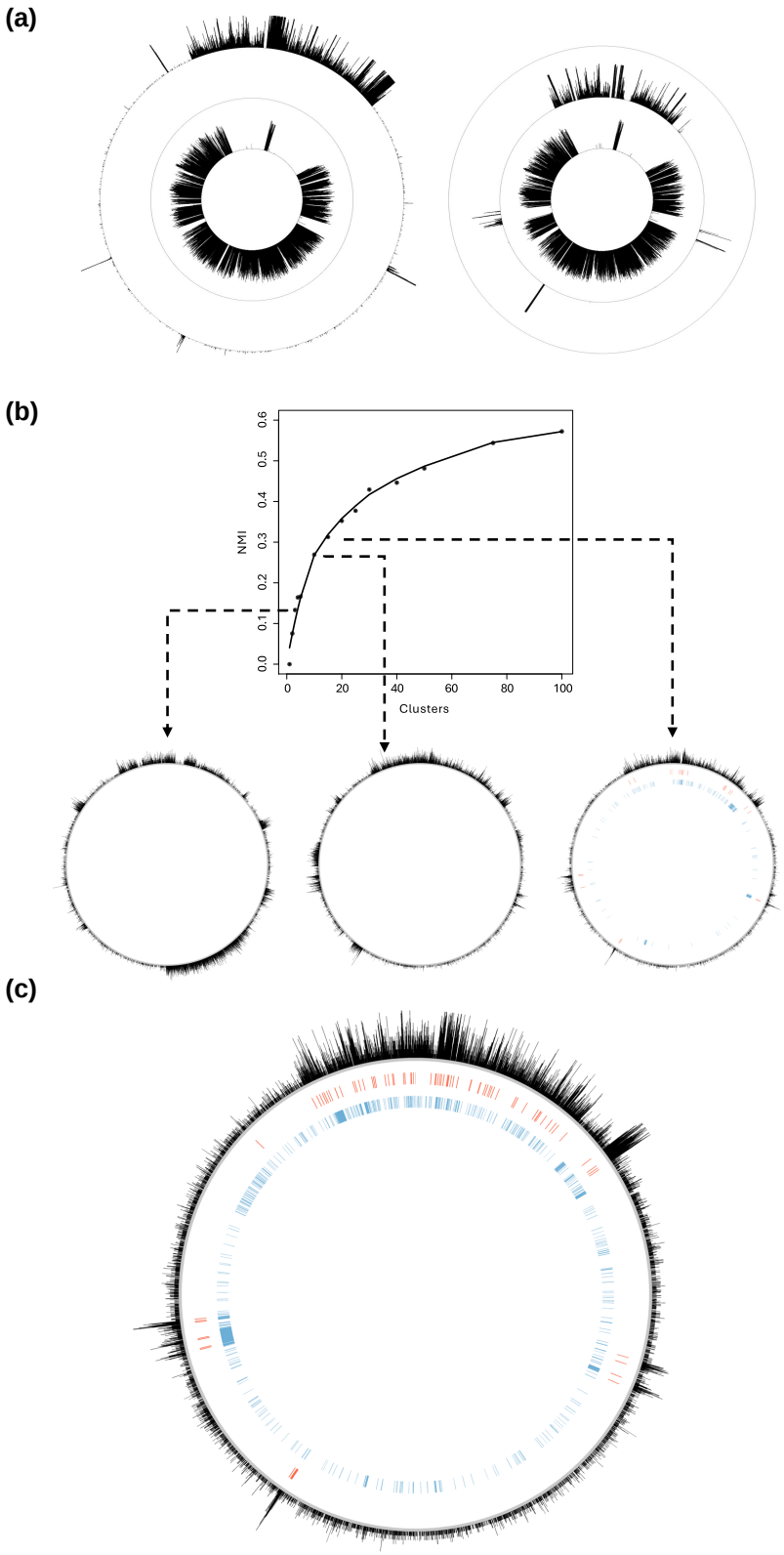

Fig. 5. Homologous recombination. We model ST239's hybridization event with *S. aureus* genomes chosen to include both the donor (COL from the CC30 donor lineage) and the acceptor (T0131 from the CC8 acceptor). In a), we show an analysis of 10 genomes. Of the 15 clusters calculated, we highlight the 3 that capture the hybridization event. The innermost track is a frequency plot of *k*-mers that define the core. The second and third tracks are flipsides of the HGT event that created ST239. In b), we show the relevance–compression curve of a series of runs with 100 genomes, 50 from CC30 and 50 from CC8. Embedded snapshots map the cluster with *k*-mers from the recombination event to the T0131 genome. We see noise polluting *k*-mers from the recombined region when we model 5 and 10 clusters. The recombination event eventually stabilizes at 15 clusters. We include recombination breakpoints as called in the SIB (middle track) and in ClonalFrameML (inner track) for the 15 cluster analysis. In c), we show an analysis of 1000 genomes, 500 from CC30 and 500 from CC8. We plot *k*-mers from the cluster defining the recombination region as the outer track on a circos plot of T0131. Breakpoints from the SIB (red, middle track) and from ClonalFrameML (blue, innermost track) are shown for comparison.

element transfers are one-sided, embedding in just a single cluster. The SIB can therefore distinguish homologous recombination from nonhomologous through clustering behavior alone.

To challenge our process with more realistic conditions, we searched for the ST239 recombination event among 100 genomes (Fig. 5b), half randomly populated from a database of CC30 strains and the other half from CC8. Across a range of clusters modeled, we calculated a relevance–compression curve to ascertain an optimum. The curve shows clear saturating behavior, but unlike our simulations, there is no stark elbow. We embed circos plots of the ST239 cluster against the T0131 genome and show that increasing the number of clusters seems to consolidate the hybridization region by c = 15. Before that, between c = 3 and c = 15, hints of the region do appear, but snapshots show pollution by *k*-mers from other locations (Fig. 5b; supplementary fig. S5, Supplementary Material online). For c = 15, we include recombination boundaries as calculated by the SIB (red hashes) and boundaries calculated by ClonalFrameML. As we saw in simulation, ClonalFrameML overcalls recombination breakpoints. False positives are clear around the entire expanse of the T0131 genome. In this case, ClonalFrameML also seems to miss smaller co-clustering regions. The SIB performs more reasonably, outlining the ST239 event and 3 smaller, co-occurring regions we detect in Fig. 5a.

In 1000 genome data sets, the noise inherent to ClonalFrameML is even more obvious. Figure 5c shows a circos plot of the ST239 cluster from an experiment (c = 15) containing 500 CC8 and 500 CC30 strains. Even with this many genomes, the recombination event is clear, and clearly demarcated by the SIB. Though ClonalFrameML does seem to have the densest number of calls at the most interesting breakpoints, its calls are noisy and filled with false positives. The smaller regions missed by ClonalFrameML in the 100-genome data set are detected here as dense tracts of breakpoints. Their omission in the smaller data set is perhaps a result of sampling artifacts. We repeated the experiment with 400, 300, 200, and 100 donor and acceptor genomes. In each case, we titrate in random *S. aureus* genomes to build out 1000 genome sets, diminishing the fraction of genomes that contain the ST239 recombination event. But in every case—even those experiments with just 100 donor and acceptor genomes—we find the ST239 region (supplementary fig. S6, Supplementary Material online). And we do so in far less time and with far less memory (supplementary table S2, Supplementary Material online).

The key to our computational savings is both conceptual and technical. Conceptually, we take a completely different approach compared to dominant methods in the field. We designate no reference. We do not align, annotate, or build trees. The cost in terms of both CPU and memory consumption of these expensive bioinformatic processes—all prerequisites to ClonalFrameML or Roary—are clear in the 100-genome set and even more obvious at 1000 genomes. Our analysis is based on the sorting of strings alone. But do we need all these strings? Technically, we do not. We implement *k*-mer sketching using a modulo hash function that runs natively in our program. In supplementary fig. S7, Supplementary Material online, we show that sketching *k*-mers from our genomes may blunt the signal we see from the ST239 cluster, but even at a sketch rate of 100, we do not lose it. Clearly, there is signal to spare, and the computational savings from this simple technical tweak are significant.

Our approach suggests a new type of comparative genomics based on compression. Figure 6 shows relevance–compression curves for 10 ST239 genomes alongside 10 genomes of *Mycobacterium tuberculosis* and *Helicobacter pylori*. The convex shape of the *M tuberculosis* curve reflects its clonality. Data from a more open pangenome that resists compression flattens the curve. Highly recombinogenic species like *H. pylori* suffer this sort of steep information loss.

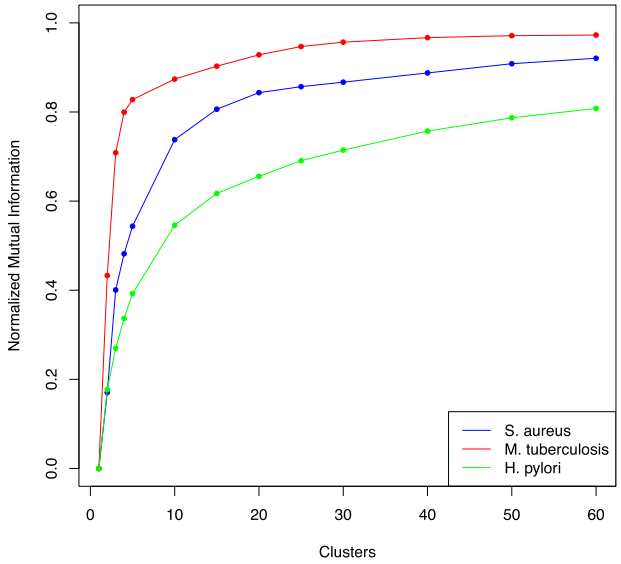

Fig. 6. Relevance–compression curves across 3 species. We show that relevance–compression curves across 3 selected species trace the increase in normalized mutual information with the number of clusters modeled. The rate of this increase can function as a marker for evolutionary strategy. *Mycobacterium tuberculosis* is thought to demonstrate little if any HGT; *S. aureus* is considered largely clonal with occasional HGT; and *H. pylori* is known to employ HGT as an engine for diversity.

In supplementary fig. S8, Supplementary Material online, we show relevance–compression curves of 100 genomes for the 28 most well studied microbial species in RefSeq. In all cases, increased modeling captures more information, but at varying rates. We also demonstrate that we can reach a steady-state NMI with a small number of randomly selected genomes regardless of clade (supplementary fig. S9, Supplementary Material online). We conclude that the shape of the relevance–compression curve is a proxy for evolutionary strategy and show that 50 randomly selected genomes can reflect the nature of the species.

The Bottleneck as a New Measure of Recombination

We have shown that in simulation, given a set number of clusters, increased homologous recombination results in decreased NMI. To test whether this pattern holds for real data, we analyzed the top 13 most sequenced bacterial species in RefSeq with at least 100 complete, contiguous genome assemblies. For each of these 13 species, we randomly selected 1 complete genome as a reference and 100 random genomes as queries. We calculated R/theta for these selections and repeated the experiment 100 times for all 13 species. Because these species are all subject to nonhomologous recombination as well, we calculated the core and accessory genomes for all selections, summarizing the effects of gene flux with parsimony steps of gene presence/absence matrices. Figure 7 shows R/theta (Fig. 7a) and parsimony steps (Fig. 7b) plotted against the corresponding 5 cluster NMI. The data indicate that NMI is an inverse mirror of R/theta and pangenome flux. NMI and r/m display a similar relationship

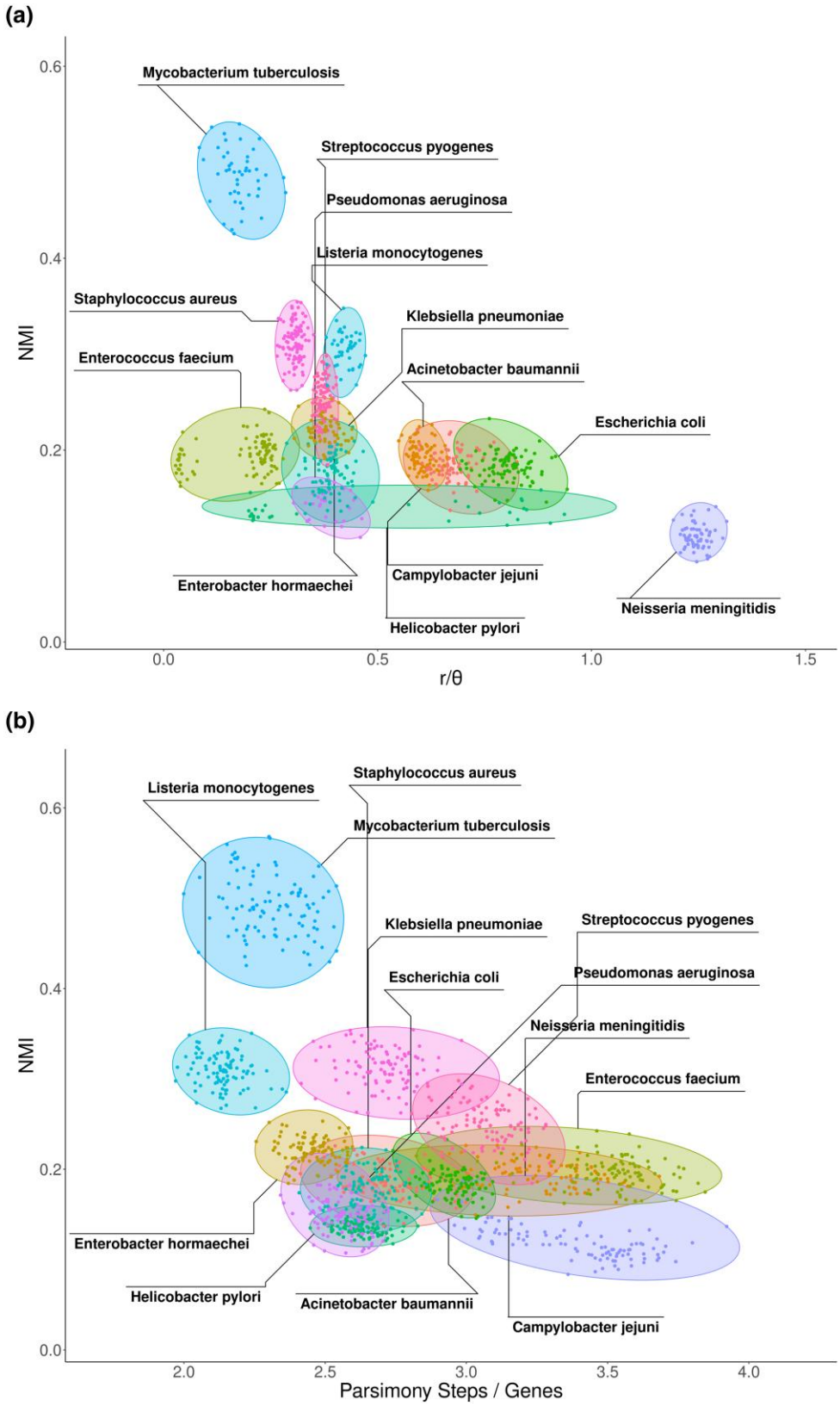

Fig. 7. NMI tracks R/theta and gene flux. We calculate R/theta a) and gene flux b) for 100 replicates of 100 randomly chosen genomes across 13 key species in RefSeq. We calculate gene flux as the parsimony steps in a presence absence matrix of the pangenome normalized over the number of total genes. More recombinogenic species are less easily compressed (lower NMI) regardless of their recombination mechanism.

(supplementary fig. S10, Supplementary Material online). An analysis restricted to accessory genome flux also returns the same relationship (supplementary fig. S11, Supplementary Material online). Higher levels of any kind of recombination tend to lower compressibility (supplementary table S3, Supplementary Material online). NMI clearly offers a

conceptual alternative to R/theta and flux calculable in a fraction of the time (supplementary fig. S12, Supplementary Material online). Like other metrics based on information compression (Narechania et al. 2024), the NMI conflates biological signal, combining the influence of homologous and nonhomologous recombination. We do not see this as a flaw. For example, a few of our species have a broad NMI range over a narrow R/theta span. We would expect this behavior if NMI were partially driven by nonhomologous recombination. Both recombination mechanisms contribute to species information diversity and the NMI provides an index for their joint effect. While NMI confers a joint index, the pattern in our *k*-mer clusters can distinguish between homologous and nonhomologous recombination (Figs. 4 and 5; supplementary fig. S2, Supplementary Material online). We conclude that NMI—a simple distortion metric produced as sequence compresses through an information bottleneck—is a broad indicator of the information diversity in a species regardless of how that diversity is generated.

The wide range in R/theta values for *H. pylori* is emblematic of the primary weakness of most dominant recombination methods: reference bias. In supplementary fig. S13, Supplementary Material online, we show the range of R/theta values calculated for 10 randomly chosen genomes of *S. aureus*, *M. tuberculosis*, and *H. pylori* against all possible contiguous, complete references known for these genomes. While *S aureus* and *M tuberculosis* show relatively tight ranges, *H. pylori* replicates vary over 1 full unit depending on the reference chosen.

Finally, in mining the genomes for Fig. 7, we observed some that were misclassified at the species level. Long-branch effects from a single misclassified sequence can ruin an entire replicate, distorting r/m and R/theta. Omitting these misclassified sequences with a MASH distance-based screen clarified the relationship between NMI and R/theta (supplementary fig. S14, Supplementary Material online). We removed replicates that contained outliers in the distribution of all pairwise MASH distances within a species. We settled on removing genomes that had average MASH distances 4.5 standard deviations from the species mean, discarding the heterogeneous set of points near the origin (supplementary fig. S14, Supplementary Material online, ALL) while retaining the bulk of the replicates for each species. Our data suggests that even vetted resources like RefSeq can be misleading. The uncertainty in RefSeq coupled with the instability inherent to reference-based analyses argues for reference free approaches like the IB.

The capacity of the bottleneck—in the Shannon sense—can be used to modulate detection. In the case of 10 ST239 genomes, asking for just 2 clusters—a very narrow channel—captures more than 40% of the relevant information (Fig. 6). Remarkably, these 2 clusters separate the core from the recombined region. Even the simplest model learns the most prominent evolutionary process. Because the technique is inherently lossy, the SIB will never recover all the information originally encoded but aims to extract the most salient bits. As we have seen with ST239, the largest gains usually come early (Still and Bialek 2004; Slonim 2003).

In the light of evolution, the bend in our relevance–compression curves may have a deeper meaning. For *S. aureus*, 15 clusters are enough to adequately capture the primary set of *k*-mer aggregation patterns across our chosen genomes. This point of diminishing returns may signify an opportunity for interpretive balance: not so many clusters that we dilute and fragment dominant evolutionary events and not so few that we lose subtle *k*-mer co-occurrence patterns to noise. This particular use of the elbow method in our information theoretic context puts a crude limit on the dominant evolutionary paths taken by the elements that comprise our pangenome. And the NMI emitted at this asymptote offers a new way to describe the lability of the species.

Methods

IB with Neck

Neck is the program we use to execute the IB in our evolutionary context. Neck accepts 2 key parameters from the user: *k*-mer size and the number of clusters to model. The program performs a *k*-mer sketch of its query genomes by sliding a length *k* window across the sequence. No reference is required. We use a 64-bit hash function on canonical *k*-mers to generate a bottom-k sketch at the desired depth. The learning process inherent to the bottleneck procedure iterates over these *k*-mers, minimizing cluster distortion on each successive loop through the sketched strings. This learning process is called the SIB. Neck will support both assembled and unassembled sequence, and in both cases, outputs clusters of unlocalized strings. To localize strings on assembled sequence, we provide an auxiliary program called neck_paint (see below). A typical run of neck looks like this:

```
neck -i genomes/-k 19 -m 1 -c 50 -n 4 -s 10 -o out_neck_c50/
```

The query genomes are found in the genomes directory (*-i genomes/*), 1 genome per file. In this run, 19-mers are hashed at a rate of 10% (*-s 10*). In a bottom-k context, this means that the lowest tenth of the hash values and their corresponding *k*-mers are sketched into the set of strings used in the bottleneck procedure. For this analysis, we model 50 clusters (*-c 50*) and compress the clusters over 4 iterations of the optimization (*-n 4*). Output files, including the clusters of *k*-mers found and the NMI value are redirected to out_neck_c50 (*-o out_neck_c50/*).

Visualization with Neck_paint

Though not strictly required, assembled genomes give a user the opportunity to contextualize clusters of *k*-mers generated in neck. The script neck_paint aligns *k*-mers from the clusters to query genomes using bowtie2 and calculates the depth of the resulting alignments with samtools. Using these depth statistics, neck_paint will calculate change points between regions of high and low *k*-mer frequency. On the core cluster, these regions typically correspond to the core genome and the recombination events. Recombination sites almost always exhibit depleted alignment in the core. De novo recombination intervals independent of any other annotations are inferred from these change points. If recombination regions are known, neck_paint can calculate Wilcoxon statistics on the alignment frequency differences between the known regions and the rest of the genome. We use Circos to visualize every cluster/genome pair, their associated change points, and any known recombination events. A typical run of neck_paint looks like this:

```
neck_paint -i genomes/-c clusters/-l 200 -r changepoint.R -p
128 -m 3 -v 1000 -x log.external -o out_neck_c50_paint/
```

In this case, neck_paint will perform alignments, perform change point analysis, and generate a circos visualization for every genome (*-i genomes/*) cluster (*-c clusters/*) pair. The script changepoint.R (*-r changepoint.R*) delimits the change points using the PELT algorithm given the penalty parameter

(*-v 1000*). The higher the penalty, the stronger the signal needed to mark a change point. A low penalty parameter therefore often suffers from false positives while a high penalty parameter might lack sensitivity. For the circos visualization, alignment frequency is binned into 200 bp intervals (*-l 200*) and shown as a plot with a maximum of 3 units (*-m 3*). If available, recombination events can also be shown as a track (*-x log.external*) and the user can generate Wilcoxon statistics for nonparametric difference between frequencies within these intervals and frequencies outside of them. Because there can be many cluster/genome pairs, the process can be parallelized (*-p 128*) and the output collected into a single directory (*-o out_neck_c50_paint*).

Simulations

We used SimBac to simulate homologous recombination across 100 1 Mb genomes with 3 recombination rates: 0.000005, 0.00001, and 0.000015. We ran SimBac as follows (for 0.00001):

```
SimBac -N 100 -B 1000000 -R 0 -T 0.01 -r 0.00001 -m 0.1
-M 0.1 -e 5000 -o genomes.fasta -c clonal_frame.nwk -b
log.internal -f log.external -d dot.file
```

After splitting the genomes into individual files, we ran neck on each rate as follows:

```
neck -i genomes/-k 19 -c 50 -n 3 -s 10 -o out_c50
```

We calculated the relevance–compression curves using 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, 75, and 100 clusters (-c). After choosing an arbitrary genome for alignment, we aligned all *k*-mers from all clusters from the c = 50 run of neck and visualized these alignments using neck_paint:

```
neck_paint -i ref/-c out_c50/core/-l 200 -r changepoint.R -p
128 -m 3 -v 200 -x log.external -o out_c50_paint
```

The option -v controls the penalty parameter supplied to the PELT change point algorithm. Lower numbers may overcall recombination breakpoints, while higher numbers could conflate them. We compared our calls to those calculated in ClonalFrameML. We ran alignments of *k*-mer clusters in bwa (Li and Durbin 2009) through the wrapper program snippy (Seemann 2015), calculated trees in RAxML (Stamatakis 2014), and used clonalframeml with default as follows:

1. *snippy –outdir <output_dir> –ref <reference.fa> –ctgs <input.kmers>*
2. *snippy-core --ref <reference.fa> --prefix <prefix_name> <output_dir/*>*
3. *raxml -T 2 -m ASC_GTRGAMMA -f d -N 1 -s <core.full.aln> -w out -n best -p 1977 --asc-corr lewis*
4. *ClonalFrameML <raxml.tre> <core.full.aln> <cf.core.full>*

We generated confusion matrices and calculated accuracy, precision, and recall using the intervals called for both neck and clonalframeml and include those as circos tracks in the visualizations.

For the simulations shown in Fig. 3, we generated 100 replicates of 100 genomes subject to every combination of mutation rates from 1e-5 to 0.1 and recombination rates ranging from 1e-5 to 0.1. This required 2500 neck runs (-c 5, -n 3 -s 10).

Neck on *S. aureus*

For our nonhomologous recombination experiment, we selected 11 genomes (supplementary table S1, Supplementary Material online) that illustrate the outbreak of the methicillin resistant *S. aureus* strains worldwide. We ran neck with 5 clusters at a 10% sketch rate and used neck_paint to visualize all 5 resulting clusters for the genomes shown.

To test our procedure on homologous recombination, we designed experiments with 10 (supplementary table S1, Supplementary Material online), 100, and 1000 CC8 and CC30 genomes across *S. aureus*. For the larger genome sets, we randomly selected genomes from supplementary table S4, Supplementary Material online. For the 100 genome set, we ran neck as described above, sweeping through 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, 75, and 100 clusters while keeping sketching and iterations constant (-s 10, -n 3), and for the 1000 genome set, we used only -c 10 and -s 100. We ran snippy, RaxML, and clonalframeml as described above against the T0131 reference.

R/Theta Calculations

For each of the organisms shown in Fig. 7, we randomly selected 100 genomes, one of which was required to be a contiguous reference. For each organism, we repeated this experiment 100 times. For all replicates, we (i) calculated R/theta using the clonalframeml procedure outlined above and (ii) calculated NMI with neck, c = 5, s = 10, and $n = 3$.

Gene Flux Calculations

For each replicate across all organisms generated for the R/theta calculation above, we also calculated gene flux. We began by annotating each genome with prokka (Seemann 2014 and calculated pangenomes using the resulting gff files in roary (Page 2015). Roary generates a presence absence matrix which we then transposed into a nexus file for input into PAUP for parsimony tree reconstruction. We interpret tree steps as a measure of gene flux. Specific commands for prokka and roary were as follows:

1. *prokka --cpus <cpus> --outdir <outoput_directory> --prefix <prefix_name> <input_data>*
2. *roary -p <cpus> -e --mafft -f <input_dir/*.gff>*

Software

NECK (https://github.com/narechan/neck).

Supplementary Material

Supplementary material is available at *Molecular Biology and Evolution* online.

Data Availability

The data used in this manuscript are publicly available on NCBI (https://www.ncbi.nlm.nih.gov).

References

Bianco CM, Moustafa AM, O'Brien K, Martin MA, Read TD, Kreiswirth BN, Planet PJ. Pre-epidemic evolution of the MRSA USA300 clade and a molecular key for classification. *Front Cell Infect Microbiol*. 2023:13:1081070. https://doi.org/10.3389/fcimb.2023.1081070.

Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. *J Mach Learn Res*. 2003:3:993–1022.

Brown T, Didelot X, Wilson DJ, Maio ND. SimBac: simulation of whole bacterial genomes with homologous recombination. *Microb Genomics*. 2016:2(1):e000044. https://doi.org/10.1099/mgen.0.000044

Chen X, Hu X, Shen X, Rosen G. 2010. Probabilistic topic modeling for genomic data interpretation. 2010 IEEE international conference on bioinformatics and biomedicine (BIBM); Hong Kong, China: IEEE. p. 149–152.

Chiu JC, Lee EK, Egan MG, Sarkar IN, Coruzzi GM, DeSalle R. OrthologID: automation of genome-scale ortholog identification within a parsimony framework. *Bioinformatics*. 2006:22(6):699–707. https://doi.org/10.1093/bioinformatics/btk040.

Cong Y, Chan Y-B, Ragan MA. A novel alignment-free method for detection of lateral genetic transfer based on TF-IDF. *Sci Rep*. 2016:6(1):30308. https://doi.org/10.1038/srep30308.

Croucher NJ, Page AJ, Connor TR, Delaney AJ, Keane JA, Bentley SD, Parkhill J, Harris SR. Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. *Nucleic Acids Res*. 2015:43(3):e15. https://doi.org/10.1093/nar/gku1196.

Darling ACE, Mau B, Blattner FR, Perna NT. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*. 2004:14(7):1394–1403. https://doi.org/10.1101/gr.2289704.

Didelot X, Wilson DJ. ClonalFrameML: efficient inference of recombination in whole bacterial genomes. *PLOS Comput Biol*. 2015:11(2):e1004041. https://doi.org/10.1371/journal.pcbi.1004041.

Diep BA, Gill SR, Chang RF, Phan TH, Chen JH, Davidson MG, Lin F, Lin J, Carleton HA, Mongodin EF, *et al.* Complete genome sequence of USA300, an epidemic clone of community-acquired meticillin-resistant Staphylococcus aureus. *Lancet*. 2006:367(9512):731–739. https://doi.org/10.1016/S0140-6736(06)68231-7.

Erlich Y. A vision for ubiquitous sequencing. *Genome Res*. 2015:25(10):1411–1416. https://doi.org/10.1101/gr.191692.115.

Fraser C, Hanage WP, Spratt BG. Recombination and the nature of bacterial speciation. *Science*. 2007:315(5811):476–480. https://doi.org/10.1126/science.1127573.

Guttman DS, Dykhuizen DE. Clonal divergence in Escherichia coli as a result of recombination, not mutation. *Science*. 1994:266(5189):1380–1383. https://doi.org/10.1126/science.7973728.

Highlander SK, Hultén KG, Qin X, Jiang H, Yerrapragada S, Mason EO Jr, Shang Y, Williams TM, Fortunov RM, Liu Y, *et al.* Subtle genetic changes enhance virulence of methicillin resistant and sensitive Staphylococcus aureus. *BMC Microbiol*. 2007:7(1):99. https://doi.org/10.1186/1471-2180-7-99.

Jamrozy DM, Harris SR, Mohamed N, Peacock SJ, Tan CY, Parkhill J, Anderson AS, Holden MTG. Pan-genomic perspective on the evolution of the Staphylococcus aureus USA300 epidemic. *Microb Genom*. 2016:2(5):e000058. https://doi.org/10.1099/mgen.0.000058.

Killick R, Eckley IA. Changepoint: an *R* package for changepoint analysis. *J Stat Softw*. 2014:58(3):1–9. https://doi.org/10.18637/jss.v058.i03.

Killick R, Fearnhead P, Eckley IA. Optimal detection of changepoints with a linear computational cost. *J Am Stat Assoc*. 2012:107(500):1590–1598. https://doi.org/10.1080/01621459.2012.737745.

Kullback S, Leibler RA. On information and sufficiency. *Ann Math Stat*. 1951:22(1):79–86. https://doi.org/10.1214/aoms/1177729694.

Kyrpides NC, Woyke T, Eisen JA, Garrity G, Lilburn TG, Beck BJ, Whitman WB, Hugenholtz P, Klenk H-P. Genomic encyclopedia of type strains, phase I: the one thousand microbial genomes (KMG-I) project. *Stand Genomic Sci*. 2014:9(3):1278–1284. https://doi.org/10.4056/sigs.5068949.

La Rosa M, Fiannaca A, Rizzo R, Urso A. Probabilistic topic modeling for the analysis and classification of genomic sequences. *BMC Bioinformatics*. 2015:16(S6):S2. https://doi.org/10.1186/1471-2105-16-S6-S2.

Li H, Durbin R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*. 2009:25(14):1754–1760. https://doi.org/10.1093/bioinformatics/btp324.

Lin J. Divergence measures based on the Shannon entropy. *IEEE Trans Inf Theory*. 1991:37(1):145–151. https://doi.org/10.1109/18.61115.

Liu L, Tang L, Dong W, Yao S, Zhou W. An overview of topic modeling and its current applications in bioinformatics. *Springerplus*. 2016:5(1):1608. https://doi.org/10.1186/s40064-016-3252-8.

Marzen SE, DeDeo S. The evolution of lossy compression. *J R Soc Interface*. 2017:14(130):20170166. https://doi.org/10.1098/rsif.2017.0166.

Narechania A, Baker R, DeSalle R, Mathema B, Kolokotronis S-O, Kreiswirth B, Planet PJ. Clusterflock: a flocking algorithm for isolating congruent phylogenomic datasets. *Gigascience*. 2016:5(1):44. https://doi.org/10.1186/s13742-016-0152-3.

Narechania A, Bobo D, Deitz K, DeSalle R, Planet PJ, Mathema B. Rapid SARS-CoV-2 surveillance using clinical, pooled, or wastewater sequence as a sensor for population change. *Genome Res*. 2024:34(10):1651–1660. https://doi.org/10.1101/gr.278594.123.

Nayfach S, Roux S, Seshadri R, Udwary D, Varghese N, Schulz F, Wu D, Paez-Espino D, Chen I-M, Huntemann M, *et al.* A genomic catalog of Earth's microbiomes. *Nat Biotechnol*. 2021:39(4):499–509. https://doi.org/10.1038/s41587-020-0718-6.

Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol*. 2016:17(1):132. https://doi.org/10.1186/s13059-016-0997-x.

Page AJ, Cummins CA, Hunt M, Wong VK, Reuter S, Holden MTG, Fookes M, Falush D, Keane JA, Parkhill J. Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*. 2015:31(22):3691–3693. https://doi.org/10.1093/bioinformatics/btv421.

Pereira F, Tishby N, Lee L. Distributional clustering of English words. *31st Annual meeting of the association for computational linguistics*. Columbus (Ohio), USA: Association for Computational Linguistics; 1993. p. 183–190.

Planet PJ, Diaz L, Kolokotronis S-O, Narechania A, Reyes J, Xing G, Rincon S, Smith H, Panesso D, Ryan C, *et al.* Parallel epidemics of community-associated methicillin-resistant staphylococcus aureus USA300 infection in North and South America. *J Infect Dis*. 2015:212(12):1874–1882. https://doi.org/10.1093/infdis/jiv320.

Pritchard JK, Stephens M, Donnelly P. Inference of population structure using multilocus genotype data. *Genetics*. 2000:155(2):945–959. https://doi.org/10.1093/genetics/155.2.945.

Robinson DA, Enright MC. Evolution of staphylococcus aureus by large chromosomal replacements. *J Bacteriol*. 2004:186(4):1060–1064. https://doi.org/10.1128/JB.186.4.1060-1064.2004.

Sabirova JS, Xavier BB, Ieven M, Goossens H, Malhotra-Kumar S. Whole genome mapping as a fast-track tool to assess genomic stability of sequenced Staphylococcus aureus strains. *BMC Res Notes*. 2014:7(1):704. https://doi.org/10.1186/1756-0500-7-704.

Sansevere EA, Robinson DA. Staphylococci on ICE: overlooked agents of horizontal gene transfer. *Mob Genet Elem*. 2017:7(4):1–10. https://doi.org/10.1080/2159256X.2017.1368433.

Seemann T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*. 2014:30(14):2068–2069. https://doi.org/10.1093/bioinformatics/btu153.

Seemann T. snippy: fast bacterial variant calling from NGS reads. 2015 [accessed 2024 Nov 7]. https://github.com/tseemann/snippy.

Shannon CE. A mathematical theory of communication. *Bell Syst Tech J*. 1948:27(3):379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Sims GE, Jun S-R, Wu GA, Kim S-H. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proc Natl Acad Sci U S A*. 2009:106(8):2677–2682. https://doi.org/10.1073/pnas.0813249106.

Slonim N. The information bottleneck: theory and applications. *Dr Diss Hebr Univ Jerus Isr*. 2003:157. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f9357064ef06a30f4533901cbc956bb25af646ad

Slonim N, Friedman N, Tishby N. 2002. Unsupervised document classification using sequential information maximization. Proceedings

of the 25th annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '02. New York (NY), USA: ACM. p. 129–136.

Smyth DS, Robinson DA. Integrative and sequence characteristics of a novel genetic element, ICE6013, in Staphylococcus aureus. *J Bacteriol*. 2009:191(19):5964–5975. https://doi.org/10.1128/JB.00352-09.

Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*. 2014:30(9):1312–1313. https://doi.org/10.1093/bioinformatics/btu033.

Still S, Bialek W. How many clusters? An information-theoretic perspective. *Neural Comput*. 2004:16(12):2483–2506. https://doi.org/10.1162/0899766042321751.

Tishby N, Pereira FC, Bialek W. 2000. The information bottleneck method, *arXiv*, *arXiv:physics/0004057*, preprint: not peer reviewed.

Wallach HM, Mimno D, McCallum A. 2009. Rethinking LDA: why priors matter. Proceedings of the 22nd international conference on neural information processing systems. NIPS'09. Red Hook (NY), USA: Curran Associates Inc. p. 1973–1981.

Wang J. The computer program structure for assigning individuals to populations: easy to use but easier to misuse. *Mol Ecol Resour*. 2017:17(5):981–990. https://doi.org/10.1111/1755-0998.12650.

Zhao Y, Wu J, Yang J, Sun S, Xiao J, Yu J. PGAP: pan-genomes analysis pipeline. *Bioinformatics*. 2012:28(3):416–418. https://doi.org/10.1093/bioinformatics/btr655.