Supplementary appendix

A dual barcoding approach to bacterial strain nomenclature: Genomic taxonomy of *Klebsiella pneumoniae* strains

Melanie Hennart [a,b], Julien Guglielmini [c], Sébastien Bridel [a], Martin M.C. Maiden [e], Keith A. Jolley [e], Alexis Criscuolo [c] and Sylvain Brisse [a]

[a] Institut Pasteur, Université Paris Cité, Biodiversity and Epidemiology of Bacterial Pathogens, Paris, France

[b] Sorbonne Université, Collège doctoral, F-75005 Paris, France

[c] Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics Hub, F-75015 Paris, France

[d]Department of Zoology, University of Oxford, 11a Mansfield Road, Oxford, OX1 3SZ, United Kingdom

Contents:

Detection of inter-phylogroup hybrids

To define hybrid genomes (*i.e.,* arising from multiple ancestral populations), the 45 reference genomes representative of the seven phylogroups Kp1-Kp7 (**Table S9**) were used as models. First, for each of the 7,388 other genome sequences, the closest reference genome was determined by estimating the average nucleotide identity (ANI) using FastANI v1.1 [1]. Every genome *x* with ANI percentage > 99% against its closest reference genome *y* was then classified without ambiguity into the same phylogroup as the one of *y*. Second, for each genome classified into a phylogroup (Kp1-Kp7), all its cgMLST alleles were labelled with this phylogroup. Third, for each of the 629 scgMLSTv2 loci, every distinct allele associated to more than one phylogroup labels was unlabeled, given that such an allele cannot be considered as a reliable representative of a unique phylogroup (*e.g.*, it was too conserved, or involved in horizontal transfer between phylogroups). Fourth, for each locus, every unlabeled sequence identical to one of the remaining labelled alleles (i.e., sequence belonging to a genome that was not assigned to a phylogroup during step 1) was labelled accordingly. Such a procedure enabled the characterization of a large set of alleles that are each representative of one of the seven phylogroups Kp1-Kp7.

As a result, almost all cgMLST profiles were mostly made up by alleles belonging to only one phylogroup label (see **Figure S15**). However, notable exceptions were observed, with some cgMLST profiles being composed of alleles belonging to two phylogroup labels (see e.g., **Figure S15**). To define putative hybrid profiles, a phylogroup homogeneity index was determined for each profile, defined as the proportion of loci labelled with the predominant phylogroup (normalized by the number of non-missing alleles called in the profile). As expected, for each phylogroup Kp1-Kp7, most profiles are associated with high phylogroup homogeneity indices (see distributions in **Figure S12**). However, a total of 138 cgMLST profiles (1.9%; mainly within Kp1, Kp2 and Kp4) were associated with atypically smaller homogeneity indices, and mapping of allele phylogroup labels along the chromosome showed that many of these 138 cgMLST profiles appeared to result from large-scale inter-phylogroup recombination, while a few were made up by many unlabeled alleles (**Figure S1**).

Large recombination events were detected in 1.9% (138/7198) genomes and mainly involved phylogroups Kp1, Kp2 and Kp4; we found 42 Kp4 genomes resulting from large-scale recombination of Kp1 out of the 50 hybrid Kp4 genomes (**Figure S1**), while others form a multitude of small-scale recombination events. Next, 17 Kp1 genomes were observed with a large-scale recombination (12 with a Kp2 insertion, 4 with a Kp4 insertion and 1 with a Kp3 insertion), as well as 3 Kp2 genomes resulting from a large Kp1 insertion. In addition, 42 profiles of phylogroup Kp3 resulted from horizontal gene transfer (but not large-scale recombination events) from non-KpSC donors (**Table 1;**

Figure S2; Table S8). The recombination breakpoints were non-randomly distributed along the genomes: most (109/126, 86.5%) were localized in the second half of the genome (3 Mb – 5.2 Mb of NTUH-K2044 genome coordinates), whereas in the first part (0 – 3 Mb) accounted for only 15 breakpoints.

These 138 genomes presenting hybrid profiles (or with multiple alleles of undefined origins) were therefore discarded during our initial population structure analyses and classification steps, which were based on the remaining 7,060 genomes that likely arose from vertical evolution.

Outbreak datasets to estimate variation of shallow-level classification groups

We searched for previously published genomic epidemiology studies. These genomic investigations of outbreaks (or clusters of cases) together comprised 9 sets of isolates defined as related based on epidemiologically and genomic evidence (**Table S7**). Distribution of the cgMLST pairwise distances among isolates within each outbreak cluster was investigated (**Tables S6, S7**).

Minimum Spanning tree-based clustering of cgMLST profiles: building and assessment

A pairwise dissimilarity between two cgMLST profiles can be defined by the proportion of loci with two distinct alleles among the loci where alleles are defined in both profiles. A pairwise dissimilarity matrix can be computed from n cgMLST profiles, and can be used to build a minimum spanning tree (MStree; *e.g.*, Kruskal, 1956; Prim, 1957a; Dijkstra, 1959), allowing to infer a clustering of the cgMLST profiles, defined by the k different connected components obtained by removing from the MStree all edges of length larger than a specified threshold t. Such an MStree-based clustering is closely related to the single-linkage classification of the n cgMLST profiles (*e.g.*, Gower and Ross, 1969; Johnson, 1967).

In order to determine optimal thresholds t, several criteria can be used. Among these criteria, the average silhouette coefficient S_t assesses the ability of an MStree-based clustering to consistently represents in k class(es) the 'natural' grouping of the cgMLST profiles [7,8]. When S_t is close to 1, the clustering can be considered as accurate. A confidence interval for S_t can be also obtained by considering the distribution of the average silhouette coefficients of different clustering computed from the distance matrix with 'noised' entries.

Further, in order to assess whether an MStree-based clustering C_t (using threshold t) is robust to any subsampling biases, a simple approach is to build another MStree-based clustering C_t' from a subsample of cgMLST profiles, and to measure the agreement between the cgMLST profile partitions

induced by C_t' and C_t. When the level of agreement remains high for different subsampling rates, the corresponding threshold t can be considered as being leading to stable clustering. Among different agreement metrics between partitions, the second adjusted Wallace coefficient w [9,10] estimates the probability of observing a pair of profiles in the same class in C_t when they are clustered in the same class in C_t'. In order to derive a single coefficient W_t from a range of different subsampling rates r (= 10% to 90%), different coefficients w were estimated and averaged for each rate r; the area under the resulting curve (*i.e.*, rates r on X-axis; coefficient w on Y-axis) was computed and normalized (using its maximum expected value). Such a normalized area W_t is close to 1 when the different adjusted Wallace coefficients w (*i.e.*, derived from varying subsampling rates) are all close to their maximum value, therefore showing that the corresponding MStree-based clustering (based on the threshold t) is robust to any subsampling biases. A confidence interval for W_t can be also obtained using the same approach as for S_t (see above).

The MStree-based clustering of cgMLST profiles, as well as the two consistency and stability indices S_t and W_t, respectively, were implemented in the MSTclust tool (https://gitlab.pasteur.fr/GIPhy/MSTclust). For more details, see https://gitlab.pasteur.fr/GIPhy/MSTclust/-/blob/0.21b/Technical.Notes.pdf).

Nomenclature inheritance algorithm

In order to attribute to each clonal group (CG), an identifier that would maximally reflect the widely adopted 7-gene ST identifier of the corresponding isolates, we developed a set of naming rules that prioritize the most abundant ST observed among isolates of each CG, as well as some supplementary rules in case of ties. This algorithm is summarized below, and its implementation as a Python script is provided at https://gitlab.pasteur.fr/BEBP/inheritance-algorithm. **Figure S15** illustrates the process for an example.

Here the process for the CG level is described, but the algorithm was also applied to the SL level. Briefly, the data (*e.g.,* a list of CG-ST pairs) can be formalized as a bipartite graph, in which each CG and ST are nodes, and each non-empty CG-ST intersection is an edge. The weight of each edge is equal to the number of isolates sharing the corresponding CG and ST identifiers. Based on this representation, the algorithm will consist of following all edges in the input graph, in the order of decreasing weight. The approach prioritizes the most frequent ST/CG pairs of isolates, *i.e.*, those that are predominant in the dataset and thus naturally transfers to the CG nomenclature, the identifiers of the highest frequency STs. Rules were implemented to treat the cases of equality of representation of two or more STs connected to the same CG. Once all edges were removed from the graph, it may be that some CGs were not named, for example, because the identifier of their unique corresponding ST was already attributed to another CG. For these orphan CGs, iteratively, the attributed identifier corresponds to the maximal CG identifier already attributed, plus one (**Figure S15**).

Definitions and notations

Let $G = (U, V, E)$ be a weighted bipartite graph where:

- U is a set of clonal groups (CG) inferred from a cgMLST scheme
- V is the set of sequence types (ST) induced by a MLST scheme
- E is the set of edges $\{u, v\}$ with $u \in U$ and $v \in V$
- $w(\{u, v\})$ is the weight of the edge $\{u, v\}$, *i.e.*, the number of isolates inside $u \cap v$

Let $L(v)$ be the label associated to node v (*i.e.*, the ST identifier), and $L(u)$ the one to determine for u (*i.e.*, the CG identifier).

Let $d_G(v)$ be the degree of a node v inside the graph G, *i.e.*, the number of edges incident to node v.

Let $s(u) := \Sigma_{v \in V}\, w(\{u, v\})$ be the size of u, *i.e.*, the number of strains belonging to the CG u.

Let $\Gamma(G)$ be the edge-induced subgraph of a graph G defined by the edge(s) of maximal weight in G.

Let $\Delta_G(u)$ be the set of nodes v that are joined to u inside the graph G and of minimum degree, *i.e.*,

$$\Delta_G(u) := \{v' \in V : \{u, v'\} \in E,\ d_G(v') = \min_{\{u, v\} \in E} d_G(v)\}.$$

Algorithm

```
(a)  ∘ λ := max_{v∈V} L(v)
(b)  ∘ while E ≠ ∅
       do
(c)      ∘ for each connected component G = (U', V', E') of Γ(G)
           do
(d)          ∘ if U' = {μ
               then
(e)              ∘ v := argmin_{v'∈V'} L(v')
               else
(f)              ∘ U'' := argmin_{u'∈U'} s(u')
(g)              ∘ if U'' ≠ {μ}
                 then
(h)                  ∘ μ := argmin_{u''∈U''} L(u'')
(i)              ∘ v := argmin_{v'∈Δ_{G'}(μ)} L(v')
(j)          ∘ L(μ) := L(v)
(k)          ∘ removing μ and v from G, as well as nodes v such that w({μ, v}) = w({μ, v})
(l)  ∘ for each μ ∈ U
       do
(l)      ∘ λ := λ + 1
(m)      ∘ L(μ) := λ
```

Impact of strains input order on LIN codes, and use of Prim's algorithm

By design, the input order of genomes into the cgLINcode nomenclature system influences their attributed code, as is the case for the original LIN code system [12]. We evaluated this impact by quantifying the variation in the number of partitions at a given threshold, as defined by the number of distinct prefixes: for each threshold varying from 1% to 99%, a LIN encoding was defined using this threshold, and the 7,060 high-quality, non-hybrid cgMLST profiles were encoded 500 times with random input orders. This experiment made it possible to determine (i) the threshold values associated with a stronger variability in the final number of values; and (ii) the magnitude of this variability. In the example illustrated in **Figure S10**, we observed that the number of distinct prefixes was affected by the order of encoding, especially in the 450 - 530 mismatches range. Note that this experiment can help to select position thresholds, for example, by favoring those that minimize the variance of the number of partitions (*i.e.*, are less affected by input order).

We next sought to minimize this problem by defining an optimal input order. The one that answered our expectations is the input order guided by a Prim's algorithm [3]. More precisely, the number of categories in a given LIN encoding bin is minimal (*i.e.*, identical to the number of groups created by a single-linkage clustering using the threshold associated to the bin) when the profiles are encoded following the order induced by the traversal of an MStree. Indeed, when following such an order, when a new profile is considered for encoding, then its closest profile is already encoded (by definition of a tree traversal). The optimal order we suggest is therefore verified by noting that the Prim's (1957) algorithm to infer a MStree induces such an MStree traversal. A comparison between the MLSL approach and the cgLIN codes was performed (cgLIN codes in optimal *versus* arbitrary order). We found that the partitioning created by the MLSL approach and that created by the cgLIN codes according to the optimal order, were identical (**Table S1**).

We generated 500 random input orders and then generated cgLIN codes with two bins (the first varies from 1 to 100% with a step of one allelic difference i.e. 100/629, and the second fixed at 100%). Then we counted the number of prefixes, up to the first bin, that were created. We observed that the 10 identifier bins differ in their sensitivity to input order (**Figure S10**). The most affected bins correspond to regions of the pairwise distance distribution with high density; in particular around 485 mismatches, before the mode that corresponds to inter-sublineage distances.

The algorithm below was used to define the input order, without even having to construct an MStree. Indeed, thanks to a simple traversal of the matrix of dissimilarities between profiles, the algorithm makes it possible to quickly determine the optimal order for LIN encoding.

Algorithm

(a) ◦ Create a set "mstSet" that keeps track of vertices already included in MST
(b) ◦ Assign a key value to all vertices in the input graph. Initialize all key values as ∞. Assign key value as 0 for the first vertex so that it is picked first.
(c) ◦ **while** "mstSet" doesn't include all vertices
do
(d) ◦ Pick a vertex *u* which is not there in "mstSet" and has minimum key value.
(e) ◦ Include *u* to "mstSet".
(f) ◦ Update key value of all adjacent vertices of *u*. To update the key values, iterate through all adjacent vertices. For every adjacent vertex *v*, if weight of edge *u* – *v* is less than the previous key value of *v*, update the key value as weight of *u* – *v*.

Note that using key values enables to pick the minimum weight edge from cut. The key values are used only for vertices which are not yet included in MStree; the key value for these vertices indicate the minimum weight edges connecting them to the set of vertices included in MStree. The time complexity required by Prim's (1957) algorithm is O(*E* log *V*) where *E* is the number of edges and *V* is the number of vertices.

List of supplementary figures

List of supplementary tables

Supplementary references

1. Jain, C., Rodriguez-R, L. M., Phillippy, A. M., Konstantinidis, K. T. & Aluru, S. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. *Nat Commun* **9**, 5114 (2018).
2. Kruskal, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.* **7**, 48–48 (1956).
3. Prim, R. C. Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal* **36**, 1389–1401 (1957).
4. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959).
5. Gower, J. C. & Ross, G. J. S. Minimum Spanning Trees and Single Linkage Cluster Analysis. *Applied Statistics* **18**, 54 (1969).
6. Johnson, S. C. Hierarchical clustering schemes. *Psychometrika* **32**, 241–254 (1967).
7. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**, 53–65 (1987).
8. Lengyel, A. & Botta-Dukát, Z. Silhouette width using generalized mean—A flexible method for assessing clustering efficiency. *Ecol Evol* **9**, 13231–13243 (2019).
9. Wallace, D. L. A Method for Comparing Two Hierarchical Clusterings: Comment. *Journal of the American Statistical Association* **78**, 569–576 (1983).
10. Severiano, A., Pinto, F. R., Ramirez, M. & Carriço, J. A. Adjusted Wallace Coefficient as a Measure of Congruence between Typing Methods. *J. Clin. Microbiol.* **49**, 3997–4000 (2011).
11. Holt, K. E. *et al.* Genomic analysis of diversity, population structure, virulence, and antimicrobial resistance in Klebsiella pneumoniae, an urgent threat to public health. *Proc Natl Acad Sci U S A* **112**, E3574-81 (2015).

12. Marakeby, H. *et al.* A system to automatically classify and name any individual genome-sequenced organism independently of current biological classification and nomenclature. *PLoS One* **9**, e89142 (2014).