KrakenHLL: Confident and fast metagenomics classification using unique k-mer counts

Breitwieser FP[1] and Salzberg SL[1,2]

1 Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins School of Medicine, Baltimore, MD, United States

2 Departments of Biomedical Engineering, Computer Science and Biostatistics, Johns Hopkins University, Baltimore, MD, United States

Abstract

Motivation: False positive identifications are a significant problem in metagenomics. Spurious identifications can attract many reads that often aggregate in the genomes. Genome coverage may be used to filter false positives, but fast k-mer based metagenomic classifiers only provide read counts as metrics, and re-alignment is expensive. We propose using k-mer coverage, which can be computed during classification, as proxy for genome base coverage.

Results: We present KrakenHLL, a metagenomics classifier that records the number of unique k-mers as well as coverage for each taxon. KrakenHLL is based on the ultra-fast classification engine Kraken and combines it with HyperLogLog cardinality estimators. We demonstrate that more false-positive identifications can be filtered using the unique k-mer count, especially when looking at species of low abundance. Further enhancements include mapping against multiple databases, plasmid and strain identification using an extended taxonomy, and inclusion of over 100,000 additional viral strain sequences. KrakenHLL runs as fast as Kraken, and sometimes faster.

Availability and Implementation: KrakenHLL is implemented in C++ and Perl, and available under the GPL v3 license at https://github.com/fbreitwieser/krakenhll.

Contact: florian.bw@gmail.com.

Introduction

Metagenomic classifiers attempt to assign taxon identifiers to each read in a sample. Typically, this is done using mapping rather than alignment, which returns the read classifications but not the aligned positions in the genomes (as reviewed by Breitwieser, et al., 2017). However, read counts can be deceiving. Sequence contamination of the samples - introduced from laboratory kits or the environment during sample extraction, handling or sequencing - can yield high numbers of spurious identifications (Salter, et al., 2014; Thoendel, et al., 2017). Having only small amounts of input material can further compound the problem of contamination. In clinical diagnosis of infectious diseases, for example, often less than 0.1% of the DNA sequenced is from microbes of interest (Brown, et al., 2018; Salzberg, et al., 2016). Furthermore, spurious matches can result from low-complexity regions of genomes, and contamination in the database genomes themselves (Mukherjee, et al., 2015).

Such false positive reads typically match only small portions of the genome. Reads from microbes that are truly present should distribute relatively uniformly across the genome rather than be concentrated in one or a few locations. Genome alignment can reveal this information. However, it is resource intensive, requires the selection of specific genomes, and it is difficult to extrapolate from the alignment of one genome to higher levels in the taxonomic tree. Some metagenomics methods use coverage information for better mapping or quantification, but usually require results from much slower alignment methods as input (Dadi, et al., 2017).

46 Notably, assembly-based methods also work, but only for highly abundant species (Quince, et
47 al., 2017).
48
49 Here, we present KrakenHLL, a novel method that combines fast k-mer based classification with
50 fast k-mer cardinality estimation. KrakenHLL is based on the Kraken metagenomics classifier
51 (Wood and Salzberg, 2014) and implements fast counting of the number of unique k-mers
52 identified for each taxon using the efficient probabilistic cardinality estimation algorithm
53 HyperLogLog (Ertl, 2017; Flajolet, et al., 2007; Heule, et al., 2013). The count and percentage of
54 the taxon's unique k-mers in the database that are covered by read k-mers can be used to discern
55 false positive from true-positive sequences. Furthermore, KrakenHLL implements other new
56 features for better metagenomics classifications: (a) searches can be done against multiple
57 databases hierarchically, (b) the taxonomy can be extended to include nodes for strains and
58 plasmids, thus enabling their detection, and (c) database build script enables adding over 100
59 thousand viral strains from the NCBI Viral Genome Resource (Brister, et al., 2015). Notably,
60 KrakenHLL, which provides a superset of the information of Kraken, is as fast or faster than
61 Kraken while using very little additional memory during classification.

62 **Results**

63 KrakenHLL was developed to provide efficient k-mer coverage information for all taxa
64 identified in a metagenomics experiment. The main workflow is as follows: As reads are
65 processed, each k-mer is assigned a taxa from the database (Figure 1 (A)). KrakenHLL
66 instantiates a HyperLogLog data sketch for each taxon, and adds the k-mers to it (Figure 1 (B)).
67 After classification, KrakenHLL traverses up the taxonomic tree and merges the estimators of the
68 child taxa to the parent. KrakenHLL reports the number of unique k-mers, and the breadth and

depth of k-mer coverage for each taxon in the taxonomic tree in the classification report (Figure 1 (C)).

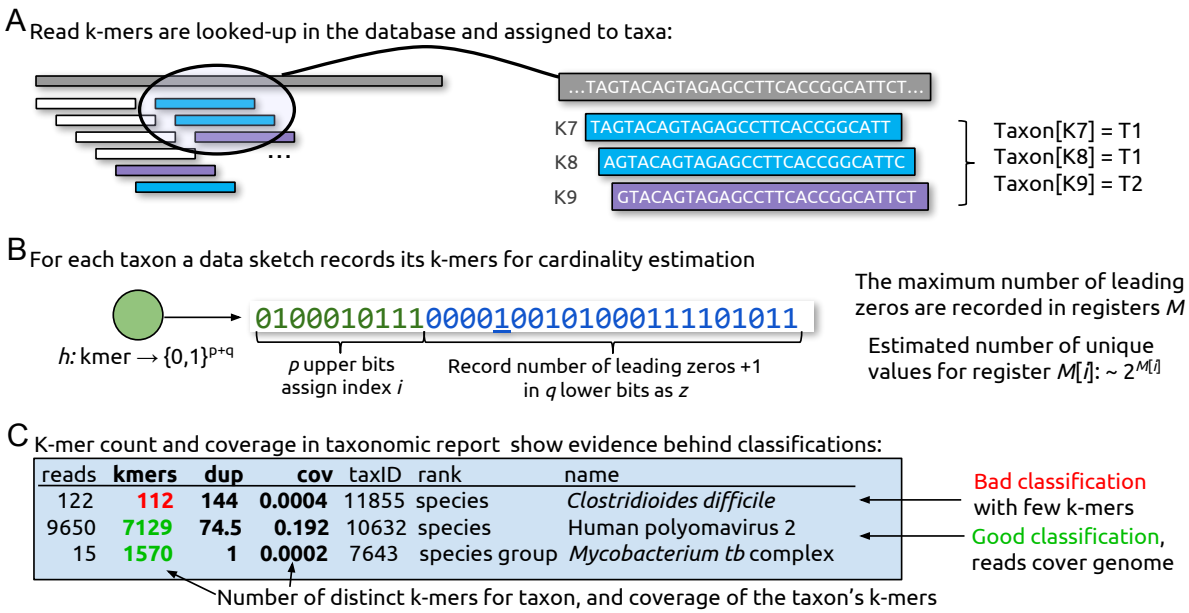


Figure 1: KrakenHLL algorithm and report. (A) The taxon mappings for each k-mer of a read are queried from the database. For each taxon, a unique k-mer counter is instantiated, and the observed k-mers are added to it. (B) Unique k-mer counting is implemented with the probabilistic estimation method HyperLogLog (HLL) with below 1% error in 16KB of memory per counter. (C) The number of unique k-mers, duplicity (average time each k-mer has been seen) and coverage are reported for each taxon in the taxonomic tree, enabling assessment of the classification.

Efficient k-mer cardinality estimation with HyperLogLog algorithm

83 Exact counting of the number of unique values (cardinality) in the presence of duplicates
84 requires memory proportional to the cardinality. Very accurate *estimation* of the cardinality,
85 however, can be achieved using only a small amount of fixed space. The HyperLogLog
86 algorithm (HLL), originally described by (Flajolet, et al., 2007), is currently one of the most
87 efficient cardinality estimators, and lends itself to k-mer counting (Irber Junior and Brown,
88 2016). The main idea behind the method is that long runs of leading zeros are unlikely in random
89 hashes. E. g., it's expected to see every fourth hash start with one 0-bit before the first 1-bit
90 (01_2), and every 32nd hash starts with 00001_2. The algorithm saves a sketch of observed data
91 based on hashes of the k-mers in 2^p one byte registers (in our implementation), where p is the
92 precision parameter. The relative error of the estimate is 1/sqrt(2^p). With p=14, the sketch uses
93 2^{14} one-byte registers, i.e. 16KB of space and has a relative error less than 1% (Figure 2).
94
95 Generating the sketch: Each k-mer is first hashed into a 64-bit string H. The sketch starts out in
96 sparse representation which has an effective p of 25, using 4 bytes per element. See (Heule, et
97 al., 2013) for more details on the encoding. Once m/4 distinct elements have been observed, we
98 switch to the standard representation of (Flajolet, et al., 2007): The first p bits of H are used as
99 index i into the registers M. The later 64-p=q bits are used to define the rank based on the
100 position of the first 1-bit (or, equivalently, the count of leading zeros plus one). If all q bits are
101 zero, the rank is q+1. The register $M[i]$ is updated if the rank is higher than the current value of
102 $M[i]$.
103 When the read classification is finished, KrakenHLL aggregates the taxon sketches up the
104 taxonomy tree. Each taxon's sketch is merged with its children's sketches. The cardinality
105 estimate is computed using a recently reported improved method (Ertl, 2017) that does not

require empirically determined thresholds to account for biases and switching between linear counting and HLL estimator (Supplementary Figures 1 and 2). Figure 2 shows the performance and memory usage of KrakenHLL's cardinality estimator for up to one million k-mers. Suppl. Methods Section 1 contains a more in-depth description of the algorithm and implementation.

P	*M*	Space (kB)	Rel. Error
10	1024	1	3.25%
11	2048	2	2.23%
12	4096	4	1.63%
13	8192	8	1.15%
14	16384	16	0.81%
15	32768	32	0.57%
16	65536	64	0.41%
17	131072	128	0.29%
18	262144	256	0.20%
25			0.02%

Figure 2: Cardinality estimation on randomly sampled microbial k-mers using HyperLogLog. (Left) Standard deviations on the relative errors of the estimate with precision p ranging from 10 to 18. As expected, higher values of p give lower relative error, and no systematic bias is apparent. Up to cardinalities of about $2^p/4$ the relative error is near zero, and at higher cardinalities the error boundaries stay constant. (Right) The size of the registers, space requirement, and expected relative error for HyperLogLog cardinality estimates with different values of p. For example, with a precision p=14, the expected relative error is less than 1%. The counter only requires 16 kB of space, which is three orders of magnitude less than that of an exact counter (at a cardinality of a million). Up to cardinalities of $2^p/4$, a sparse representation of

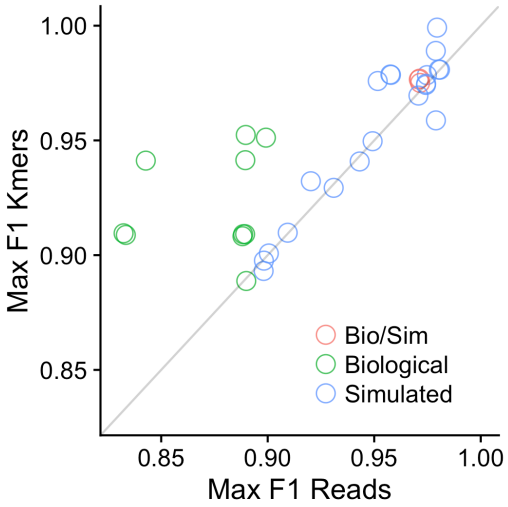
121 the counter is used with a higher precision of 25 and an effective relative error rate of about
122 0.02%.
123

124 **Results on simulated and biological data**

125 Simulated test datasets are invaluable in assessing the performance of bioinformatics algorithms.
126 Read simulators can create arbitrarily complex artificial communities and we know the source of
127 every read. However, simulated datasets do not necessarily represent biological data.
128 Specifically, laboratory and environmental contamination, a main reason behind false
129 identifications in metagenomics samples (Salter, et al., 2014), are hard to model. Biological test
130 datasets that are generated by mixing bacterial isolates at known quantities, on the other hand,
131 usually have very few species and limited complexity.
132

133 (McIntyre, et al., 2017) recently reviewed eleven metagenomics classifiers and compiled a list of
134 simulated and biological test datasets from 16 distinct sources (McIntyre-Mason, Suppl. Table
135 2). Eleven of these datasets were from biological mock communities. The largest biological
136 datasets consist of 23 species that were mixed at even proportions (Human Microbiome Project
137 mock communities, sequenced with Illumina and 454 machines). We tested KrakenHLL on ten
138 biological and 21 synthetic datasets to see if better separation of false positives and true positives
139 can be achieved using unique k-mer counts instead of read counts (see Suppl. Table 3). Our main
140 measure for comparison is the maximum F1 score, defined as 2*precision*recall/(precision +
141 recall).
142

143 Unique k-mer count thresholds worked very well in biological datasets, performing better than
144 the read count threshold in nine out of ten datasets, with a tie in one (Figure 3 and Suppl. Table
145 3). On average, the maximum F1 was 0.05 higher when using k-mer instead of read thresholds,
146 improving from 0.87 to 0.92. As expected, the difference was not as clear in simulated datasets,
147 even though the k-mer count still performed better than the read count. In eight out of the 21
148 datasets, both metrics performed equally well, as the datasets were easily separated into true and
149 false identifications. In eight datasets k-mer count achieved better F1 scores, and in five read
150 count achieved better F1 scores. The average F1 with k-mer count was slightly higher with 0.945
151 against 0.940. This difference in difference in performance is likely due to simulated datasets
152 lacking some features of biological data.



153

154 Figure 3: Using unique k-mers as thresholds instead of reads can give higher F1 scores. Each dot
155 is a dataset described in McIntyre, et al. Excludes (for visual purposes) dataset LC5 of Segata et
156 al. with a F1 read score of 0.73 and F1 Kmer score of 0.75.

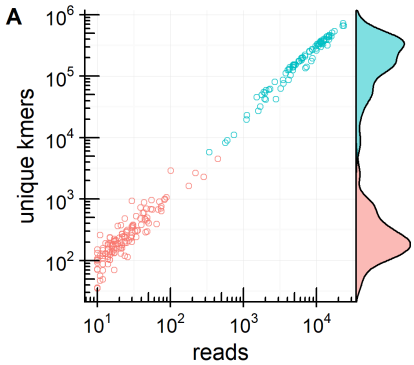
157

158 Figure 4 shows the results on two simulated and one biological datasets. In simple simulated and
159 biological datasets, the true species often separate nearly perfectly using either a read count or a

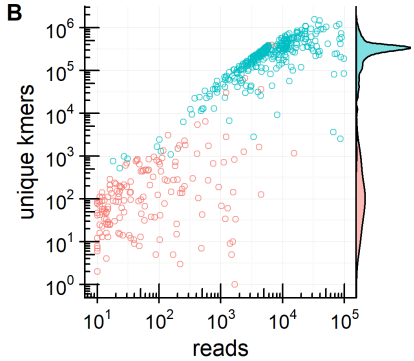
160 unique k-mer count threshold (Figure 3 (A)). In more complex datasets, however, read count

161 thresholds often contain more false species than k-mer thresholds (Figure 3 (B) and (C)).

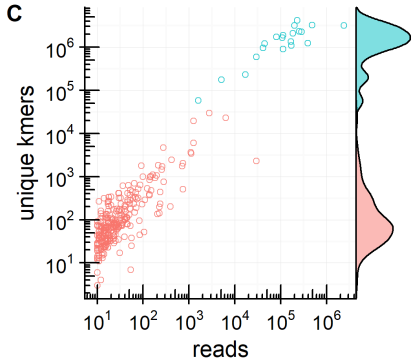
162



Data Type	Simulated
Description	high-complexity, evenly distributed mock community
No. of Genomes	100
Total No. of Reads	999991
Read Length	88.6
Publication Source	Segata et al., 2013
Max F1 Reads	0.974
Max F1 Kmers	0.979



Data Type	Simulated
Description	Unambiguously mapping reads, "simBA-525"
No. of Genomes	525
Total No. of Reads	5727654
Read Length	100
Publication Source	Ounit and Lonardi, 2016
Max F1 Reads	0.952
Max F1 Kmers	0.975



Data Type	Biological
Description	HMP Mock community - Illumina sequencing
No. of Genomes	23
Total No. of Reads	6562065
Read Length	75
Publication Source	HMP, http://hmpdacc.org/HMMC/
Max F1 Reads	0.889
Max F1 Kmers	0.952

163

164 Figure 3: Unique k-mer counts separate true identifications better from false ones. The plot

165 shows the number of reads *vs* the number of unique k-mers in two simulated and one biological

166 test datasets. Each point is a species identification. Blue dots are the 'true' species, red dots are

167 false positive or background identifications. (A) Relatively easy case of a simulated test dataset

168 with 100 genomes. The true and false positives separate nearly perfectly with either read or k-

169 mer count. (B) Largest simulated test dataset shows better separation by unique k-mer counts.
170 (C) Largest biological dataset also separates well between true and false positives.
171
172
173 **Results on biological samples for infectious disease diagnosis**
174 Metagenomics is increasingly used to find species of low abundance. A special case is the
175 emerging use of metagenomics for the diagnosis of infectious diseases (Simner, et al., 2017;
176 Zhang, et al., 2015). Host tissue or body fluids are used to find the likely culprit of a disease.
177 Usually, most (often 95% and more) of the reads match to the host, and maybe 10 to 100 out of
178 the millions of reads are matched to the target species. Skin bacteria from the patient, physician
179 or lab personal and other contamination from sample collection or preparation can easily
180 accumulate a similar number of reads, and thus cloud the detection of the pathogen.
181
182 To assess if the unique k-mer count metric can be used to rank and identify pathogen
183 identification, we reanalyzed ten patient samples (Salzberg, et al., 2016). (See Supplementary
184 Methods for details on the database, which also contains over 100 thousand viral strain
185 sequences.). (Salzberg, et al., 2016) sequenced spinal cord mass and brain biopsies from ten
186 patients in the intensive care unit, for whom routine tests for pathogens returned inconclusive. In
187 three out of the ten cases, a likely diagnosis could be made with the help of metagenomics, and
188 in a fourth case, a diagnosis could be made with an updated database. For confirmation of
189 metagenomics class, the authors re-aligned pathogen reads to individual genomes.
190

191 Table 1 shows the results of our reanalysis for the confirmed identifications in the four patients,
192 including the number of reads and unique k-mers of the pathogen, as well as the number of
193 covered bases of a re-alignment. Even though the read numbers are low in some cases, the
194 number of unique k-mers suggests that they are distributed across the genome. For example, in
195 PT8, 15 reads are matching 1570 k-mers, and re-alignment shows 2201 covered base pairs. In
196 contrast, Table 2 shows examples of identifications in the same dataset that are not well
197 supported by a high unique k-mer count.
198
199 Table 1: Pathogen identifications in patients with suspected neurological infections. The
200 pathogens were identified with as little as 15 reads, but those mapped to a high number of unique
201 k-mers, indicating random distribution of the reads on the genome. "Bases" are the number of
202 covered bases in the re-alignment of a selected genome. Interestingly, the k-mer count in PT5
203 reveals that there seems to be more than one viral strain present, as the k-mers cover more than
204 one genome.

Sample	Name	Reads	K-mers	Bases
PT5	Human polyomavirus 2	9650	7129*	5130
PT7	*Elizabethkingia genomosp. 3*	403	20724	52921
PT8	*Mycobacterium tuberculosis*	15	1570	2201
PT10	Human gammaherpesvirus 4	20	2084	2780

205
206 Table 2: Dubious identifications have few k-mers. Note that the viral identifications in PT4 and
207 PT10 stem from non-RefSeq viral genomes form the NCBI Viral Genome Resource. Since the
208 KrakenHLL reports sequence-level matches, the source genomes are easy to find.

Sample	Name	Reads	K-mers
PT3	*Clostridioides difficile*	122	126
PT4	Hepatitis C virus JF343788.1 Recombinant Hepatitis C virus	101	3
PT5	*Akkermansia muciniphila*	936	136
PT10	Human betaherpesvirus 5 JN379815.1 UNVERIFIED: Human herpesvirus 5 strain U04, partial genome	63	5

209

210 **Storing strain genomes with assembly project and sequence accessions**

211 Kraken stores a NCBI taxonomic identifier for each k-mer in its database. This strategy worked
212 well when new taxonomy IDs were assigned to each new microbial strain in GenBank. However,
213 in 2014 the NCBI Taxonomy project stopped giving new IDs to microbial strains – only novel
214 species get new taxonomy IDs (Federhen, et al., 2014). New strains, therefore, have the
215 taxonomy ID of the species, or the taxonomy ID of a strain that was added before 2014.
216 Microbes that have been intensively surveyed, such as *Escherichia coli* or *Salmonella spp.*, have
217 up to hundreds of genomes indexed with the same taxonomy ID, and are thus indistinguishable
218 by Kraken. The new way of identifying microbial strains is to use the Bioproject, Biosample and
219 Assembly accession codes (Breitwieser, et al., 2017). KrakenHLL thus adds new nodes to the
220 taxonomy tree as children of the assigned taxon. A taxonomic node may also be added for each
221 sequence – e.g. specific bacterial chromosomes or plasmids. Those new nodes in the taxonomy
222 tree are given taxonomy IDs starting at 1,000,000,000. Having these extended nodes can help
223 identify specific strains as well as bad database sequences (see Table 2 and Suppl. Table 3).

224

225 **Hierarchical read classification with multiple databases**

226 KrakenHLL allows using multiple databases hierarchically in order of confidence. In the
227 following example each k-mer is matched first against the HOST, then the PROK, then the
228 EUK_DRAFT database.
229
230 krakenhll --db HOST --db PROK --db EUK_DRAFT
231
232 Note that all database need to share the same taxonomy database. If taxIDs are added for
233 genomes or sequences, then it is necessary that the databases are consecutively constructed with
234 the same taxonomy database.
235

236 **Timing and memory requirements**

237 The additional features of KrakenHLL come without a runtime penalty. In fact, due to code
238 improvements, KrakenHLL can run faster than Kraken especially when most of the reads are
239 from one species (See Suppl. Table 2 for timings on patient data, Suppl. Table 3 for timings on
240 the test datasets). On the patient data, the processing speed (base-pairs per minute) was on
241 average 57% higher with KrakenHLL compared to Kraken, while it was 8% higher. Overall wall
242 clock time was slower, too, when comparing the runtime of both kraken and kraken-report with
243 krakenhll (which generates the report with the classification binary). The average additional
244 memory requirements were less than 1GB. On the patient datasets, the average maximum
245 memory usage went from 118 to 118.35GB, and for the test datasets, the usage went up from
246 46.28 to 46.99GB.

247 # Conclusions

248 We present a novel method that combines fast k-mer based classification with efficient
249 cardinality estimation. We demonstrated that unique k-mer counts can help discard false
250 identifications in real samples. When the reads from a species yield many unique k-mers, we are
251 more confident that the taxon is truly present, while a low number of unique k-mers suggests a
252 possible false positive identification. It is important to note that choice of the appropriate
253 threshold will depend on the application. For example, in infectious disease diagnosis, unique k-
254 mers can be used for ranking of the identifications. Conversely, in microbial ecology, a global
255 threshold on the number of unique k-mers can be applied at any desired taxonomic rank. We
256 believe that the ability to summarize to higher levels of the tree is a great advantage of the k-mer
257 count over using covered bases in a genome alignment. In summary, KrakenHLL gives more
258 confident identifications by reporting the unique k-mer count and coverage, without any runtime
259 penalty.
260

261 **Acknowledgements**

262 We'd like to thank Jen Lu, as well as David Karig, Susan Bewick, Peter Thielen, Thomas
263 Mehoke for valuable discussions. Furthermore, we'd like to thank Jessica E. Atwell for
264 proofreading of the manuscript.
265

266 **Funding**

267 This work was supported in by the National Institutes of Health [grant number R01-HG007196];
268 and the U. S. Army Research Office [grant number W911NF-14-1-0490].
269

References

Breitwieser, F.P., Lu, J. and Salzberg, S.L. A review of methods and databases for metagenomic classification and assembly. *Brief Bioinform* 2017.

Brister, J.R.*, et al.* NCBI Viral Genomes Resource. *Nucleic Acids Research* 2015;43(D1):D571-D577.

Brown, J.R., Bharucha, T. and Breuer, J. Encephalitis diagnosis using metagenomics: application of next generation sequencing for undiagnosed cases. *Journal of Infection* 2018.

Dadi, T.H.*, et al.* SLIMM: species level identification of microorganisms from metagenomes. *PeerJ* 2017;5:e3138.

Ertl, O. New Cardinality Estimation Methods for HyperLogLog Sketches. *arXiv:1706.07290* 2017.

Federhen, S.*, et al.* Toward richer metadata for microbial sequences: replacing strain-level NCBI taxonomy taxids with BioProject, BioSample and Assembly records. *Stand Genomic Sci* 2014;9(3):1275-1277.

Flajolet, P.*, et al.* HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In, *AofA: Analysis of Algorithms*. Juan les Pins, France: Discrete Mathematics and Theoretical Computer Science; 2007. p. 137-156.

Heule, S., Nunkesser, M. and Hall, A. HyperLogLog in practice. 2013:683.

Irber Junior, L.C. and Brown, C.T. Efficient cardinality estimation for k-mers in large DNA sequencing data sets. *bioRxiv* 2016.

McIntyre, A.B.R.*, et al.* Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome biology* 2017;18(1).

Mukherjee, S.*, et al.* Large-scale contamination of microbial isolate genomes by Illumina PhiX control. *Stand Genomic Sci* 2015;10:18.

Quince, C.*, et al.* Shotgun metagenomics, from sampling to analysis. *Nat Biotechnol* 2017;35(9):833-844.

Salter, S.J.*, et al.* Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biol.* 2014;12:87.

Salzberg, S.L.*, et al.* Next-generation sequencing in neuropathologic diagnosis of infections of the nervous system. *Neurology(R) neuroimmunology & neuroinflammation* 2016;3(4):e251.

Simner, P.J., Miller, S. and Carroll, K.C. Understanding the Promises and Hurdles of Metagenomic Next-Generation Sequencing as a Diagnostic Tool for Infectious Diseases. *Clinical Infectious Diseases* 2017.

Thoendel, M.*, et al.* Impact of Contaminating DNA in Whole-Genome Amplification Kits Used for Metagenomic Shotgun Sequencing for Infection Diagnosis. *J Clin Microbiol* 2017;55(6):1789-1801.

Wood, D.E. and Salzberg, S.L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome biology* 2014;15(3):R46.

Zhang, C.*, et al.* Identification of low abundance microbiome in clinical samples using whole genome sequencing. *Genome biology* 2015;16(1).