

Kaptive Tutorial

Klebsiella 2024

Kaptive developers contacts:

kelly.wyres@monash.edu

tom.stanton@monash.edu

Trainers contacts:

Mary.gathoni@lshtm.ac.uk


Zoe.dyson@lshtm.ac.uk

Overview

In this tutorial, we will explore Kaptive tool and databases for the identification of surface polysaccharide synthesis loci from bacterial whole genome sequences. We will focus on *Klebsiella pneumoniae* and its close relatives in the *K. pneumoniae* Species Complex (KpSC). We will briefly introduce the biology of KpSC capsule and outer lipopolysaccharide loci and demonstrate how to install and run Kaptive (using the command line). We will explore some example Kaptive outputs of the genomes published as part of the BARNARDS study of neonatal sepsis (Sands *et al.*, 2021), and the European survey of carbapenemase-producing Enterobacteriaceae (EuSCAPE) project (Grundmann *et al.*, 2017). Finally, we will demonstrate how we can use third-party tools to further investigate specific capsule loci of interest e.g. putative novel loci.

Introduction

Surface polysaccharide synthesis loci

Like other human pathogens, members of the KpSC produce immunogenic surface polysaccharides which protect against host defences, phages and desiccation. In recent years there has been a resurgence of interest in these polysaccharides as targets for anti-KpSC vaccines . In order to design effective vaccines, we need to understand the underlying diversity and epidemiology of polysaccharide variants in the population. We can now achieve this goal at scale by predicting surface polysaccharide antigens from rapidly growing collections of whole genome sequences.

K antigen and locus

The capsular polysaccharide (CPS), or K-antigen, is an extracellular layer of polysaccharide chains, anchored to the outer-membrane (OM). The K-antigen is synthesised and exported by the Wzy pathway for which the associated genes are co-located in a gene cluster known as the K-locus. Each KpSC genome contains a single K-locus between 10 and 30 Kbp in length with a set of conserved genes located at the ends of the locus and encoding the capsular export machinery (*galF*, *cpsACP*, *wzi*, *wza*, *wzb*, *wzc*, *ugd*). The centre of the locus contains the genes required for the capsule-specific sugar synthesis and assembly. This central region can vary substantially between isolates and to-date 163 distinct loci have been defined on the basis of gene content. Each variant is thought to result in the expression of a distinct capsule polysaccharide, however at present just 77 capsule types have been defined serologically (although there are ongoing efforts to define additional types so this number will likely change over the next few years). As a result, a significant proportion of KpSC clinical isolates are non-typeable via serological techniques.

A note on KpSC capsule antigen and locus nomenclature: Each KpSC capsule antigen variant that has been defined phenotypically is referred to as a K-type and labelled Kx where x is an integer e.g. K1, K2, K3 etc. Each distinct capsule synthesis locus variant is known as a K-locus and labelled KLx e.g. KL1, KL2, KL3, KL102, KL157 etc, which may or may not be associated with a defined serological type. K-

locus numbers below 100 indicate the loci associated with the 77 historically defined K-types e.g. KL1 is the locus resulting in the expression of the K1 capsule antigen, KL2 results in expression of the K2 antigen etc (excludes any new types that may be defined as part of ongoing works). K-locus numbers above 100 indicate loci defined on the basis of gene content for which there were no historically defined serological types.

O antigen and locus

The outer lipopolysaccharide (LPS), or O-antigen, locus is located next to the K-locus. Each KpSC genome contains a single O-locus, for which X variants have been defined to-date. Unlike the K antigen and locus, there is not a simple one-to-one relationship between the O loci and O antigen types. Instead, the O1 and O2 antigens and the associated O2 sub-types are determined by combination of the O locus variant (one of three possible variants labelled O1/O2v1, O1/O2v2 and O1/O2v3) and genes located elsewhere in the KpSC genome. Kaptive v1 was able to distinguish and report O1 and O2 types but did not report the O2 sub-types. The latter capability was added to Kaptive v2.

Kaptive

Kaptive v1 was developed in 2016 for KpSC capsule locus typing ([Wyres et al., 2016](#)). In 2018 Kaptive was updated to include KpSC O locus typing and was deployed as an interactive online tool known as Kaptive Web ([Wick et al., 2018](#)). Kaptive v2 ([Lam, Wick, Judd, et al., 2021](#)), was released in 2021, which differs from Kaptive v1 because it reports K and O loci as well as K and O antigen predictions, and can distinguish the O2 antigen subtypes. Kaptive 3 is the most recent version, and it was released in 2024 ([Tom et al., 2024](#))

In this tutorial, we will be demonstrating Kaptive v3.

Kaptive 3

Kaptive is a system for surface polysaccharide typing from bacterial genome sequences. It consists of two main components:

1. Curated reference [databases](#) of surface polysaccharide gene clusters (loci).
2. A command-line interface (CLI) with three modes:
 - **assembly**: surface polysaccharide typing from assemblies
 - **extract**: extract features from Kaptive databases in different formats
 - **convert**: convert Kaptive results to different formats

Kaptive for KpSC surface antigen and locus typing is available as a standalone software tool and is implemented by both Kleborate (Lam, Wick, Watts, *et al.*, 2021) and PathogenWatch (Argimón *et al.*, 2021)

Kaptive method

kaptive assembly

For each input assembly, Kaptive runs the `typing_pipeline` which does the following:

1. Aligns locus gene nucleotide sequences to the assembly contig sequences using minimap2.
2. Identifies the best matching locus type using the [scoring algorithm](#).
3. Extracts the locus gene sequences from the assembly contig sequences.
4. Predicts the serotype/phenotype based on the gene content.

Scoring algorithm

1. A matrix is initialised with one row per locus and one column per score metric.

2. For each gene found, the best alignment is chosen, and the scores are added to the respective locus row if the coverage passes the threshold (**--min-cov**) .
3. The matrix is then weighted by the **--weight-metric** and the scores are selected from the column corresponding to the **--score-metric** .
4. The top N loci (**--n-best**) are selected to be fully aligned to the assembly.
5. Steps 1 and 2 are repeated and the best locus is selected from the column corresponding to the **--score-metric** .

The score metric can be explicitly set by the flag **--score-metric**, the options are:

- **0 (AS)** - Sum of the alignment scores per query
- **1 (mlen)** - Sum of the number of matching bases in the alignment per query
- **2 (blen)** - Sum of the number of aligned bases per query
- **3 (q_len)** - Sum of the number of bases of each query found (regardless of whether they are aligned)

The weighting can be explicitly set by the flag **--weight-metric**; the options are:

- **0** - No weighting
- **1** - Number of genes found
- **2** - Number of genes expected
- **3** - Proportion of genes found
- **4** - Sum of the number of aligned bases per query (blen)
- **5** - Sum of the number of bases of each query found (regardless of whether they are aligned) (q_len)

Locus reconstruction

After the best matching locus type has been identified, Kaptive will:

1. For each contig, the ranges from the full-length locus alignments of the best match are extracted.
2. The ranges are merged if they are within the distance of the largest locus in the database.
3. The merged ranges are used to create **LocusPiece** objects, and the sequence is extracted from the assembly contig.
4. Gene alignments are culled twice to determine the gene content:
 1. The first removes alignments overlapping genes from the best match.
 2. The remaining alignments that are not part of the best match are culled so that the best alignment is kept, and each alignment represents the best gene for that part of the contig.
5. For each remaining gene alignment, the gene is then evaluated.

Gene evaluation

For each **GeneResult** object, Kaptive will:

1. Check whether the gene is **partial** by determining if the gene overlaps the contig boundaries.
2. Extract the DNA sequence from the assembly contig and translate to amino acid.
3. Perform pairwise alignment to the reference gene amino acid sequence and calculate percent identity.
4. Check for **truncation** by determining if the amino acid sequence length is <95% of the reference gene protein length.
5. Determine whether the gene belongs to the biosynthetic gene cluster (**inside locus**) or not (**outside locus**).

Phenotype prediction

As of Kaptive 3, have the ability to predict the resulting phenotype of the assembly. This is similar to how the *Type* was reported in previous versions, but now includes the ability to predict specific phenotypes based on known mutations/modifications in a given set of locus genes as defined in the database [logic file](#).

Confidence score

Kaptive will finally calculate how confident it is in the prediction, which is explained below.

Typeable loci

The locus was found in a single piece in the query assembly with no genes below the minimum translated identity according to the [locus thresholds](#) and:

- no missing genes (as determined by at least 50% nucleotide mapping coverage)
- no unexpected genes (genes from other loci) inside the locus region of the assembly

OR

The locus was found in more than one piece in the query assembly with no genes below the minimum translated identity according to the [locus thresholds](#) and:

- no more than 1 missing gene
- no more than 1 unexpected gene (genes from other loci) inside the locus region of the assembly

Untypeable loci

These are loci that do not meet the above criteria. **We recommend that users do not accept these results** unless they are able to perform manual exploration of the data.



Database distributed under Kaptive3

When Kaptive is installed, it may be difficult to find the databases in the file system. However, each `<database>` argument in the Kaptive CLI accepts either a path to a Genbank file or a keyword that refers to a database distributed with Kaptive. The keywords are listed below.

Database	Keywords
<i>Klebsiella pneumoniae</i> K locus primary reference database	kpsc_k kp_k k_k
<i>Klebsiella pneumoniae</i> K locus variant reference database	kpsc_k_variant kp_k_variant k_k_variant
<i>Klebsiella pneumoniae</i> O locus primary reference database	kpsc_o kp_o k_o
<i>Acinetobacter baumannii</i> K locus primary reference database	ab_k
<i>Acinetobacter baumannii</i> OC locus primary reference database	ab_o

Refer to [Kaptive documentation](#) for more information about databases.

How to install Kaptive

We will demonstrate how to install and run Kaptive using a command-line computing environment.

Kaptive and Kleborate can be installed for use on your personal computer or cluster. They rely on the use of third-party dependencies including [Biopython](#) and [Minimap2](#). To reduce the chances of dependency-related errors, we highly recommend that these tools be used within a virtual environment with [conda](#) 🐍.

Kaptive requires the following software and libraries.

- [Python](#) >=3.9
- [Biopython](#) >=1.83
- [minimap2](#)
- [DNA Features Viewer](#)

Set up a conda environment

Create an environment for both Kaptive and Kleborate. We will call this environment **klebsiella_analysis** with the **-n/-name** flag, specify the Python version as 3.9 and list our dependencies for installation.

N.B. Kleborate has the additional dependency of [mash](#), so we'll add this to the list of software to install.

```
conda create -n klebsiella_analysis python=3.9 biopython
minimap2 mash -y
```

Once your environment has been installed successfully, we need to ‘activate’ it and run the Installation from PyPI

```
conda activate klebsiella_analysis
pip install kaptive
```



How to use Kaptive

To see the full list of commands and options, run `kaptive -h/--help`.

```
usage: kaptive <command>
```

```
KAPTIVE
```

In silico serotyping

:

<code>assembly</code>	In silico serotyping of assemblies
<code>extract</code>	Extract entries from a Kaptive database
<code>convert</code>	Convert Kaptive results into different formats

:

<code>-V, --verbose</code>	Print debug messages to stderr
<code>-v, --version</code>	Show version number and exit
<code>-h, --help</code>	Show this help message and exit



Exercise 1

K Locus typing using Kaptive

In this exercise, we will perform K locus typing on *Klebsiella pneumoniae* genomes published as part of the BARNARDS study of neonatal sepsis, and the European survey of carbapenemase-producing Enterobacteriaceae (EuSCAPE) project.

Instructions

Klebsiella pneumoniae assemblies (in FASTA format) are found on [Google drive](#).

Download them on your laptop

Run Kaptive using the following command:

```
cd kaptive_workshop_data

kaptive assembly kpssc_k *.fasta -o kaptive_results.tsv -f -p
```

Explanation of the command:

- **assembly**: Specifies that Kaptive will perform typing on assemblies.
- **kpssc_k**: This is the database keyword for the *Klebsiella pneumoniae* K locus database provided by Kaptive.
- ***.fasta**: Matches all FASTA files in the directory.
- **-f**: Turn on fasta output
- **-p**: plots the results
- **-o kaptive_results.tsv**: Outputs the results into a tab-delimited file named kaptive_results.tsv.

The command will generate a tabular file (kaptive_results.tsv) containing each assembly's K locus typing results.

Note:

[Database keywords](#) like `kpsc_k`, `kpsc_o` are a convenient way to access pre-distributed databases with Kaptive. They are located in the `reference_databases` directory. Alternatively, you can specify the full path to your custom database if needed.

Kaptive output files

The main output of the assembly typing mode is a tab-delimited table of the results with the following columns:

Column name	Description
Assembly	The name of the input assembly, taken from the assembly filename.
Best match locus	The locus type which most closely matches the assembly.
Best match type	The predicted serotype/phenotype of the assembly.
Match confidence	Typeable or Untypeable
Problems	Characters indicating issues with the locus match (see problems).
Identity	Weighted percent identity of the best matching locus to the assembly.
Coverage	Weighted percent coverage of the best matching locus in the assembly.
Length discrepancy	If the locus was found in a single piece, this is the difference between the locus length and the assembly length.
Expected genes in locus	A fraction indicating how many of the genes in the best matching locus were found in the locus part of the assembly.
Expected genes in locus, details	Gene names for the expected genes found in the locus part of the assembly.
Missing expected genes	A string listing the gene names of expected genes that were not found.
Other genes in locus	The number of unexpected genes (genes from loci other than the best match) which were found in the locus part of the assembly.

Other genes in locus, details	Gene names for the other genes found in the locus part of the assembly.
Expected genes outside locus	A fraction indicating how many of the expected genes which were found in the assembly but not in the locus part of the assembly (usually zero)
Expected genes outside locus, details	Gene names for the expected genes found outside the locus part of the assembly.
Other genes outside locus	The number of unexpected genes (genes from loci other than the best match) which were found outside the locus part of the assembly.
Other genes outside locus, details	Gene names for the other genes found outside the locus part of the assembly.
Truncated genes, details	Gene names for the truncated genes found in the assembly.
Extra genes, details	Gene names for the extra genes found in the assembly.

NB: If the summary table already exists, Kaptive will append to it (not overwrite it) and suppress the header line. This allows you to run Kaptive in succession on sets of assemblies, all outputting to the same table file.

Other Kaptive outputs

Fasta

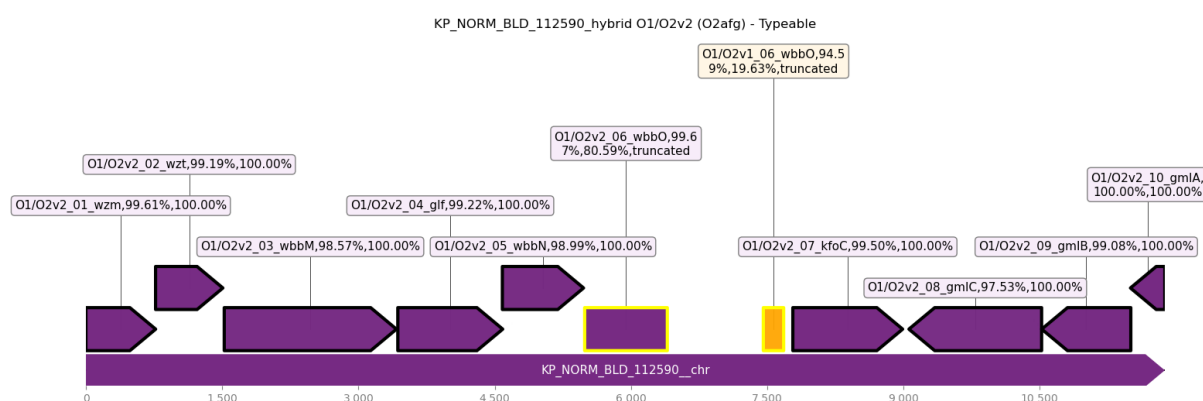
The `-f/--fasta` flag produces a fasta file of the region(s) of the assembly which correspond to the best locus match. This may be a single piece (in cases of a good assembly and a strong match) or it may be in multiple pieces (in cases of poor assembly and/or a novel locus).

JSON

The `-j/--json` flag produces a JSON file of the results which allows Kaptive to reconstruct the `TypingResult` objects after a run which can be used with [kaptive convert](#). Unlike the previous version (2 and below), this is a JSON lines file (or “-” for stdout), where each line is a JSON object representing the results for a single assembly.

Plot

Kaptive can now produce a visual representation of the locus match in the assembly. This is done using the `-p/--plot` flag, which produces a plot in the format specified by the `--plot-fmt` flag (default: png).



The plot is a visual representation of the locus genes and the corresponding assembly contig pieces.

- The gene transparencies are based on the protein percent identity.
- The contig transparencies are based on the final percent identity.
- Unexpected genes are shown in orange.
- Genes are blunt and outlined in yellow if they are truncated.
- Genes are outlined in red if they are below the gene identity threshold.

Interpreting the results

Here is the first few columns of the Kaptive output file from our example above, viewed in Microsoft Excel:

Assembly	Best match locus	Best match type	Match confidence	Problems	Identity	Coverage	Length discrepancy	Expected genes in locus
ERR1204817	KL15	K15	Typeable		99.92%	100.00%	1115 bp	16 / 16 (100.00%)
ERR1415571	KL2	K2	Typeable		99.80%	100.00%	1 bp	13 / 18 (72.22%)
ERR4920378	KL102	unknown (KL102)	Typeable	?2-	99.51%	79.91%	n/a	12 / 17 (70.59%)
ERR4920436	KL57	K57	Typeable	!	99.52%	100.00%	-1 bp	18 / 18 (100.00%)
ERR4920450	KL57	K57	Typeable	?2-!	99.52%	91.09%	n/a	17 / 18 (94.44%)
ERR4920551	KL112	unknown (KL112)	Typeable	?5-	99.52%	79.98%	n/a	16 / 18 (88.89%)

- **Best match locus** indicating the locus that is best supported by the available sequence data
- **Best match type** indicating the predicted phenotype based on that associated with the **Best match locus** and taking into account any special phenotype logic e.g. where a known essential gene is truncated, Kaptive will report the **Best match type** as 'Capsule null'.
- **Match confidence** indicates a qualitative level of confidence that the reported **Best match locus** is correct
- **Problems** indicates the features of the assembly locus that may impact **Match confidence**

Problems

- ? = the match was in multiple pieces, possibly due to a poor match or discontinuous assembly. The number of pieces is indicated by the integer directly following the ? symbol).
- - = genes expected in the locus were not found.
- + = extra genes were found in the locus.
- * = one or more expected genes was found but with translated identity below the minimum threshold.
- ! = one or more genes was found but truncated

Other columns

- Missing genes may indicate a novel locus or deletion variant is present in the assembly
- Length discrepancies can indicate a novel locus or deletion variant is present in the assembly. For *Klebsiella* K loci, positive length discrepancies of >700bp often indicate insertion sequence insertions resulting in so called 'IS variants' of the locus.
- Other genes inside the locus may indicate a novel locus
- Truncated genes may have an impact on the resultant phenotype. Kaptive will consider truncations when reporting predicting phenotypes, but it currently considers only gene truncations for which there is good supporting evidence in the literature, and such evidence is very limited.

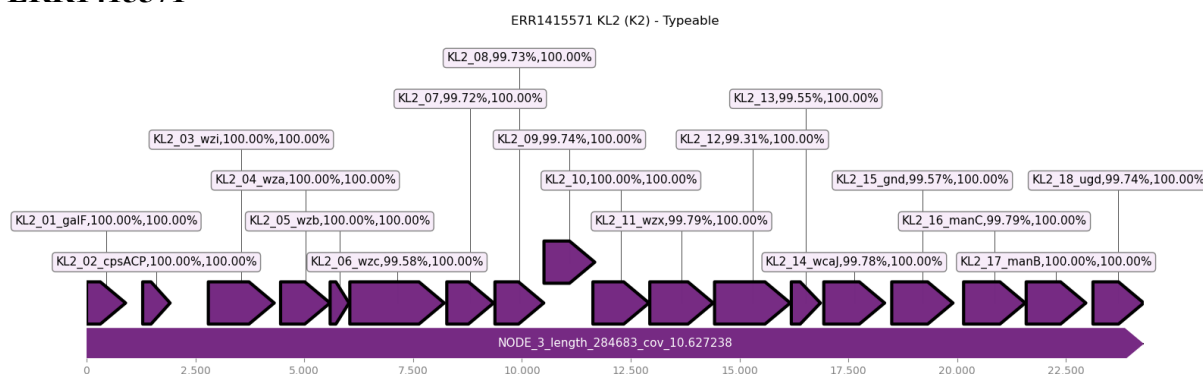
Exploring the results

We will now go through the columns in the Kaptive output .tsv file, using genome **ERR1415571** and **ERR4920436** as an example.

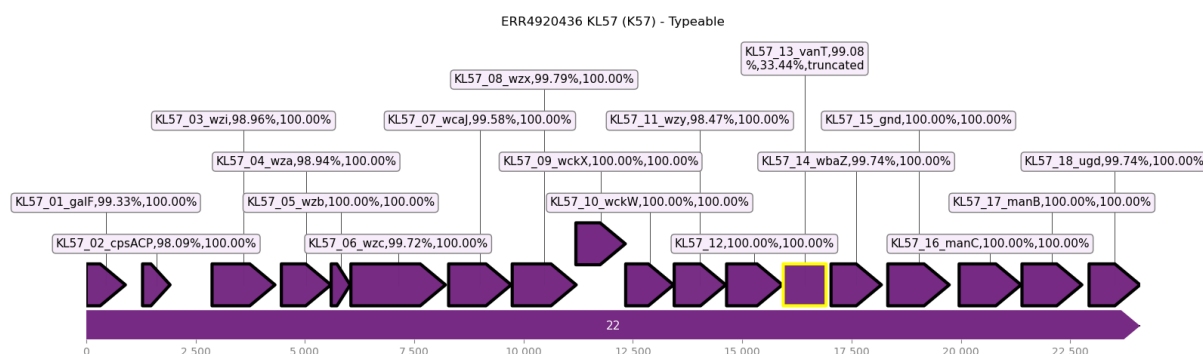
Match confidence

The **ERR1415571** and **ERR4920436** K-locus matches have the confidence call “Typeable” which means that the locus was found in a single piece with 100% coverage and 99% identity to the reference.

ERR1415571



ERR4920436



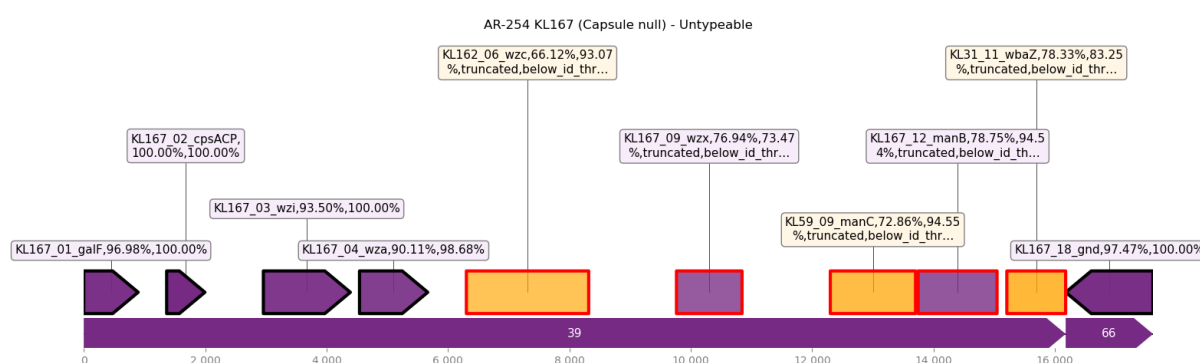
In these 2 assemblies we see that we have perfectly match known K loci: KL2 and KL57, respectively. The coverages are 100%, and identities 99% with 1 and -1 length discrepancies. The KL2 locus is known to encode the K2 serotype, as indicated as the 'Best match type', while the KL57 encodes for K57 serotype.

Examples of 'Untypeable' match

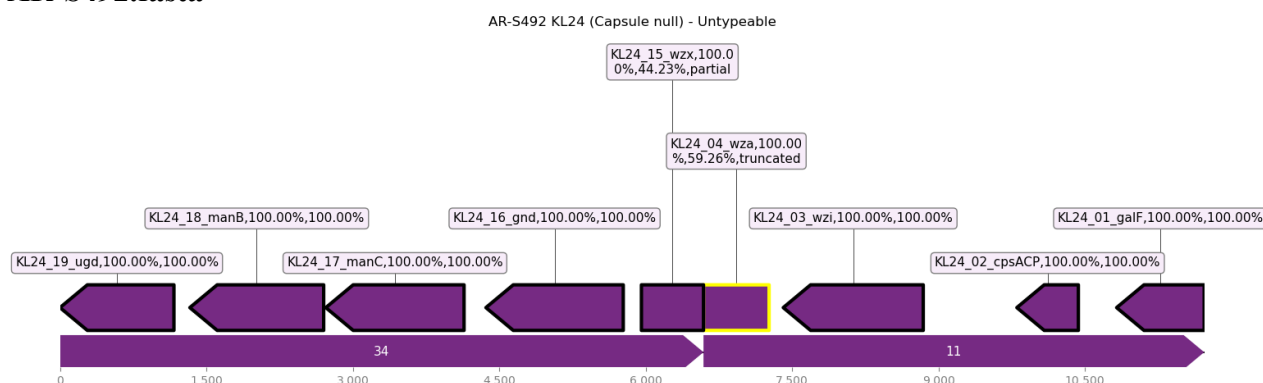
We will explore the results from 2 genomes

Assembly	Best match locus	Best match type	Match confidence	Problems	Identity	Coverage	Length discrepancy	Expected genes in locus
AR-254	KL167	Capsule null	Untypeable	?2-+*!	90.54%	34.85%	n/a	7 / 19 (36.84%)
AR-S492	KL24	Capsule null	Untypeable	?2-!	100.00%	42.91%	n/a	9 / 19 (47.37%)

AR-254.fasta



AR-S492.fasta



AR-254.fasta, the best match locus was KL167. Minimap2 found only found alignments for 34.85% of the KL167 sequence. 12 out of 19 (36.84%) KL167 genes were not found, and the locus is split across different assembly contigs.

AR-S492.fasta, the best match locus was KL24. Minimap2 found only found alignments 42.91% of the KL24 sequence. 10 out of 19 (47.37%) genes were not found.



These loci do not meet the required criteria. **The users are recommended not to accept these results** unless they are able to perform manual exploration of the data

Exercise 2: Analysis of novel loci

In this exercise, you will analyse three *Klebsiella variicola* assemblies from the study deposited under the project ID **PRJNA420954**. You will use Kaptive to perform K locus analysis on these assemblies, followed by a discussion of the results.

The *K. variicola* assemblies can be downloaded from [here](#)

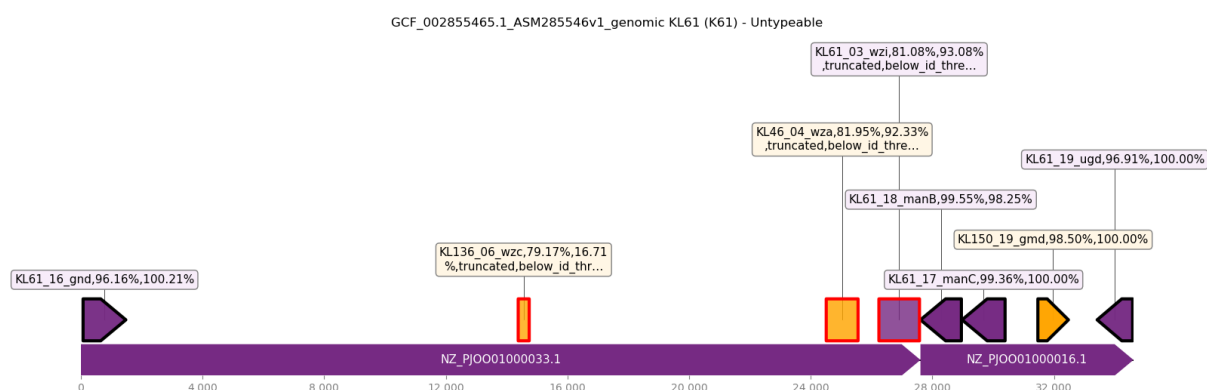
Download them on your laptop and run K Locus Analysis with Kaptive

```
kaptive assembly kpvc_k *.fna -o kaptive_k_locus_results.tsv
-f -p
```

Exploring the results.

Assembly	Best match locus	Best match type	Match confidence	Problems	Identity	Coverage	Length discrepancy	Expected genes in locus
GCF_002855465.1_ASM285546v1_genomic	KL61	K61	Untypeable	?2-+*!	94.61%	29.61%	n/a	5 / 19 (26.32%)
GCF_003988975.1_ASM398897v1_genomic	KL61	K61	Untypeable	?3-+*!	94.61%	29.61%	n/a	5 / 19 (26.32%)
GCF_003988985.1_ASM398898v1_genomic	KL61	K61	Untypeable	?3-+*!	94.61%	29.61%	n/a	5 / 19 (26.32%)

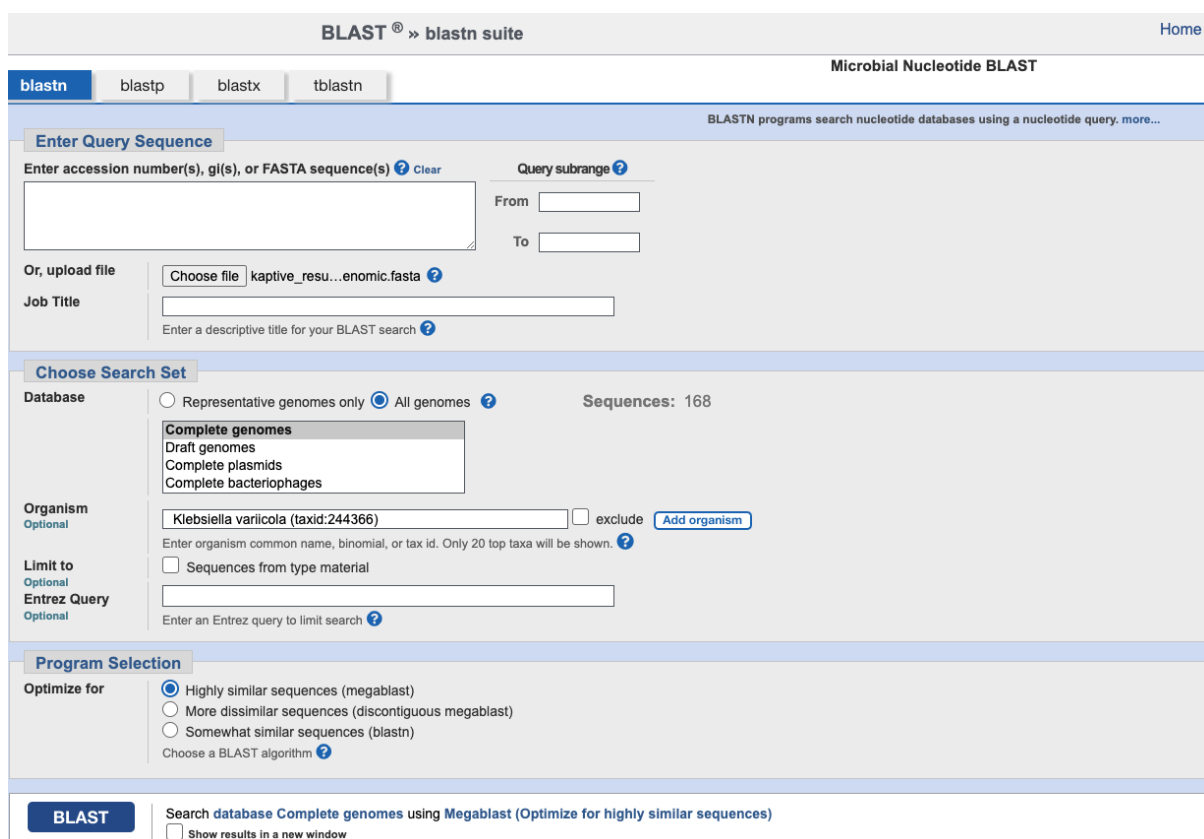
Based on the results, we can see that all our assemblies have match confidence of “**Untypeable**”, have many missing genes and the locus is in multiple pieces. The **Best match locus** was KL61 and there is high percent identity and coverage to the *gnd*, and *ugd*, genes from this locus. This is normal as these are amongst the most conserved genes of the K-locus.



Unfortunately, we cannot accept novel loci into the database unless they are in a **single piece** because we cannot be certain that we have captured the full locus sequence. The K-loci of all three of the assemblies from this study were split across multiple contigs 😞 If we're really determined to validate and submit our novel locus, we can try and find complete locus in similar assemblies.

Luckily, we have access to the largest repository for public bacterial genomes... the NCBI! We can take the fasta output from Kaptive and use this to perform a BLASTn against public *K. variicola* assemblies to try and find one that isn't broken.

Go to the [Microbial Nucleotide Blast webpage](#). This is a web-based blast search optimised for searching sequences against public microbial genomic sequences.



BLAST® » blastn suite Home

Microbial Nucleotide BLAST

BLASTn programs search nucleotide databases using a nucleotide query. [more...](#)

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s) [Clear](#) Query subrange [?](#)

From

To

Or, upload file Choose file kaptive_resu...enomic.fasta [?](#)

Job Title

Enter a descriptive title for your BLAST search [?](#)

Choose Search Set

Database Sequences: 168

☐ Representative genomes only ☒ All genomes [?](#)

Complete genomes

Draft genomes

Complete plasmids

Complete bacteriophages

Organism Optional

☐ exclude [Add organism](#)

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown. [?](#)

Limit to Optional

☐ Sequences from type material

Entrez Query Optional

Enter an Entrez query to limit search [?](#)

Program Selection

Optimize for

☒ Highly similar sequences (megablast)

☐ More dissimilar sequences (discontiguous megablast)

☐ Somewhat similar sequences (blastn)

Choose a BLAST algorithm [?](#)

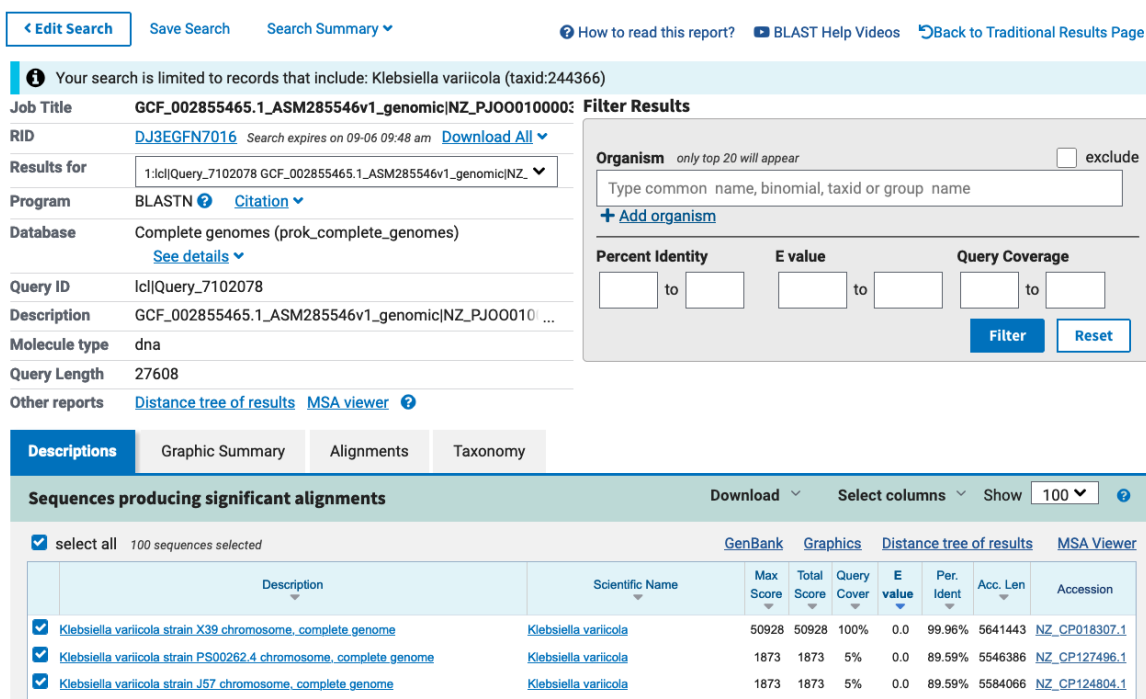
BLAST Search database **Complete genomes** using **Megablast** (Optimize for highly similar sequences)

☐ Show results in a new window

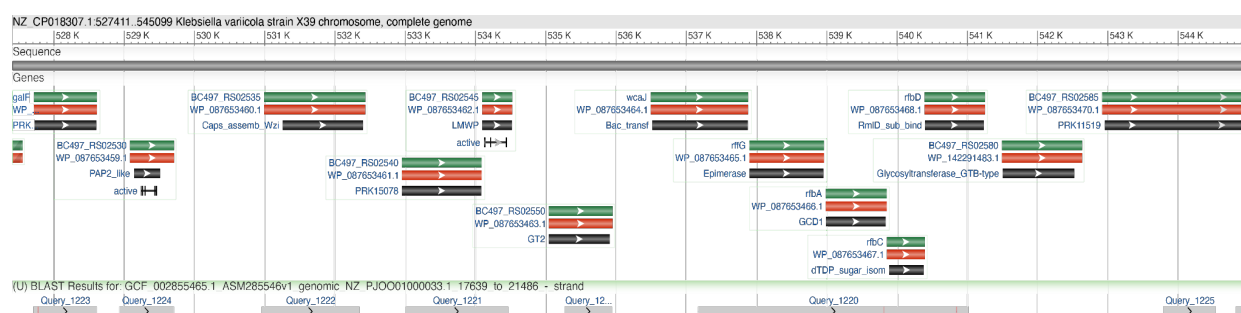
Click **Choose File** and select the .fna file from Kaptive results **GCF_002855465.1_ASM285546v1_genomic_kaptive_results.fna**. You will see that **Complete Genomes** is already highlighted. This is perfect because we know these assemblies are complete (single contig per amplicon) so the K-locus is likely to be

intact. As the assemblies from the publication are *K. variicola*, we can type this into the **Organism** box.

We can leave all the other settings as they are for now and hit the **BLAST** button.



Wow, one [public *K. variicola* genome](#) has 100% coverage and 99.96% identity to our putative novel locus nucleotide sequence! Let's check out the alignment visualisation.

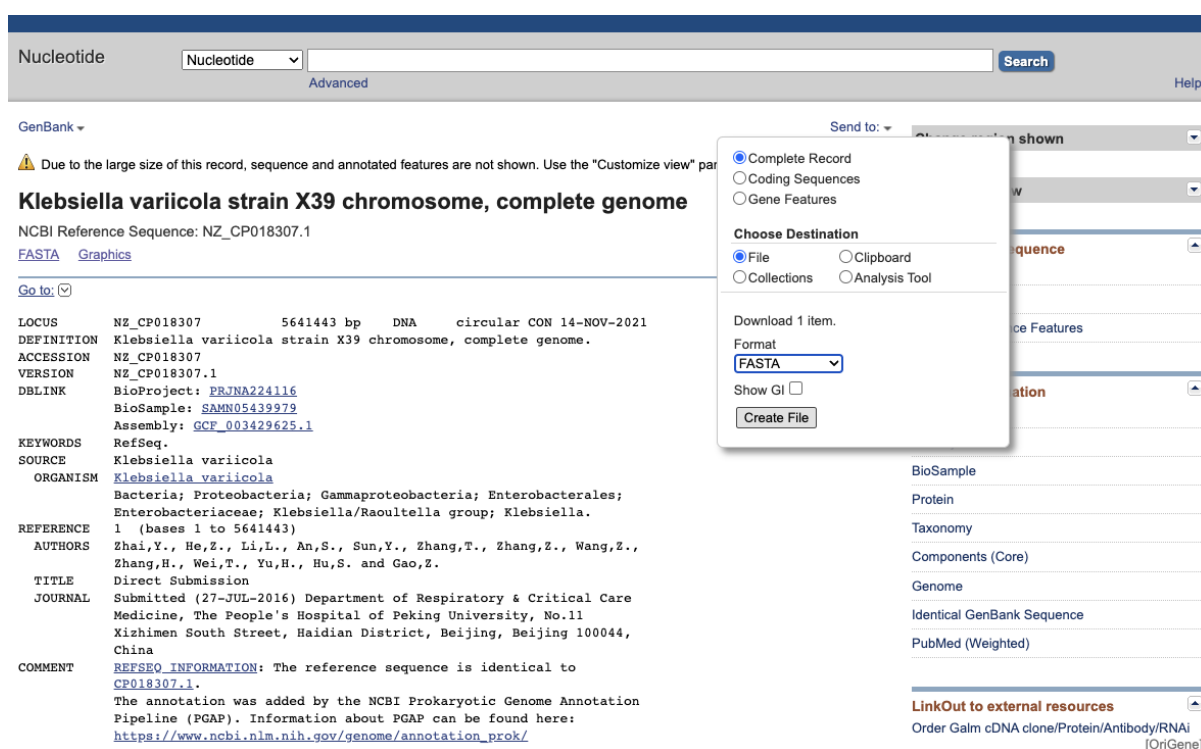


Nice, our novel locus pieces align nicely to the K-locus of this assembly, and it is not broken up over multiple contigs! We can now download this assembly and run Kaptive on it as before to extract the novel locus sequence. If you would like to try this on the command line from the [genome FTP directory](#), you can use the following command:

curl

```
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/003/429/625/GCF_003429625.1_ASM342962v1/GCF_003429625.1_ASM342962v1_genomic.fna.gz | gunzip -c > GCF_003429625.1_ASM342962v1_genomic.fna
```

To download the assembly manually, head back to the **BLAST results**, click on the **Descriptions** tab, and select the [accession](#) of the **top hit**. Once on the Nucleotide record for this [genome](#), click on ‘**Send to**’, make sure **Complete Record** and **File** are ticked and the format is **FASTA**, then click on ‘**Create file**’ and save in a sensible location. You can also find the assembly fasta file here.



Nucleotide Advanced Help

GenBank Send to:

Klebsiella variicola strain X39 chromosome, complete genome

NCBI Reference Sequence: NZ_CP018307.1

[FASTA](#) [Graphics](#)

Go to:

LOCUS NZ_CP018307 5641443 bp DNA circular CON 14-NOV-2021

DEFINITION Klebsiella variicola strain X39 chromosome, complete genome.

ACCESSION NZ_CP018307

VERSION NZ_CP018307.1

DBLINK BioProject: [PRJNA224116](#)
BioSample: [SAMN05439979](#)
Assembly: [GCF_003429625.1](#)

KEYWORDS RefSeq.

SOURCE [Klebsiella variicola](#)
ORGANISM [Klebsiella variicola](#)
Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacterales; Enterobacteriaceae; Klebsiella/Raoultella group; Klebsiella.

REFERENCE 1 (bases 1 to 5641443)

AUTHORS Zhai,Y., He,Z., Li,L., An,S., Sun,Y., Zhang,T., Zhang,Z., Wang,Z., Zhang,H., Wei,T., Yu,H., Hu,S. and Gao,Z.

TITLE Direct Submission

JOURNAL Submitted (27-JUL-2016) Department of Respiratory & Critical Care Medicine, The People's Hospital of Peking University, No.11 Xizhimen South Street, Haidian District, Beijing, Beijing 100044, China

COMMENT [REFSEQ INFORMATION](#): The reference sequence is identical to [CP018307.1](#).
The annotation was added by the NCBI Prokaryotic Genome Annotation Pipeline (PGAP). Information about PGAP can be found here: https://www.ncbi.nlm.nih.gov/genome/annotation_prok/

Download 1 item.

Format

Show GI ☐

BioSample

Protein

Taxonomy

Components (Core)

Genome

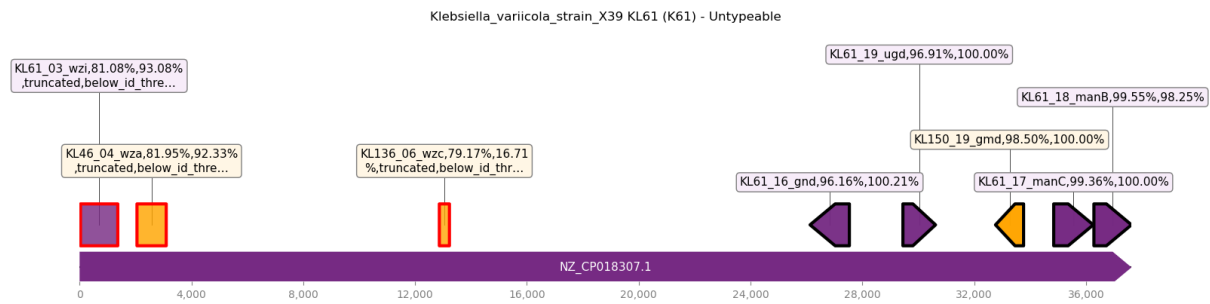
Identical GenBank Sequence

PubMed (Weighted)

[LinkOut to external resources](#)

Order Galm cDNA clone/Protein/Antibody/RNAi [OriGene]

Let's run Kaptive again on our complete assembly to see if we extract the full-length putative novel K-locus.



Success! 🙌 . From the results, We can see that we have the same locus as the other neonatal sepsis isolates, and it is contained within a single contig.

NB: If you have a new locus you can submit to [Kelly](#) and [Tom](#) for validation and inclusion in the database

References

Argimón, S. *et al.* (2021) 'Rapid Genomic Characterization and Global Surveillance of *Klebsiella* Using Pathogenwatch', *Clinical Infectious Diseases*, 73(Supplement_4), pp. S325–S335. Available at: <https://doi.org/10.1093/cid/ciab784>.

BARNARDS Group *et al.* (2021) 'Characterization of antimicrobial-resistant Gram-negative bacteria that cause neonatal sepsis in seven low- and middle-income countries', *Nature Microbiology*, 6(4), pp. 512–523. Available at: <https://doi.org/10.1038/s41564-021-00870-7>.

Camacho, C. *et al.* (2009) 'BLAST+: architecture and applications', *BMC Bioinformatics*, 10(1), p. 421. Available at: <https://doi.org/10.1186/1471-2105-10-421>.

Carver, T. *et al.* (2008) 'Artemis and ACT: viewing, annotating and comparing sequences stored in a relational database', *Bioinformatics*, 24(23), pp. 2672–2676. Available at: <https://doi.org/10.1093/bioinformatics/btn529>.

Carver, T. *et al.* (2012) 'Artemis: An integrated platform for visualization and analysis of high-throughput sequence-based experimental data', *Bioinformatics*, 28(4), pp. 464–469. Available at: <https://doi.org/10.1093/bioinformatics/btr703>.

Carver, T.J. *et al.* (2005) 'ACT: the Artemis comparison tool', *Bioinformatics*, 21(16), pp. 3422–3423. Available at: <https://doi.org/10.1093/bioinformatics/bti553>.

Cock, P.J.A. *et al.* (2009) 'Biopython: freely available Python tools for computational molecular biology and bioinformatics', *Bioinformatics*, 25(11), pp. 1422–1423. Available at: <https://doi.org/10.1093/bioinformatics/btp163>.

Farzana, R. *et al.* (2018) 'Outbreak of Hypervirulent Multi-Drug Resistant *Klebsiella variicola* causing high mortality in neonates in Bangladesh', *Clinical Infectious Diseases*, 68(7), pp. 1225–1227. Available at: <https://doi.org/10.1093/cid/ciy778>.

Gilchrist, C.L.M. and Chooi, Y.-H. (2021) 'clinker & clustermap.js: automatic generation of gene cluster comparison figures', *Bioinformatics*. Edited by P. Robinson, 37(16), pp. 2473–2475. Available at: <https://doi.org/10.1093/bioinformatics/btab007>.

Gorrie, C.L. *et al.* (2022) 'Genomic dissection of *Klebsiella pneumoniae* infections in hospital patients reveals insights into an opportunistic pathogen', *Nature Communications*, 13(1), p. 3017. Available at: <https://doi.org/10.1038/s41467-022-30717-6>.

van der Graaf – van Bloois, L. *et al.* (2021) *Development of Kaptive databases for Vibrio parahaemolyticus O- and K-antigen serotyping*. preprint. Microbiology. Available at: <https://doi.org/10.1101/2021.07.06.451262>.

Lam, M.M.C., Wick, R.R., Watts, S.C., *et al.* (2021) 'A genomic surveillance framework and genotyping tool for *Klebsiella pneumoniae* and its related species complex', *Nature Communications*, 12(1), p. 4188. Available at: <https://doi.org/10.1038/s41467-021-24448-3>.

Lam, M.M.C., Wick, R.R., Judd, L.M., *et al.* (2021) *Kaptive 2.0: updated capsule and LPS locus typing for the Klebsiella pneumoniae species complex*. preprint. Genomics. Available at: <https://doi.org/10.1101/2021.11.05.467534>.

Wick, R.R. *et al.* (2015) 'Bandage: interactive visualization of *de novo* genome assemblies: Fig. 1.', *Bioinformatics*, 31(20), pp. 3350–3352. Available at: <https://doi.org/10.1093/bioinformatics/btv383>.

Wick, R.R. *et al.* (2018) 'Kaptive Web: User-Friendly Capsule and Lipopolysaccharide Serotype Prediction for *Klebsiella* Genomes', *Journal of Clinical Microbiology*. Edited by D.J. Diekema, 56(6), pp. e00197-18. Available at: <https://doi.org/10.1128/JCM.00197-18>.

Wyres, K.L. *et al.* (2016) 'Identification of *Klebsiella* capsule synthesis loci from whole genome data', *Microbial Genomics*, 2(12). Available at: <https://doi.org/10.1099/mgen.0.000102>.

Wyres, K.L. *et al.* (2020) 'Identification of *Acinetobacter baumannii* loci for capsular polysaccharide (KL) and lipooligosaccharide outer core (OCL) synthesis in genome



assemblies using curated reference databases compatible with Kaptive', *Microbial Genomics*, 6(3). Available at: <https://doi.org/10.1099/mgen.0.000339>.