# Practical limits in resolving species and genomes from metagenomic data

*This manuscript (permalink) was automatically generated from dib-lab/2022-paper-genomic-tax-redundancy@80053ad on December 9, 2022.*

## Authors


- **John Doe**
  XXXX-XXXX-XXXX-XXXX · johndoe · johndoe
  Department of Something, University of Whatever · Funded by Grant XXXXXXXX

- **Jane Roe** ✉
  XXXX-XXXX-XXXX-XXXX · janeroe
  Department of Something, University of Whatever; Department of Whatever, University of Something

✉ — Correspondence possible via GitHub Issues or email to Jane Roe <jane.roe@whatever.edu>.

## Abstract

A central challenge in bacterial genomics and taxonomy is that genomic databases are increasingly large and contain redundant content. This impacts the accuracy of taxonomic profilers and metagenome analysis tools that rely on these databases. Here, we probe the practical and theoretical limits of genomic identification and taxonomic classification by exploring *unicity distance* and *Shannon entropy*. Unicity distance provides an estimate of how many k-mers are required to precisely identify a single reference genome from within a database, while Shannon entropy of k-mers describes the informativeness of a k-mer for taxonomic classification. We show that approximately 30% of genomes in GTDB rs207 have infinite unicity and that 99% of k-mers can resolve taxonomy at the species, genus, or family level. We conclude that unicity distance and Shannon entropy provide simple metrics for evaluating genomic redundancy and taxonomic coherence of large genomic reference databases.

# Introduction

Introduction goes here.

Nasko et al. (2018) showed that increasing database size substantially degraded classification accuracy with Kraken, a k-mer based approach that offers high recall [1]. Recently, Portik et al. showed that a different k-mer based approach, sourmash, achieved both high recall [2] and high precision. Our proximal motivation in this work is to understand why.

Our larger motivation is to explore the information content that can be used by k-mer-based techniques such as Kraken and sourmash for genomic and taxonomic classification of metagenomes. This is particularly relevant as reference collections grow larger and include many genomes that belong to the same species. Strain-resolved metagenomic classification is likely to grow in importance as well.

We seek a generative solution that will guide analysis approaches in the future. In particular, we are interested in defining metrics that can help characterize approaches at a theoretical level so that we can understand the limitations of current approaches and improve future approaches to reach maximal use of available information.

Strain-level metagenomics tools: https://hackmd.io/54VvAP2FR4GCiCeJA25lVQ?both

Large-scale k-mer-based analysis of the informational properties of genomes, comparative genomics and taxonomy https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0258693

Fast and flexible bacterial genomic epidemiology with PopPUNK
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6360808/

KMCP intro makes lots of good points. Note combinatoric statement in SBT paper. Make point that FracMinHash is picking independent k-mers.
https://www.biorxiv.org/content/10.1101/2022.03.07.482835v2.full

CLARK https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-015-1419-2

BLEND: https://www.biorxiv.org/content/10.1101/2022.11.23.517691v1

More points to make, somewhere:

- you can use combinatorics with more than just FracMinHash. All k-mers, or seeds (like BLEND), or ...

# Results

Hackmd for tables here: https://hackmd.io/GvngZ4gHQE-9ERB4Gd71HQ

Note assumptions:

- reference genomes are correct
- we are not trying to generalize - this is just about looking things up

Overall messages to pick from, in order of CTB pref:

- It is hard to do genome-resolved analysis from metagenomes (but easy to do highly specific taxonomic resolution with just k-mers). AND Genome resolved analysis is probably important for strain resolved analysis.
- FracMinHash-based measures (hash specificity, hash unicity distance) provide valuable information that highlight genomes and taxonomic units that are hard to classify with any approach; these genomes and taxonomic units are where to focus on improvements, because the rest are easy.
- short reads have limited capability / correspond well to hashes; long reads do not seem to add significant capability. (THIS ARGUMENT NEEDS TO BE SHARPENED)
- sourmash works well because of combinatorics automatically picking discriminatory hashes.
- NCBI is more confused than GTDB.

Challenges:

- FracMinHash has false negatives, so we have to be careful about conclusions about what k-mers *cannot* do.
- Do we want to make this (mostly) about FracMinHash, or not? Could make title “Practical limits in resolving species and genomes from metagenomic data with FracMinHash” :) and/or parameter tuning for FracMinHash.

## Many individual hashes are genome specific, but not all genomes have perfectly informative hashes

We first ask, if we see a hash from the reference database in a metagenome, what is the likelihood of that hash being specific to a genome? And how many genomes have hashes that uniquely identify that genome?

Consider a collection of the size of GTDB - 320,000 genomes with an average size of 5e6 bp. In the simplest case where these genomes were completely random and had neither bias nor redundancy, most 31-mers present in one genome should be distinct to that genome. Thus the presence of any one k-mer in a sequencing data set should be capable of perfectly distinguishing that genome’s presence even in low-coverage data sets (Appendix A).

In practice, genomes are neither unbiased nor contain distinct content. In order to analyze 300k genomes at scale, we used FracMinHash. At reasonable parameters (S=10,000, k=31) we generated a representative collection of hashes and found that 15378449 of 22792206 hashvals (67.5%) in GTDB rs207 are perfectly informative at genome level.

These 15.4m perfectly discriminating hashes are not equally distributed across genomes, however. Of the 318k genomes in GTDB rs207, 149k (47.1%) of the genomes contain a hash that perfectly identifies

that genome; the remaining 52.9% of genomes cannot be unambiguously identified due to the presence of any single hash.

Inspection of these genomes using all k-mers confirms that, in general, a lack of discriminating hashes for FracMinHash=10,000 means that there are fewer than 10,000 distinguishing k-mers present in the genome (e.g. see results for `s__Bosea`). That is, the FracMinHash results provide an approximate upper bound on the number of distinguishing k-mers

Conclusion of this results section (to go into discussion): it is not easy to identify individual genomes based on k-mers alone, and doing so would/will require making distinctions between genomes based on very small fractions of k-mers. This has significant implications for sensitivity and specificity of k-mer based techniques.

## Most species do have perfectly informative hashes.

We next ask, if we see a k-mer from the reference database in a metagenome, what is the likelihood that we can uniquely pinpoint a specific taxonomic unit's presence?

Using GTDB, we calculate that 6.2% of k-mers are not species specific, and 0.9% of k-mers are neither species, family, or genus specific (Table 1). That is, 99.1% of hashes uniquely identify a specific family within the GTDB taxonomy.

**Table 1:** Entropy measurements for GTDB taxonomy using 318k genomes from rs207 genomes.

| Taxonomic level (GTDB) | # perfectly informative hashes | cumulative total % |
|---|---|---|
| species | 21,150,287 | 92.8% |
| genus | 1,262,281 | 98.3% |
| family | 170,249 | 99.1% |
| order and above | 209,389 | 0.9% |

There are 73 genera (of 16686) with no perfectly identifying hashes, and 8 families (of 4107), and 4 orders (out of 1593). However many of these (all of these? :) are pathological cases where there are very few genomes at the given taxonomic rank. CTB: fix these numbers to reflect some of the weird edge cases we found :)

Conclusion of this results section for discussion: it is straightforward to do taxonomic identification of sequencing data within the GTDB taxonomy.

## Taxonomies vary in their k-mer specificity

We can also calculate these numbers for the same genomes using the NCBI taxonomy instead of the GTDB taxonomy. Table 2 uses the NCBI taxonomy with the same genomes used above. Here we see that approximately 4.5% of hashes cannot be used to distinguish between different *families* - a full 5 times as many as with the GTDB taxonomy. These 1.0 million hashes represent approximately 10 billion k-mers, or approximately 2,000 bacterial genomes worth of sequence.

**Table 2:** Entropy measurements for same 318k GTDB rs207 genomes as in Table 1, but using NCBI taxonomic labels.

| Taxonomic level (NCBI) | # perfectly informative hashes | cumulative total % |
|---|---|---|
| species | 20,744,791 | 91.0% |

| Taxonomic level (NCBI) | # perfectly informative hashes | cumulative total % |
|---|---|---|
| genus | 779,234 | 94.4% |
| family | 245,718 | 95.5% |
| order and above | 1,022,463 | 4.5% |

CTB: check NCBI results.

CTB: provide tax results for NCBI.

Conclusion of this results section for discussion: this is a way to evaluate taxonomies. Do we have a good argument for why k=31 should be specific to species/genus? Or is this empirical?

- Can we link to ANI / hash relatedness work?
- can we calculate k-mer size vs ANI?

## Many genomes cannot be uniquely distinguished based on combinations of hashes

From the results above, of the 318k genomes in GTDB rs207, only 149k (47.1%) of the genomes can be identified by considering individual hashes - each of the remaining 52.9% genomes have no hashes that are unique to that genome.

K-mers can be further ranked based on *how much* discriminatory power they offer - for example, a k-mer that is only present in two genomes in the collection is much more informative than a k-mer present in 20. One way to formalize this is with Shannon entropy (formula, base 2).

**Table 3:** Distribution of 31-mers among distinct genomes in GTDB rs207, estimated with FracMinHash (scaled=10,000).

| # genomes containing | H (Shannon entropy) | # hashes | % total |
|---|---|---|---|
| 1 | 0 | 15,378,449 | 67.5% |
| 2 | 1 | 3,407,595 | 15.0% |
| 3 | 1.58 | 1,337,246 | 5.9% |
| 4 | 2 | 694,687 | 3.0% |
| 5 or more | >= 2.32 | 1,974,229 | 8.7% |

When we measure the Shannon entropy of hashes, we see that more than 80% of hashes are in only two genomes, and 90% of hashes are contained in four or fewer genomes. This suggests that combinations of hashes could be used to identify specific genomes.

So we next ask how many genomes can be distinguished from each other using a combinatorial collection of hashes? To do this, we estimate the *unicity distance* of each genome in the database, where the unicity distance is defined as the smallest set of hashes capable of uniquely identifying an individual genome. (k=31, scaled=10,000)

Table 5 shows that approximately 29.2% of the genomes in GTDB rs207 cannot be distinguished uniquely by *any* combination of 31-mer hashes at a scaled of 10,000.

Comparing with informative k-mer results above, we see that 47.1% of genomes can be classified with a single hash, and 29.2% cannot be classified with *any* combination of hashes. So combinatorial

approaches like sourmash gather can resolve an additional 23.7% of genomes beyond single hashes, but not more.

**Table 4:** Estimated unicity distances with hashes for 318k GTDB rs207 genomes using FracMinHash as implemented in sourmash (k=31, scaled=10,000).

| Unicity distance | Number of genomes | Percent of genomes |
|---|---|---|
| 1 | 48,630 | 15.3% |
| infinite | 92,564 | 29.2% |

(CTB: calculate for unicity 2, and 3.)

Taxonomic summarization confirms that most infinite unicity genomes are at species level or below. (CTB: link/connect to previous results above.)

(Do we want to calculate scaled=1 k-mer unicity in this section?)

Conclusions for discussion:

- sourmash/FracMinHash can perfectly distinguish most species based on combinations of hashes
- the value of gather here is that it automatically uses discriminatory hashes without any taxonomy-aware preprocessing. should be even more valuable at low scaled. Can apply to any containment based approach!
- this will not necessarily work for pinpointing specific genomes / strain resolution!

## Implications for short-read mapping

Since FracMinHash results typically correspond nicely to short-read mapping, the above results suggest that short-reads may struggle to distinguish between many closely related genomes.

TODO:

- distinguish between “just” using end-to-end alignment vs calculating detailed SNP/SVs (which is coverage dependent, will decrease sensitivity)
- show for select pairs of genomes that short-read mapping approaches struggle, and/or that they match containment.

## Implications for long-read mapping

- should be very effective
- can use fracminhash etc to identify likely genomes once over 5kb

Challenges in long-read mapping approaches:

- may be too specific / lose sensitivity to genome rearrangements?

## Other TODO:

- flesh out theoretical/simulation results
- can we / should we link any of the above to ANI?
- do we want to look at all Genbank? ick.

## LEFTOVER TEXT BELOW

(What is value of H here, exactly? Other than H=0? Link to taxonomy, and combinatorics; also see unicity distance at bottom.)

We measured the species distribution in GTDB rs207 for 21.2 million hashes, representing 212 billion 31-mers, and calculated the Shannon entropy for each hash (equationXX) at the species, genus, and family levels.

## Shannon entropy can summarize the taxonomic cohesion of taxonomies based on genomic relationships.

We can also calculate the Shannon entropy with respect to different taxonomies.

## Unicity distance can be used to estimate genomic redundancy

Of the 318k genomes in GTDB rs207, only 149k (47.1%) of the genomes can be identified by considering individual k-mers - each of the remaining 52.9% genomes have no k-mers that are unique to that genome. Suppose we use *combinations* of k-mers to identify genome presence/absence?

We next ask, how many genomes can be distinguished from each other using a combinatorial collection of k-mers? To do this, we estimate the *unicity distance* of each genome in the database, where the unicity distance is defined as the smallest set of hashes capable of uniquely identifying an individual genome. (k=31, scaled=10,00)

Table 5 shows that approximately 29.2% of the genomes in GTDB rs207 cannot be distinguished uniquely by *any* combination of 31-mers at a scaled of 10,000.

Comparing with informative k-mer results above, we see that 47.1% of genomes can be classified with a single hash, and 29.2% cannot be classified with *any* combination of hashes. So combinatorial approaches like sourmash gather can resolve an additional 23.7% of genomes beyond single hashes, but not more.

(Compare also with k-mer informativeness; can we tie entropy computation at top back to number of genomes with unicity of 1, and cross validate?)

**Table 5:** Estimated unicity distances with hashes for 318k GTDB rs207 genomes using FracMinHash as implemented in sourmash (k=31, scaled=10,000).

| Unicity distance | Number of genomes | Percent of genomes |
|---|---|---|
| 1 | 48,630 | 15.3% |
| infinite | 92,564 | 29.2% |

(FIX table above ;))

## The large majority of taxa can be distinguished by combinations of k-mers

Suppose we sequence a genome completely as part of a metagenome. Are there any taxonomic units whose presence cannot be precisely determined using a combination of k-mers?

Out of 65,703 species in GTDB rs207, 64,620 (95.7%) can be detected by a combination of k-mers.

But:

- at rank species, there are 1083 taxa with infinite unicity, out of total of 65703
- at rank genus, there are 73 taxa with infinite unicity, out of total of 16686
- at rank family, there are 8 taxa with infinite unicity, out of total of 4107
- at rank order, there are 4 taxa with infinite unicity, out of total of 1593

In this case, we see that there are a small number of taxa whose presence cannot be determined based solely on the basis of combinations of k-mers. This is usually because there are very few members of that taxon in the database; for example, of the ~1000 species with infinite unicity, 977 have only two members.

This means that no matter the data, presence of these 1083 species cannot be uniquely determined with (e.g.) sourmash.

(Analyze further. Fix k-mer results above.)

## Implications for read mapping and (pseudo)alignment

Infinite unicity at a scaled=10,000 implies that genomes cannot be distinguished via pseudoalignment of reads under 10kb in length

Arguments to make and evaluate:

- genomes that are identical with gather cannot be distinguished based on end-to-end alignment of reads - you have to look at the actual alignment. This means pseudoalignment cannot distinguish them. Maybe call these “infinite unicity groups”?
- genomes that have infinite unicity across a collection of other genomes at a given scaled value cannot easily be distinguished by end-to-end read mapping.
- this implies that SNPs or small SVs will need to be used to distinguish between these genomes.

Regardless, all genomes and taxa *but* these are easily distinguishable with combinations of k-mer.

---

# Discussion

Automatic selection of discriminating k-mers through combinatorics is one reason why sourmash performs so well.

Foreword to Discussion or Conclusion, or maybe end of Introduction -

Microbial bioinformatics is increasingly challenged by an abundance of closely related genomes, which has led in turn to pangenomics. The features of archaeal and bacterial genomes include considerations such as open and closed pangenomes, along with a seemingly endless diversity of accessory elements and a highly conserved set of core genes. The subject of this study is how the *genomic* structure of pangenomes, as represented in reference databases, affects reference-based interpretation of metagenomic data.

The key perspectives we bring to this work are as follows: we are facing a substantial over-abundance of genome sequences, and there is an increasing interest in using them to interpret metagenomic data at the strain or individual genomic level. This, in turn, is made considerably more difficult by the empirical observation that the vast majority of shared nucleotide sequence is present at and below the species level. Here we focus on evaluating and measuring the extent to which species and genomes can be identified at the nucleotide level, using both single k-mers and k-mers in combination.

We also show that human-constructed taxonomies both simplify and complicate matters: while they provide helpful resolution up to a point, they also complicate tool development and design. Taxonomic benchmarks on their own may not be a useful way forward. (Unload sourmash perspective. Discuss contamination here too.)

We model things as a database lookup problem where we assume that the right match is in the database, and the challenge is distinguishing them in the face of the actual structure in the database. For this study we explicitly ignore questions of sensitivity across sequence divergence as well as errors in the database sequences, e.g. incorrect sequence or contamination.

## Species-level classification of sequencing data is straightforward with k-mers

- Taxonomic classification of sequencing data is statistically straightforward with long k-mers. You can easily achieve high sensitivity and high precision/recall.
- Most species can be identified by the presence of a single k-mer from any genome within that species.
- Most k-mers in the reference database identify a species, a genus, or a family.

## Genome-based classification of sequencing data with k-mers is more challenging.

- We use genomes instead of strains because if you can identify genomes presumably you can identify strains, and strains are conceptually challenging. This is basically the hardest version of the problem!
- A full 47.3% of genomes in the database can be readily identified at a low level of sequencing by using individual hashes. Yay!
- Another 23.7% of genomes can be identified by using hashes in combination with each other.

- However, the remaining 29.2% of genomes are not that different from other genomes in the database - we can put a hard constraint of < 10k unique k-mers on each of them, or less than 2% of the k-mers in the genome! This small a signal makes identification challenging in the face of genome divergence/dropout, etc.

## Mapping reads to genomes suffers from similar challenges

- We use k-mers to identify challenging genomes for mapping. This works for two reasons: first, k-mers are more stringent than mapping, so provide an upper bound; and second, if you can use k-mers, you don't need to use mapping.
- We evaluate end-to-end alignment of reads to reference genomes and show that, without consideration of SNPs or SVs, many genomes are hard to distinguish with read mapping.
- We do not consider SNPs and SVs here because reliably calling them from sequencing data requires higher coverage data and is relatively challenging (and subject to many false positives).
- Long reads do not necessarily solve problems for the 23.7% of genomes that cannot easily be distinguished by k-mers.

## Different taxonmies confound things.

- The Nask et al. paper from 2018 is largely a commentary on two things: too short a k-mer size for individual k-mers, and how "bad" taxonomies can mess things up.
- Measuring k-mer specificity as above is a simple way to evaluate taxonomies.
- GTDB is a nicer, simpler taxonomy.
- NCBI taxonomy causes more problems.
- New organizational schemes such as LINS may be better, but other things may come into play - contamination and lateral gene transfer. Out of scope of this work.
- Organizing things taxonomically helps when you just need species-level classification but hurts when you care about genome-level classification.

## Reduced-representation databases are not good.

- Reduced-representation/non-redundant databases such as used in (many programs) are not good when you want strain level analysis. You end up omitting sequence that may be important functionally.
- We should use all available data and sort through it afterwards.
- Several approaches (sourmash, ganon, shei wen) for doing this.

## Sourmash works well up to a point.

- We were motivated to explore the above because of sourmash high recall and precision in long read paper.
- Sourmash works well because it automatically finds discriminatory k-mers.
- But sourmash will fail to classify the 23.7% of genomes above with default parameters.
- These genomes are not readily distinguishable by read mapping, either.
- This is an interesting challenge for future classifier benchmarks.

## LEFTOVER TEXT

## Species-level classification should be straightforward with k-mers

Taxonomic classification to the species level is straightforward, largely because GTDB taxonomy is closely tied to genomic content and most of the genomic redundancy lies within species and genus level. Thus GTDB taxonomy largely encapsulates this redundancy. The entropy measurements

demonstrate that it is possible to choose an informative subset of k-mers that would robustly classify at the species level, and that doing so would not compromise sensitivity. LCA-style approaches such as those used by Kraken should work even if we use genus and family level k-mers, while eliminating those above.

Shared genomic content at higher levels confounds taxonomic classification methods. While surely some shared genomic content is real, our analysis suggests that significant portions of it are contamination.

## Classification below the strain level

Detecting genomes from sequences is easy with k-mers, but significant redundancy prevents straightforward classification to the genome/strain level. Here leveraging combinatorial application of k-mers provides significant leverage; this is how sourmash achieves high precision. Nonetheless sourmash cannot distinguish a full 30% of the genomes in the database from each other.

Some implications are that it should be possible to use information from both short and long reads to classify robustly to the species level, but it is unlikely to work below that (at the strain level). This is because many reads will map to shared content within a species, and some of that shared content may not distinguish a particular genome from others in the pangenome at the resolution of the available reads.

A simple thought experiment also suggests that reduced-representation /slimmed-down databases will not support strain-level classification. Suppose that a technique exists that can classify reads to a strain level. First, choose a read that belongs to two or more different strains; there is no way to identify which strain this belongs to. Second, choose a read that belongs to a strain that is not represented in the database; while it clearly belongs to a known species, there is no way to identify which. Classifiers should be using all available information and it is clearly possible to do so, viz sourmash.

(Probably need to spend some time here talking about core vs accessory genomes.)

Approaches such as sourmash can try to operate “above” individual k-mers, but will also be stymied by infinite unicity. Here combinatorial uses of k-mers (via e.g. containment) may be able to resolve strains, but will need to do so at higher resolution than sourmash’s current parameters. Here approaches such as Agamemnon may be useful.

## A method to evaluate, compare, and study taxonomic lables

The GTDB and NCBI taxonomies are not entirely consonant and our studies using entropy suggest that a substantial portion of the NCBI taxonomy is confused. Comparing Table 1 and Table 2, we see that 5x as many k-mers belong to genomes that do not share the same family-level labels. This difference is due solely to the taxonomic labels. It is not necessarily surprising that NCBI is so different since GTDB is directly constructed using content-based phylogeny, but it does suggest that there are many places where the NCBI taxonomy should be examined closely.

(drill down; contamination, etc.)

K-mers are not the problem; taxonomy is.

### Combinatorial approaches may minimize the impact of contamination and taxonomy

Maybe sourmash gather does a good job of picking out genome based on contextual clues?

Sourmash gather doesn't need to pick good k-mers a priori - it uses all of them in combination. But the key one(s) are the unique ones.

### Shannon entropy and unicity on k-mers are robust ways to study, evaluate, and summarize large databases

Exploration of these results suggest that k-mer size and scaled do not dramatically affect our conclusions. (Confirm me, please :).

## Conclusion

This is easy mode: this kind of taxonomic classification is "just" a database lookup. What messes up taxonomic classification with k-mers is (1) biology (redundancy and laterally transferred genetic elements) and (2) humans (taxonomy). (1) is resolvable to a significant extent with combinatorics. (2) can be tackled with better metrics and systematic improvement. Here we provide measures that assist with both.

Despite this, biological questions remain that are out of scope of this paper: correctness and completeness of reference databases matters. And we really also say nothing about generalizability, where we know that we have problems.

# Methods

Methods go here.

# References


1. **RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification**
Daniel J Nasko, Sergey Koren, Adam M Phillippy, Todd J Treangen
*Genome Biology* (2018-10-30) https://doi.org/ggc9db
DOI: 10.1186/s13059-018-1554-6 · PMID: 30373669 · PMCID: PMC6206640

2. **Evaluation of taxonomic classification and profiling methods for long-read shotgun metagenomic sequencing datasets**
Daniel M Portik, CTitus Brown, NTessa Pierce-Ward
*Cold Spring Harbor Laboratory* (2022-02-02) https://doi.org/hhqs
DOI: 10.1101/2022.01.31.478527

# Appendix

## Appendix A - Individual k-mers are very sensitive to even low coverage genomes

## Appendix B - Combinatorial k-mers can resolve genomes even at low k-mer sizes.

## Appendix C - FracMinHash sketching can be modeled as a Poisson process

We can model the FracMinHash sketch algorithm as follows: for a stream of randomly generated k-mers (with k sufficiently large that the k-mers are effectively sampled without replacement (e.g. k=21)), FracMinHash with scaled=S adds hashes to the sketch at an average rate of one in every S k-mers.

The process of adding hashes to the sketch is a Poisson process, since for any given number of k-mers a discrete non-negative number of k-mers will be retained in the sketch, the events occur independently, and at most one k-mer is hashed at each time point.

Thus the distribution of output sketch sizes for a given number of input k-mers M is a Poisson distribution with lambda=M/S.

For genome comparison purposes, we want to know the average runlength of k-mers for which no hashes will be retained; this will inform the choice of the scaled parameter S for sensitively detecting overlaps of a size M k-mers. The average runlength can be easily be evaluated for random sequences (or, in genomic parlance, high complexity / non-repetitive genomic sequence) by calculating the probability of generating a sketch of size 0 given M k-mers.

Using Poisson statistics, this is:

Poisson(k=0, lambda=M / S) = exp(- M / S)

Then the probability of *at least* one hash being generated for a collection of k-mers M in size is:

1 - exp(-M / S)

and we see that this matches empirical observations using sourmash. (CTB insert notebook results.)

Here we note that the distribution of lengths of k-mer runs with zero hashes is dependent only on S, and independent of any genome size. This is what we would expect from a memoryless process such as FracMinHash sketching with a good hash fnuction.

## Appendix D - Limits on genome differences for genomes with identical hash collections

Now consider two genomes A and B, identical in size, with identical FracMinHash sketches. Suppose we want to choose the minimum scaled value S necessary to guarantee a maximum difference between the k-mers in A and B? Or, to rephrase: what S is sufficient to assert that A and B share *more than* a specific fraction f (say, 99.9%) of their k-mers?

This is equivalent to asking the question: for genomes with N distinct k-mers what is the scaled value S such that for 99.9% of those genomes, all runs of k-mers longer than 1% of N that is not sketched? (CTB: This needs to be connected to “runs” of k-mers from the genome somehow.)

For this, we take the equation

$p = \exp(- fN / S)$

and solve for S, which yields:

$S = - fN / \log(p)$

Choosing N=5e6 (a typical genome size for bacteria) and setting p = .001 and f = 0.01, we find S=7238.