# LINflow: A computational pipeline that combines an alignment-free with an alignment-based method to accelerate generation of similarity matrices for prokaryotic genomes

Long Tian [Equal first author, 1], Reza Mazloom [Equal first author, 2], Lenwood S Heath [2], Boris A Vinatzer [Corresp. 1]

[1] School of Plant and Environmental Sciences, Virginia Tech, Blacksburg, VA, United States

[2] Department of Computer Science, Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, United States

Corresponding Author: Boris A Vinatzer
Email address: vinatzer@vt.edu

**Background** Computing genomic similarity between strains is a prerequisite for genome-based prokaryotic classification and identification. Genomic similarity was first computed as Average Nucleotide Identity (ANI) values based on the alignment of genomic fragments. Since this is computationally expensive, faster and computationally cheaper alignment-free methods have been developed to estimate ANI. However, these methods do not reach the level of accuracy of alignment-based methods.

**Methods** Here we introduce LINflow, a computational pipeline that infers pairwise genomic similarity in a set of genomes. LINflow takes advantage of the speed of the alignment-free sourmash tool to identify the genome in a dataset that is most similar to a query genome and the precision of the alignment-based pyani software to precisely compute ANI between the query genome and the most similar genome identified by sourmash. This is repeated for each new genome that is added to a dataset. The sequentially computed ANI values are stored as Life Identification Numbers (LINs), which are then used to infer all other pairwise ANI values in the set. We tested LINflow on four sets, 484 genomes in total, and compared the needed time and the generated similarity matrices with other tools.

**Results** LINflow is up to 150 times faster than pyani and pairwise ANI values generated by LINflow are highly correlated with those computed by pyani. However, because LINflow infers most pairwise ANI values instead of computing them directly, ANI values occasionally depart from the ANI values computed by pyani. In conclusion, LINflow is a fast and memory-efficient pipeline to infer similarity among a large set of prokaryotic genomes. Its ability to quickly add new genome sequences to an already computed similarity matrix makes LINflow particularly useful for projects when new genome sequences need to be regularly added to an existing dataset.

1   **LINflow: A Computational Pipeline that Combines an Alignment-free with an**

2   **Alignment-based Method to Accelerate Generation of Similarity Matrices for**

3   **Prokaryotic Genomes**

5   Long Tian[1*], Reza Mazloom[2*], Lenwood S. Heath[2], Boris A. Vinatzer[1]

6   [1]School of Plant and Environmental Sciences, Virginia Tech, Blacksburg, VA, USA

7   [2]Department of Computer Science, Virginia Tech, Blacksburg, VA, USA

8   Corresponding author:

9   Boris A. Vinatzer, Latham Hall, room 551, Blacksburg, VA 24061, email: vinatzer@vt.edu

10  * equal contribution

12    **ABSTRACT**

13    **Background**

14    Computing genomic similarity between strains is a prerequisite for genome-based prokaryotic

15    classification and identification. Genomic similarity was first computed as Average Nucleotide Identity

16    (ANI) values based on the alignment of genomic fragments. Since this is computationally expensive,

17    faster and computationally cheaper alignment-free methods have been developed to estimate ANI.

18    However, these methods do not reach the level of accuracy of alignment-based methods.

19    **Methods**

20    Here we introduce LINflow, a computational pipeline that infers pairwise genomic similarity in a set of

21    genomes. LINflow takes advantage of the speed of the alignment-free sourmash tool to identify the

22    genome in a dataset that is most similar to a query genome and the precision of the alignment-based

23    pyani software to precisely compute ANI between the query genome and the most similar genome

24    identified by sourmash. This is repeated for each new genome that is added to a dataset. The

25    sequentially computed ANI values are stored as Life Identification Numbers (LINs), which are then used

26    to infer all other pairwise ANI values in the set. We tested LINflow on four sets, 484 genomes in total,

27    and compared the needed time and the generated similarity matrices with other tools.

28    **Results**

29    LINflow is up to 150 times faster than pyani and pairwise ANI values generated by LINflow are highly

30    correlated with those computed by pyani. However, because LINflow infers most pairwise ANI values

31    instead of computing them directly, ANI values occasionally depart from the ANI values computed by

32    pyani. In conclusion, LINflow is a fast and memory-efficient pipeline to infer similarity among a large set

33    of prokaryotic genomes. Its ability to quickly add new genome sequences to an already computed

34    similarity matrix makes LINflow particularly useful for projects when new genome sequences need to be

35    regularly added to an existing dataset.

37    LINflow is available at https://code.vt.edu/linbaseproject/LINflow/

## INTRODUCTION

The number of prokaryotic genome assemblies available at the National Center for Biotechnology Institute (NCBI) is growing rapidly and has reached 615,000 in 2020. It can be anticipated that many more genome assemblies will be published in the near future because of continued improvements in next-generation DNA sequencing technologies concerning throughput and sequence quality and a concomitant drop in sequencing cost. The ever-growing collection of prokaryotic genomes provides the opportunity to explore evolutionary relationships among species, genomic boundaries of species, and the genetic diversity within species.

DNA-DNA hybridization (DDH) was the first method that incorporated genome content in prokaryotic classification and became the gold standard in the 1970s (Brenner 1973). Two strains that have a reciprocal DDH value of over 70% are considered to belong to the same species (Brenner 1973). However, the low resolution, laborious experimental procedures, and limited portability of results present serious limitations (Stackebrandt & Goebel 1994). After the advent of next-generation sequencing, DDH has largely been replaced by Average Nucleotide Identity (ANI). ANI is a measure of similarity between two genomes based on the comparison of whole genome sequences (Konstantinidis & Tiedje 2005a). In its original implementation, one genome is used as a query genome and is cut into consecutive 1020 nt-long fragments. Each fragment is then aligned to the second genome, the subject genome, using BLAST (Konstantinidis & Tiedje 2005a). Alignments that result in over 30% coverage and 70% identity are retained and ANI is computed as the average identity of these alignments. An ANI between 95% and 96% has been found to correspond to 70% DDH (Goris et al. 2007). ANI even provides the resolution necessary to separate strains into different genome similarity groups within species (Rodriguez-R et al. 2018; Vinatzer et al. 2016).

While computing ANI between a single pair of genome sequences using the alignment-based BLAST algorithm is reasonably efficient, computing all pairwise ANI values for thousands of genomes

63     (thus requiring up to millions of pairwise comparisons) is slow. Therefore, various methods have been

64     developed to infer ANI based on alignment-free algorithms. In 2015, Ondov and colleagues published

65     the first implementation of MinHash (Broder 1997) for prokaryotic genome comparisons, Mash (Ondov

66     et al. 2016). Mash and similar tools, such as sourmash (Brown & Irber 2016; Pierce et al. 2019), produce

67     a reduced representation of a genome as a sketch (also referred to as a signature). This is done by

68     determining the presence of all k-mers in a genome sequence and using a hash function to translate

69     these k-mers into hashes, of which a subset is used as the sketch. Mash and sourmash then compare

70     genomes by calculating the Jaccard similarity between their sketches (Ondov et al. 2016). This results in

71     an estimate of the Jaccard similarity between the entire k-mer sets of the two genomes. Not only was it

72     possible with this approach to process and calculate the pairwise similarity of 54,118 microbial genomes

73     from NCBI RefSeq release 70 in 33 CPU hours, but this approximate Jaccard similarity also correlates

74     with ANI almost linearly for ANI values from approximately 90% to 99%. Therefore, Mash and sourmash

75     can be used to precisely and quickly cluster genomes into species.

76        The k value, *i.e.*, the k-mer length employed when a sketch is made, significantly impacts the

77     computed Jaccard similarity. A smaller k value enables the detection of similarity between genomes of

78     distantly related strains but loses resolution when the ANI between genomes is high. On the other hand,

79     longer k-mers provide high resolution when ANI is high but they cannot detect any similarity between

80     genomes of more distantly related strains (Brown & Irber 2016; Pierce et al. 2019).

81        FastANI is another tool to determine how similar genome sequences are to each other (Jain et

82     al. 2018b; Jain et al. 2018c). However, instead of building a sketch of a whole-genome sequence, FastANI

83     maintains the conceptual framework of BLAST-based ANI: it breaks the query genome into non-

84     overlapping fragments and only in the next step replaces BLAST with an alignment-free k-mer approach,

85     called MashMAP (Jain et al. 2018a). FastANI is 50 to 4000 times faster than BLAST-based ANI, while

86     inferring ANI accurately for ANI values as low as 80% (Jain et al. 2018b; Jain et al. 2018c).

87      The Life Identification Number (LIN) system is a genome similarity-based system to classify

88    individual organisms based on reciprocal ANI (Marakeby et al. 2014; Vinatzer et al. 2017; Vinatzer et al.

89    2016; Weisberg et al. 2015). A LIN consists of a series of positions, where each position indicates an ANI

90    threshold, from low to high, starting from the leftmost position. The LIN of a genome is assigned based

91    on the ANI to its most similar genome whose LIN has been already assigned. Therefore, the more similar

92    two genomes are to each other, the longer their LINs are identical starting from the leftmost position. A

93    group of strains sharing the same leading part of LINs is called a LINgroup, denoted by the shared part of

94    their LINs. It has been shown that LINgroups can be used to circumscribe groups of prokaryotes from the

95    genus level to the intraspecies level, almost reaching outbreak resolution (Vinatzer et al. 2016).

96      To analyze the diversity of a collection of prokaryotic genomes, computing all pairwise

97    comparisons cannot be avoided by any of the above methods and their implementations. However,

98    when dealing with a large number of genomes, pairwise comparisons are computationally expensive.

99    Furthermore, because of the ever-growing number of sequenced genomes and their frequent addition

100    to existing datasets, re-analysis of datasets each time new genomes are added becomes necessary.

101      Here we alleviate the bottleneck of pairwise ANI computations by developing LINflow, a pipeline

102    that efficiently constructs highly resolved similarity matrices from 70% to 99.9% ANI by combining the

103    speed of the MinHash-based sourmash tool (Brown & Irber 2016; Pierce et al. 2019) with the precision

104    of the BLAST-based pyani tool (Pritchard 2014) and the LIN concept (Weisberg et al. 2015). The obtained

105    results can then provide the basis for genome-based classification of prokaryotes.

106

107     **METHODS**

108         **Overview**

109     In short, to minimize the number of computationally expensive ANI computations when constructing a

110     genomic similarity matrix, LINflow sequentially adds genomes to a dataset, at each step efficiently

111     identifies the genome already in the dataset that is most similar to the newly added genome, precisely

112     calculates the ANI value only between this pair of genomes, and assigns a LIN to the new genome based

113     on this ANI value and the LIN of the most similar genome. The LINs are then used to further accelerate

114     the identification of the most similar genome and, most importantly, to efficiently infer all remaining

115     pairwise ANI values to construct the complete genome similarity matrix. In other words, the main

116     purpose of LINs in LINflow is to reduce the number of necessary ANI computations to one per genome

117     when computing a genome similarity matrix.

118         The LINflow flowchart is shown in **Figure 1**. When a new genome is added to the dataset,

119     LINflow identifies the genome that is most similar to this new genome among the genomes that were

120     previously added using the computationally efficient alignment-free tool sourmash (Brown & Irber 2016;

121     Pierce et al. 2019). This is accomplished via a two-step procedure, which consists of first identifying the

122     LINgroup to which the new genome belongs and then identifying the most similar genome in this

123     LINgroup. By default, LINflow uses the 95%-level LINgroup in this step, but this can be modified by the

124     user based on the expected genomic similarities in a specific dataset. The precise ANI value between the

125     two genomes is then computed using the more computationally expensive, but precise, pyani tool

126     (Pritchard 2014). LINflow uses this ANI value to assign a LIN to the new genome based on the LIN of its

127     most similar genome (which LIN was previously assigned based on the ANI value to its most similar

128     genome when that genome itself was previously added to the dataset). The assigned LINs can then be

129 used to infer all-against-all ANI values even though only a single ANI computation is performed for each

130 genome.

131   To make the results reusable and easily accessible in terms of reading and writing, a relational

132 database managed by MySQL (MySQL) is used to store data, with the schema shown in **Figure 2**. This

133 relational database connects tables with primary and foreign keys, and the connections between tables

134 are represented by arrows. The genome table stores the locations of the genome sequences. The

135 taxonomy table stores the taxonomic information corresponding to each genome in the database. LIN

136 schemes (*i.e.*, the number of LIN positions and the corresponding ANI thresholds), based on which LINs

137 are assigned, are kept in the Scheme table. Besides three default LIN schemes, new schemes can be

138 added by users so that LINs can be assigned according to the users' needs in resolution. LINs are

139 assigned with the three default schemes and with one user-defined scheme if there is any.

140   The individual steps of the pipeline are described in detail below.

141   **Generation and storing of signatures**

142 LINflow uses sourmash version 2.0 (Pierce et al. 2019) to generate signatures for both k=21 and k=51

143 with n=2000 (*i.e.*, each signature consists of 2000 hashes) for all genomes and stores them as individual

144 files. The first member of each 95%-level LINgroup is chosen as the representative genome and a copy of

145 its signature file is saved in a second directory together with signature files of all other representative

146 genomes.

147   **Choice of LIN scheme**

148 LINflow allows the user to choose from four default LIN schemes: (1) a 20-position LIN scheme that

149 ranges from 70% ANI to 99.999% ANI to cope with genus to strain level differentiation and is currently

150 used in LINbase (Tian et al. 2020) (**Table 1**), (2) a 300-position scheme with positions starting at 70% and

151 increasing by 0.1, and reaching 99.9% (used in this manuscript with the datasets listed below), (3) a

152    3000-position scheme starting at 70% and reaching 99.99% at 0.01% intervals (recommended for

153    constructing highly resolved similarity matrices for strains belonging to the same species), (4) a 300,000-

154    position scheme starting at 70% and reaching 99.99% at 0.00001% intervals. The user can also define

155    any custom LIN scheme with any number of positions using any ANI value of choice (however, it is not

156    advised to use ANI values below 70% since ANI does not reflect evolutionary relationships when ANI falls

157    below 70%).

158    **Initiation of LINflow**

159    LINflow, by default, includes the 20-position LIN scheme as one of the schemes during each run. An

160    arbitrary genome will be selected to assign the first LIN with 0 in each position. Its signature will be

161    generated and saved as the representative of the LINgroup $0_A0_B0_C0_D0_E0_F$ using the 20-position default

162    scheme both in the directories for representative genomes and this LINgroup.

163    **LIN assignment**

164    The new genome's ($G_{Query}$) signature $S_{Query}$ will be first queried against the representative genomes of all

165    existing 95%-level LINgroups (or F LINgroups) with k=21, using sourmash. Based on the analysis of more

166    than 6000 bacterial genomes from different families with k=21, the Jaccard similarity of 0.2475 was

167    found to correspond to 95% ANI (data not shown). If the highest Jaccard similarity $J$ to one of the

168    representative genomes $S_{Query}$ is above 0.2475, then the corresponding genome represents the 95%-

169    level LINgroup ($L_{95\%}$), belongs to. If $0.0025 < J <= 0.2475$, then the corresponding genome is at least 70%

170    similar to $G_{Query}$, which means they share at least the A position in the default LIN scheme. If $J <= 0.0025$,

171    the corresponding genome is less than 70% similar to $G_{Query}$.

172    If $J > 0.2475$, $S_{Query}$ will be queried against all members of $L_{95\%}$ by sourmash with k = 51. The

173    most similar genome $G_{Subject}$ according to Jaccard similarity in $L_{95\%}$ is identified as the most similar

174    genome to $G_{Query}$ in the entire database.

175    If $0.0025 < J \leq 0.2475$, $S_{Query}$ will be queried against all the members of $L_{95\%}$ by sourmash

176    with k = 21. The most similar genome $G_{Subject}$ according to Jaccard similarity in $L_{95\%}$ is identified as the

177    most similar genome to $G_{Query}$ in the entire database.

178    For the above cases, ANI between $G_{Query}$, and $G_{Subject}$ , $ANI_{Query}$, will be calculated with pyani. To

179    assign the LIN to the query genome, $G_{Subject}$'s LIN, $LIN_{Subject}$, will be used as the reference from A to the

180    last position that the ANI threshold is lower than or equal to $ANI_{Query}$, the first position that ANI

181    threshold is larger than $ANI_{Query}$ will be assigned a number that has not been used with the prefix in the

182    database, and the rest of the positions will be filled with 0s. For example, $ANI_{Query}$ = 95.4575%, it is over

183    95% at the F position but lower than 96% at the G position, so it will use $LIN_{Subject}$ from A to F as $LIN_{Query}$'s

184    A to F. At $LIN_{Query}$'s G position, a number that has never been used together with $LIN_{Subject}$'s prefix from A

185    to G will be assigned. Each of $LIN_{Query}$'s H to T positions will be assigned 0.

186    If $J \leq 0.0025$, no genome in the current database has over 70% ANI to $G_{Query}$, so that a new

187    number that has never been used in the A position before will be assigned to $LIN_{Query}$'s A position, and

188    the rest of $LIN_{Query}$ will be filled with 0s.

189    **Update of database and signature file system**

190    $G_{Query}$, $G_{Subject}$, $ANI_{Query}$, and $LIN_{Query}$ will all be written to the database. If $LIN_{Query}$ creates a new 95%-level

191    LINgroup, a new directory for this LINgroup will be created and $S_{Query}$ will be saved in this directory as a

192    member and as a representative genome with other representative genomes, otherwise, $S_{Query}$ will be

193    only saved in the existing 95%-level LINgroup it belongs to.

194    **Datasets**

195    We compared the performance of LINflow with sourmash, pyani, and FastANI for 4 datasets of

196    484 genomes in total. Dataset A includes 248 genome sequences belonging to the genus *Pseudomonas*.

197    Among the 248 genomes, 222 are from 46 named species. The remaining 26 genomes are from yet

198     unnamed and undescribed *Pseudomonas* species. Dataset B consists of 43 *Xanthomonas perforans*

199     genomes. Dataset C includes 140 genomes of the genus *Lysinibacillus*, whereby 96 of them belong to 27

200     named species. Dataset D includes 53 genomes from the species *Xylella fastidiosa* and two genomes

201     from the species *Xylella taiwanensis*. A separate dataset E with the genome sequence of *Pseudomonas*

202     *caeni* DSM 24390 was used to assess the computational speed of the above tools when adding a new

203     genome to the already-analyzed dataset A. **Supplementary Table 1** lists the genomes included in each

204     dataset and the actual genome sequences can be accessed directly in this repository:

205     https://code.vt.edu/linbaseproject/linflow_datasets .

206     **Comparison of tools**

207           LINflow was compared with pyani (blast option), sourmash, and FastANI in regard to speed,

208     memory usage, and accuracy. Parameters used when running these programs are listed in **Table 2**. Note

209     that for each pair of genomes A and B, pyani computes ANI by calculating both the ANI of A to B and the

210     ANI of B to A [8]. We used the average of the two pairwise ANI values. The calculations were executed

211     on a 2.4 GHz CPU on Cascades, an Advanced Research Computing (ARC) system at Virginia Tech, and the

212     execution time and memory usage were monitored by the job scheduler built in the system.

213           Hierarchical clustering using the complete linkage method was applied to the similarity matrices

214     of each dataset calculated by the four software suites using custom R scripts. Heatmaps were generated

215     based on the hierarchical clustering results to investigate whether FastANI, LINflow and Sourmash are

216     able to classify bacteria into species as accurate as pyani. Rows and columns of the heatmaps were

217     reordered to be in the same order as the heatmap generated by pyani.

218           Finally, pairwise Mantel tests (Mantel 1967) in combination with Pearson correlation

219     coefficients were performed using custom R scripts to determine how well the similarity matrices (ANI

220     for LINflow, FastANI, and pyani and Jaccard similarity for sourmash) obtained with the different tools.

221

222 **RESULTS**

223 **Computational speed and memory usage**

224 The CPU time needed to analyze datasets A, B, C, and D by each software is shown in **Table 3**. For

225 sourmash, two separate times are indicated since sourmash runs two separate commands: "sourmash

226 compute", which computes signatures, and "sourmash compare", which computes the pairwise Jaccard

227 similarity between signatures. Although the FastANI workflow is also split into two phases, the indexing

228 phase and the compute phase, the two phases cannot be executed separately. Therefore, only the total

229 execution time is shown.

230 Sourmash was the fastest out of the four tools, being 1000-4000 times faster than pyani,

231 depending on the dataset. FastANI and LINflow took a similar time to execute compared to pyani, being

232 84-253 times and 20-150 times faster, respectively.

233 The memory usage of LINflow was lower than pyani and lower than FastANI when analyzing

234 datasets A and C, which have relatively larger numbers of genomes compared to datasets B and D (**Table**

235 **4**). Furthermore, the time cost for adding a new genome to an existing dataset by LINflow did not

236 increase as significantly as for pyani and FastANI. This can be seen by comparing the time cost for adding

237 a new genome to dataset A (**Table 5**) with the average processing time for each genome in dataset A

238 **(Table 3**).

239 **Accuracy**

240 Similarity matrices obtained with sourmash, FastANI, and LINflow were compared to those obtained

241 with pyani, which we considered the gold standard, since it is exclusively based on BLAST reflecting the

242 original description of ANI (Konstantinidis & Tiedje 2005a; Konstantinidis & Tiedje 2005b). For sourmash,

243 we determined its performance separately for k=21 and k=51. For LINflow, we used two of the four

244 default schemes: the 20-position scheme used in LINbase (Tian et al. 2020) (see **Table 1** for the LIN

245     scheme and **Supplementary Table 1** for the result) and the 300-position scheme (with ANI values

246     increasing from 70% at the left-most LIN position to 99.9% at the right-most LIN position with 0.1%

247     intervals between neighboring positions). The LINbase scheme was used to assign LINs to the genomes

248     and classify them as LINgroups. The 300-position scheme was used to determine the ANI similarity

249     matrix. After similarity matrices were computed with all tools, heatmaps were generated to visualize the

250     genomic relatedness among the analyzed genomes.

251          Heatmaps derived from the ANI matrices obtained with pyani (**Figure 3A**), FastANI (**Figure 3B**),

252     and LINflow (**Figure 3C**) show the same species level (ANI ≥ 95%) clustering of the 247 *Pseudomonas*

253     genomes of dataset A visible as red blocks along the diagonal. Five major clusters are easily visible.

254     Cluster 1 consists of genomes belonging to *P. aeruginosa*, cluster 2 represents the species *P.*

255     *chlororaphis*, clusters 3, 4, and 5 constitute the *P. syringae* species complex and related genomes. Note

256     that LINflow not only classified the genomes as species but also distinguished intraspecific groups as

257     LINgroups. The LIN prefixes denote LINgroups and show both intergroup and intragroup relationships.

258          Sourmash was able to perform species-level clustering with k values of both 21 (**Figure 3D**) and

259     51 (**Figure 3E**). The obtained results suggest that Jaccard similarity calculated with k=51 only weakly

260     correlates with ANI for low ANI values compared to k=21, for example, clusters 2, 4 and 5. Instead of

261     genomes that are highly similar to each other, for example, genomes in cluster 5, k=51 provides higher

262     resolution than k=21.

263          To further evaluate the accuracy of LINflow compared to the other tools, the complete similarity

264     matrices obtained for all four datasets with all tools were compared with each other using the Mantel

265     test (Mantel 1967) using Pearson's correlation coefficients. Results are reported in **Figure 4**. One can

266     easily see how the results obtained with LINflow are highly correlated with those obtained with pyani

267     for datasets A, C, and D (Pearson correlation coefficient of 0.99 or 1) but not for dataset B, which has a

268     Pearson correlation coefficient of only 0.78. Since many pairs of genomes in set B have ANI values above

269   99.8% and differ from each other by less than 0.1%, we computed ANI values for set B also using the

270   3,000-position and a 300,000-position LIN scheme hypothesizing that the lower correlation was due to

271   rounding of ANI values to the first decimal place when using the 300-position LIN scheme. However,

272   switching to the 3,000-position and 300,000-position LIN scheme only increased the correlation

273   between LINflow and pyani slightly to a Pearson correlation coefficient of 0.82.

274         When comparing Pearson Correlation coefficients for the FastANI *versus* pyani comparison and

275   the sourmash *versus* pyani comparison with the LINflow *versus* pyani comparison, LINflow shows the

276   same or higher correlation with pyani for all datasets with the exception of dataset B, for which LINflow

277   shows the lowest correlation with pyani.

278         Finally, to determine how LINflow and FastANI correlate with pyani from low to high pairwise

279   ANI values, we plotted all pairwise ANI results obtained with LINflow and FastANI against pyani results

280   similar to what was done by Jain et al. (2018c). Figure 5 shows how FastANI and LINflow both correlate

281   very well with pyani for ANI values above 85% but deviate from pyani at ANI values from 85% to 70%.

282   While ANI values computed with FastANI are higher than the corresponding pyani ANI values in this

283   range, ANI values inferred by LINflow merge into a relatively small number of ANI values for many

284   different pyani ANI values (see Discussion for an explanation of this phenomenon).

285

286         **DISCUSSION**

287   Here we developed a new tool, LINflow, to efficiently compute genome similarity matrices for genome-

288   based classification of prokaryotes. We compared the performance of LINflow with that of sourmash,

289   FastANI, and pyani when analyzing four sets of genomes and when adding a new genome to an already-

290   analyzed dataset.

291         The execution time to compare each new genome to a growing dataset does not increase as

292   significantly for LINflow as for the other tools. This is because, for each genome, the LINflow algorithm

293    involves only a one-time sourmash signature generation, at most two sourmash signature comparisons,

294    and a one-time two-way ANI calculation with pyani between the new genome and its most similar

295    genome identified by sourmash. We thus expect that, with larger datasets, LINflow will outperform

296    FastANI in terms of speed and memory usage and that the relative increase in speed compared with

297    pyani will be even more significant.

298         The LIN approach had been shown previously to correlate well with core genome phylogenetic

299    trees within the genus *Pseudomonas* (Vinatzer et al. 2016). Previous comparisons of FastANI (Jain et al.

300    2018c), sourmash (Pierce et al. 2019), and pyani (Pritchard 2014) focused on accuracy in assigning

301    strains to species around the 95% ANI species threshold. Here we performed a species level comparison

302    and found LINflow to perform similarly well to sourmash, FastANI, and pyani (Figure 3). However, we

303    then went on and compared the relative performance of all four tools in creating complete similarity

304    matrices from around 70% ANI to almost 100% ANI. We did this comparison using the Mantel test

305    (Mantel 1967) and computing Pearson correlation coefficients for all pairwise tool comparisons, similar

306    to what was done by Jain et al. (2018c), when comparing ANI values obtained by FastANI with ANI values

307    obtained by BLAST. LINflow had the better correlation with pyani compared to sourmash and fastANI

308    except for dataset B, which is composed of highly similar genomes. Even changing the LIN scheme to the

309    300,000-position LIN scheme did not improve performance of LINflow showing that this was

310    independent of the resolution of the deployed LIN scheme. Moreover, for pairwise ANI values below

311    85%, we noticed how ANI values computed by LINflow for many different pairs of genomes merged into

312    a few identical ANI values. This is a direct result of the way LINflow infers ANI values based on

313    sequentially assigned LINs. For example, if a hypothetical group of genomes (group 1) contains genomes

314    that are all over 99% similar to each other and all genomes in a second group of genomes (group 2) are

315    all over 99% similar to each other but the first genome in group 1 that was assigned a LIN and the first

316    genome in group 2 that was assigned a LIN have a pairwise ANI value of only 70.75% , then all pairwise

317    ANI values between group 1 and group 2 genomes will be inferred to be 70.75% by LINflow. Because

318    LINs are assigned sequentially, this is an inherent limitation of LINflow that users need to weigh against

319    the time savings LINflow provides compared to pyani or FastANI.

320        The comparison between tools also revealed that FastANI's performance is affected by genomic

321    diversity of the analyzed datasets. In fact, FastANI's correlation with pyani was relatively low for

322    datasets A and C, where for each dataset, the genomes are from different species; FastANI's correlation

323    with pyani was high for datasets B and D, where most of the genomes for each dataset are from the

324    same species.

325        Finally, LINflow stores data in a MySQL relational database that organizes genomic data and the

326    corresponding metadata. MySQL is a relational database that can be accessed from its command-line

327    interface, various application programming interfaces, and graphical user interfaces. Therefore, users

328    can easily retrieve genome sequences for other analyses, *e.g.,* comparative genomics or customized

329    reference databases, by querying the database with filters of taxonomic information and/or LINs.

330

331        **CONCLUSIONS**

332    LINflow is a fast and memory-efficient pipeline to infer similarity among a large set of prokaryotic

333    genomes and achieves accuracy that approximates, but does not reach, that of pyani. Its ability to

334    quickly add new genome sequences to an already computed similarity matrix makes LINflow particularly

335    useful for projects when new genome sequences need to be regularly added to an existing dataset.

336    Further improvements to LINflow in regard to speed and resolution are underway.

337

338    **ACKNOWLEDGEMENTS**

339   The authors acknowledge Advanced Research Computing (ARC) at Virginia Tech for providing

340   computational resources and technical support that have contributed to the results reported within this

341   paper.

342

343        **REFERENCES**

344   BRENNER DJ. 1973. Deoxyribonucleic Acid Reassociation in the Taxonomy of Enteric Bacteria.

345        *International Journal of Systematic and Evolutionary Microbiology* 23:298-307.

346        https://doi.org/10.1099/00207713-23-4-298

347   Broder AZ. 1997. On the resemblance and containment of documents. Proceedings Compression and

348        Complexity of SEQUENCES 1997 (Cat No97TB100171). p 21-29.

349   Brown CT, and Irber L. 2016. sourmash: a library for MinHash scketching of DNA. *Journal of Open Source*

350        *Software* 1:27. doi: 10.21105/joss.00027

351   Galili T. 2015. dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical

352        clustering. *Bioinformatics (Oxford, England)* 31:3718-3720. 10.1093/bioinformatics/btv428

353   Goris J, Konstantinidis KT, Klappenbach JA, Coenye T, Vandamme P, and Tiedje JM. 2007. DNA–DNA

354        hybridization values and their relationship to whole-genome sequence similarities. *International*

355        *Journal of Systematic and Evolutionary Microbiology* 57:81-91. doi:10.1099/ijs.0.64483-0

356   Jain C, Dilthey A, Koren S, Aluru S, and Phillippy AM. 2018a. A Fast Approximate Algorithm for Mapping

357        Long Reads to Large Reference Databases. *Journal of computational biology : a journal of*

358        *computational molecular cell biology* 25:766-779. 10.1089/cmb.2018.0036

359   Jain C, Koren S, Dilthey A, Phillippy AM, and Aluru S. 2018b. A fast adaptive algorithm for computing

360        whole-genome homology maps. *Bioinformatics (Oxford, England)* 34:i748-i756.

361        10.1093/bioinformatics/bty597

362     Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT, and Aluru S. 2018c. High throughput ANI

363         analysis of 90K prokaryotic genomes reveals clear species boundaries. *Nature Communications*

364         9:5114. 10.1038/s41467-018-07641-9

365     Konstantinidis KT, and Tiedje JM. 2005a. Genomic insights that advance the species definition for

366         prokaryotes.   102:2567-2572.

367     Konstantinidis KT, and Tiedje JM. 2005b. Towards a Genome-Based Taxonomy for Prokaryotes.

368         187:6258-6264. 10.1128/JB.187.18.6258

369     Mantel N. 1967. The detection of disease clustering and a generalized regression approach. *Cancer*

370         *research*, 27(2), 209–220

371     Marakeby H, Badr E, Torkey H, Song Y, Leman S, Monteil CL, Heath LS, and Vinatzer BA. 2014. A System

372         to Automatically Classify and Name Any Individual Genome-Sequenced Organism Independently

373         of Current Biological Classification and Nomenclature. *PLOS ONE* 9:e89142.

374         10.1371/journal.pone.0089142

375     MySQL. https://dev.mysql.com/

376     Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, and Phillippy AM. 2016. Mash:

377         fast genome and metagenome distance estimation using MinHash. *Genome biology* 17:132-132.

378         10.1186/s13059-016-0997-x

379     Pierce N, Irber L, Reiter T, Brooks P, and Brown C. 2019. Large-scale sequence comparisons with

380         sourmash [version 1; peer review: 2 approved]. *F1000Research* 8.

381         10.12688/f1000research.19675.1

382     Pritchard L. 2014. pyani: Python module for average nucleotide identity analyses.

383         https://github.com/widdowquinn/pyani.

384     Rodriguez-R LM, Gunturu S, Harvey WT, Rosselló-Mora R, Tiedje JM, Cole JR, and Konstantinidis KT.

385         2018. The Microbial Genomes Atlas (MiGA) webserver: taxonomic and gene diversity analysis of

386        Archaea and Bacteria at the whole genome level. *Nucleic Acids Res* 46:W282-W288.

387        10.1093/nar/gky467

388    Sokal RR, and Rohlf FJ. 1962. The Comparison of Dendrograms by Objective Methods. *Taxon* 11:33-40.

389        10.2307/1217208

390    Stackebrandt E, and Goebel BM. 1994. Taxonomic Note : A Place for DNA-DNA Reassociation and s rRNA

391        Sequence Analysis in the Present Species Definition in Bacteriology.846-849.

392    Tian L, Huang C, Mazloom R, Heath LS, and Vinatzer BA. 2020. LINbase: a web server for genome-based

393        identification of prokaryotes as members of crowdsourced taxa. *Nucleic Acids Res*.

394        10.1093/nar/gkaa190

395    Vinatzer BA, Tian L, and Heath LS. 2017. A proposal for a portal to make earth's microbial diversity easily

396        accessible and searchable. *Antonie van Leeuwenhoek* 110:1271-1279. 10.1007/s10482-017-

397        0849-z

398    Vinatzer BA, Weisberg AJ, Monteil CL, Elmarakeby HA, Sheppard SK, and Heath LS. 2016. A Proposal for a

399        Genome Similarity-Based Taxonomy for Plant-Pathogenic Bacteria that Is Sufficiently Precise to

400        Reflect Phylogeny, Host Range, and Outbreak Affiliation Applied to Pseudomonas syringae sensu

401        lato as a Proof of Concept. *Phytopathology* 107:18-28. 10.1094/PHYTO-07-16-0252-R

402    Weisberg AJ, Elmarakeby HA, Heath LS, and Vinatzer BA. 2015. Similarity-Based Codes Sequentially

403        Assigned to Ebolavirus Genomes Are Informative of Species Membership , Associated Outbreaks

404        , and Transmission Chains.1-11. 10.1093/o

405

406        **FIGURE LEGENDS**

407    **Figure 1. Workflow of LINflow.** The flowchart of the LIN assignment algorithm used in LINflow.

408

409    **Figure 2. Database schema used by LINflow.** The relational database connects tables with primary and

410    foreign keys. Connections between tables are represented by arrows.

411

412    **Figure 3. Similarity matrices obtained by LINflow, FastANI, sourmash and pyani**. Heatmaps based on

413    hierarchical clustering using the complete linkage method using the similarity matrices obtained with

414    pyani (A), FastANI (B), LINflow (C), Sourmash k=21 (D), and k=51 (E) for dataset A. Cluster 1 corresponds

415    to *P. aeruginosa*, cluster 2 represents *P. chlororaphis*, clusters 3, 4, and 5 are different phylogroups within

416    the *P. syringae* species complex and related genomes. The same figure showing strain names is included

417    as Supplementary Figures 1 through 5 (corresponding to panels A through E). See Supplementary Figures

418    6 through 20 for heatmaps of datasets B, C, and D.

419

420    **Figure 4. Comparison of similarity matrices obtained by LINflow, FastANI, sourmash and pyani.**

421    Heatmaps showing Pearson correlation coefficients based on the Mantel test performed between the

422    similarity matrices obtained with pyANI, sourmash, FastANI, and LINflow for datasets A, B, C, and D.

423

424    **Figure 5. Correlation between the ANI results obtained with LINflow and FastANI and the ANI results**

425    **obtained with pyani (using the BLAST option) for datasets A through D. A** Plot showing the ANI values

426    computed by LINflow (in blue) and FastANI (in red) on the Y axis for ANI results obtained with pyani (X

427    axis) for all pairwise genome comparisons in datasets A through D. Pearson correlation coefficients for

428    FastANI *versus* pyani and LINflow *versus* pyani results and all other tool comparison results are shown in

429    Figure 4. **B** Plot showing the differences between the ANI values computed by FastANI and LINflow

430    compared to the pyani ANI values for all pairwise ANI values for datasets A though D.

431

432

The primary LIN assignment scheme of LINbase used to assign LINs by LINflow in this study.

| ANI | 70 | 75 | 80 | 85 | 90 | 95 | 96 | 97 | 98 | 98.5 | 99 | 99.25 | 99.5 | 99.75 | 99.9 | 99.925 | 99.95 | 99.975 | 99.99 | 99.999 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Position | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |

Software, sub-commands, and parameters used for pyani, sourmash, FastANI, and LINflow.

| Software | Version | Parameters |
|---|---|---|
| pyani[1] | 0.2.9 | -m ANIb -worker 100 |
| sourmash | 2.0.0 | -k 21, 51 -n 2000 |
| FastANI | 1.2 | -k 16 -t 1 |

2    [1]Multiprocessing was enabled.

Runtime of each software and sub-command used to analyze data sets A, B, C and D.

| Data set | No. of genomes | pyani[1] | sourmash compute | sourmash compare | FastANI[1] | LINflow |
|---|---|---|---|---|---|---|
| A | 247 | 60862m 6s | 15m 3s | 14s | 401m 18s | 829m 39s |
| B | 43 | 2107m 18s | 2m 34s | 2s | 17m 49s | 105m 50s |
| C | 140 | 7274m 18s | 7m 43s | 8s | 86m 12s | 171m 38s |
| D | 55 | 3292m 37s | 1m 41s | 3s | 13m 25s | 22m 58s |

2  [1]Total CPU time is listed for pyani and FastANI although multiprocessing was enabled.

Memory usage (GB) of each software and sub-command used to analyze data sets A, B, C and D.

| Data set | pyani | sourmash compute | sourmash compare | FastANI | LINflow |
|---|---|---|---|---|---|
| A | 5.8 | 0.05 | 1.4 | 4.6 | 0.6 |
| B | 1.2 | 0.04 | 0.1 | 0.5 | 0.3 |
| C | 5.7 | 0.05 | 1.2 | 2.2 | 0.3 |
| D | 1.3 | 0.05 | 0.1 | 0.4 | 0.3 |

Runtime of each software and sub-command used to add a single new genome to the analyzed data set A.

| Software and sub-command | CPU time |
|---|---|
| pyani | 258m 23s |
| sourmash compute | 0m 4s |
| sourmash search | 0m 11s |
| FastANI | 4m 4s |
| LINflow | 3m 36s |

# Figure 1

Workflow of LINflow.

The flowchart of the LIN assignment algorithm used in LINflow.

$G_{query}$

sourmash compute -k 21,51 -n 2000

$S_{query}$

sourmash search -k 21

Signatures of representative genomes

Jaccard similarity $J$ to the $S_{rep}$

$J <= 0.2475$ | $J > 0.2475$

$J <= 0.0025$ | $J > 0.0025$

$L95\%$ which $S_{rep}$ belongs to

$L95\%$ which $S_{rep}$ belongs to

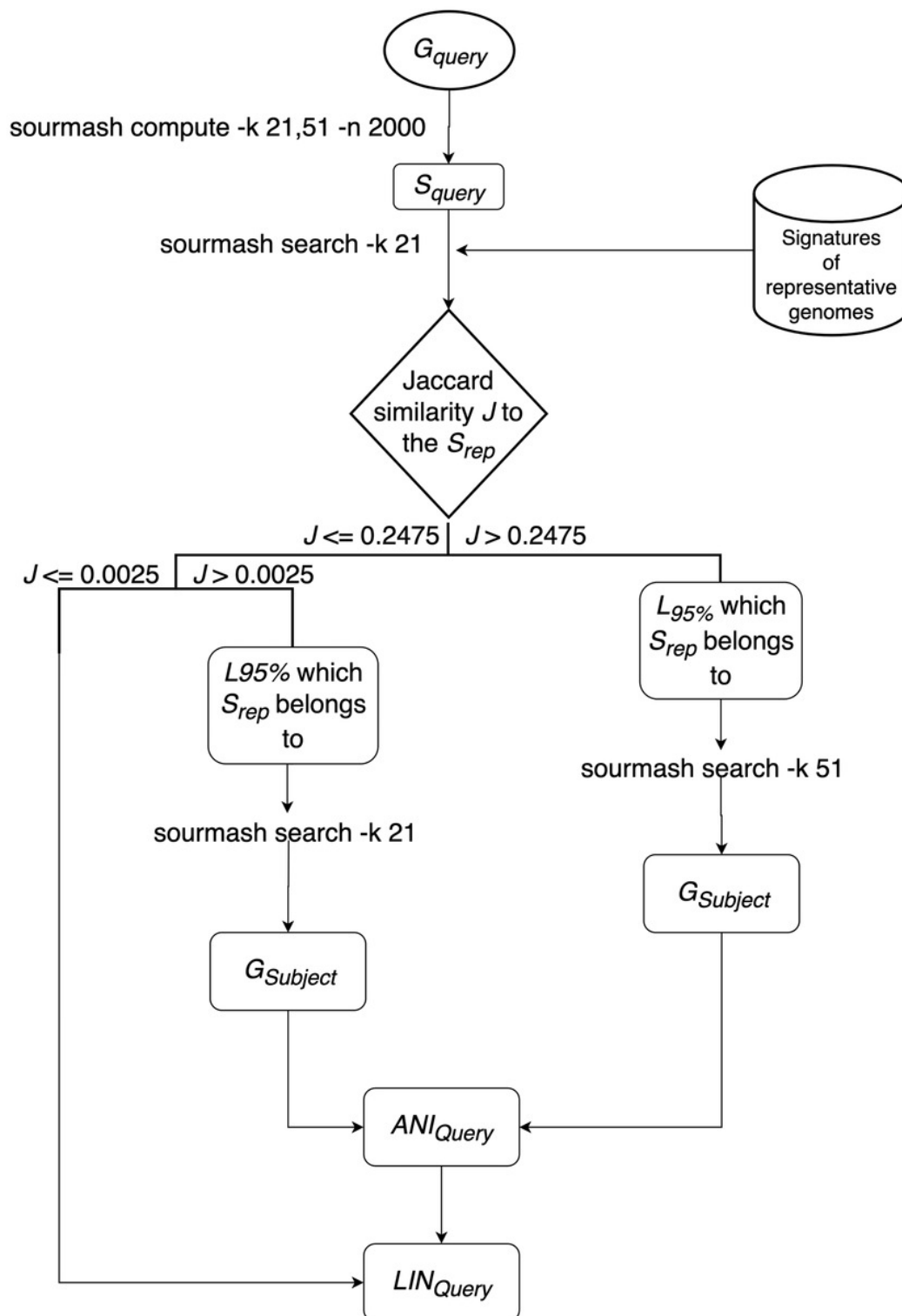
sourmash search -k 21

sourmash search -k 51

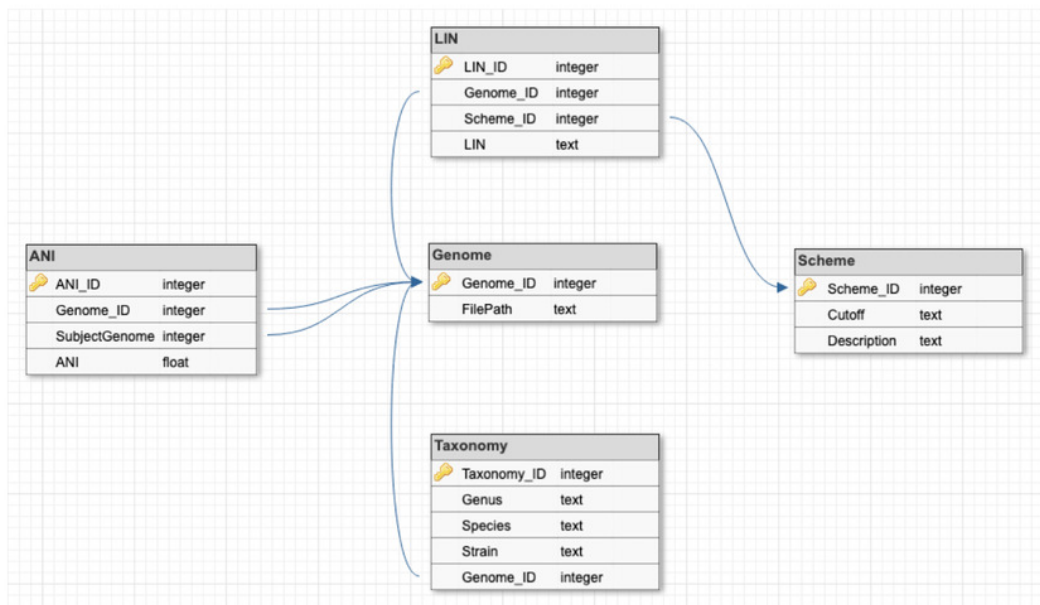
$G_{Subject}$

$G_{Subject}$

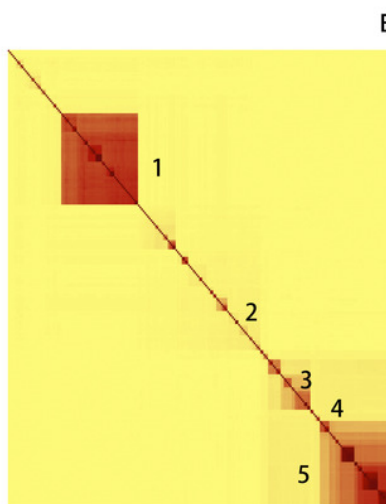
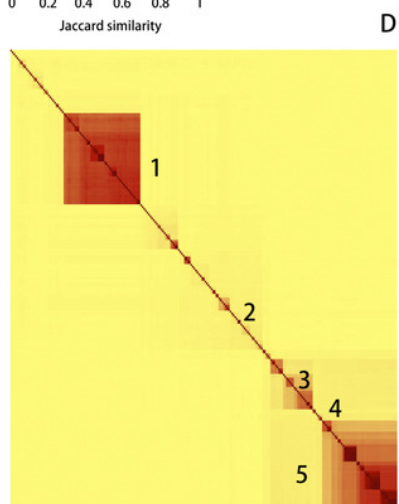
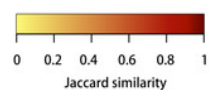
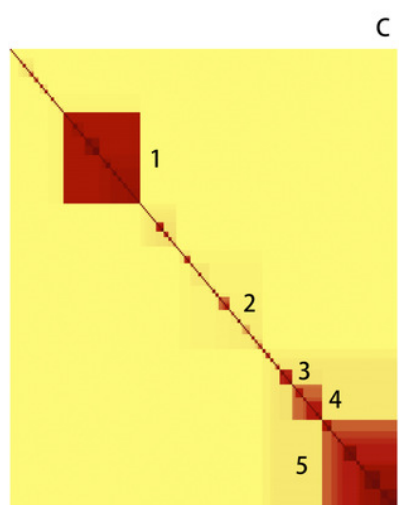
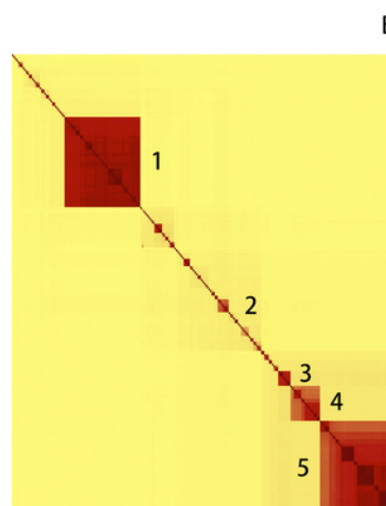
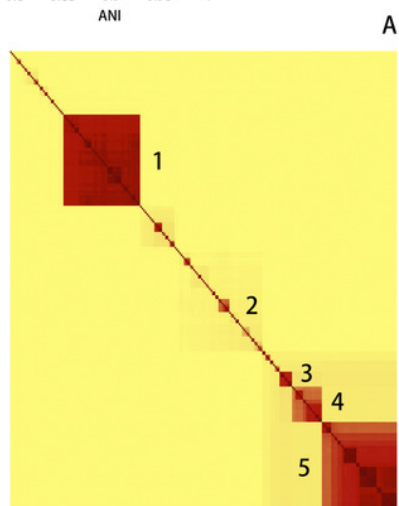
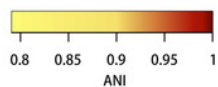
$ANI_{Query}$

$LIN_{Query}$

# Figure 2

Database schema used by LINflow.

The relational database connects tables with primary and foreign keys. Connections between tables are represented by arrows.

# Figure 3

Similarity matrices obtained by LINflow, FastANI, sourmash and pyani.

Heatmaps based on hierarchical clustering using the complete linkage method using the similarity matrices obtained with pyani (A), FastANI (B), LINflow (C), Sourmash k=21 (D), and k=51 (E) for dataset A. Cluster 1 corresponds to *P. aeruginosa*, cluster 2 represents *P. chlororaphis*, clusters 3, 4, and 5 are different phylogroups within the *P. syringae* species complex and related genomes. The same figure showing strain names is included as Supplementary Figures 1 through 5 (corresponding to panels A through E). See Supplementary Figures 6 through 20 for heatmaps of datasets B, C, and D.

# Figure 4

Comparison of similarity matrices obtained by LINflow, FastANI, sourmash and pyani

Heatmaps showing Pearson correlation coefficients based on the Mantel test performed between the similarity matrices obtained with pyANI, sourmash, FastANI, and LINflow for datasets A, B, C, and D.

Data set A

| | | | | |
|---|---|---|---|---|
| LINflow | 0.99 | 0.87 | 0.86 | 0.78 | 1 |
| Sourmash k=51 | 0.8 | 0.71 | 0.98 | 1 | 0.78 |
| Sourmash k=21 | 0.87 | 0.77 | 1 | 0.98 | 0.86 |
| FastANI | 0.88 | 1 | 0.77 | 0.71 | 0.87 |
| pyANI | 1 | 0.88 | 0.87 | 0.8 | 0.99 |

pyANI  FastANI  Sourmash k=21  Sourmash k=51  LINflow

Data set B

| | | | | | | |
|---|---|---|---|---|---|---|
| LINflow_300k | 0.82 | 0.82 | 0.84 | 0.87 | 0.87 | 1 | 1 |
| LINflow_3k | 0.82 | 0.81 | 0.83 | 0.86 | 0.85 | 1 | 1 |
| LINflow_300 | 0.78 | 0.8 | 0.83 | 0.85 | 1 | 0.85 | 0.87 |
| Sourmash k=51 | 0.88 | 0.88 | 0.99 | 1 | 0.85 | 0.86 | 0.87 |
| Sourmash k=21 | 0.86 | 0.85 | 1 | 0.99 | 0.83 | 0.83 | 0.84 |
| FastANI | 0.91 | 1 | 0.85 | 0.88 | 0.8 | 0.81 | 0.82 |
| pyANI | 1 | 0.91 | 0.86 | 0.88 | 0.78 | 0.82 | 0.82 |

pyANI  FastANI  Sourmash k=21  Sourmash k=51  LINflow_300  LINflow_3k  LINflow_300k

Data set C

| | | | | |
|---|---|---|---|---|
| LINflow | 0.99 | 0.69 | 0.74 | 0.61 | 1 |
| Sourmash k=51 | 0.62 | 0.34 | 0.97 | 1 | 0.61 |
| Sourmash k=21 | 0.76 | 0.42 | 1 | 0.97 | 0.74 |
| FastANI | 0.61 | 1 | 0.42 | 0.34 | 0.69 |
| pyANI | 1 | 0.61 | 0.76 | 0.62 | 0.99 |

pyANI  FastANI  Sourmash k=21  Sourmash k=51  LINflow

Data set D

| | | | | |
|---|---|---|---|---|
| LINflow | 1 | 1 | 0.79 | 0.62 | 1 |
| Sourmash k=51 | 0.65 | 0.6 | 0.97 | 1 | 0.62 |
| Sourmash k=21 | 0.81 | 0.77 | 1 | 0.97 | 0.79 |
| FastANI | 1 | 1 | 0.77 | 0.6 | 1 |
| pyANI | 1 | 1 | 0.81 | 0.65 | 1 |

pyANI  FastANI  Sourmash k=21  Sourmash k=51  LINflow

# Figure 5

Correlation between the ANI results obtained with LINflow and FastANI and the ANI results obtained with pyani (using the BLAST option) for datasets A through D.

**A** Plot showing the ANI values computed by LINflow (in blue) and FastANI (in red) on the Y axis for ANI results obtained with pyani (X axis) for all pairwise genome comparisons in datasets A through D. Pearson correlation coefficients for FastANI *versus* pyani and LINflow *versus* pyani results and all other tool comparison results are shown in Figure 4. **B** Plot showing the differences between the ANI values computed by FastANI and LINflow compared to the pyani ANI values for all pairwise ANI values for datasets A though D.