Open and closed pangenomes with *k*-mer counting

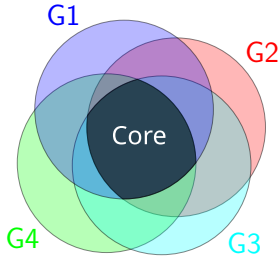
Luca Parmigiani

University of Bielefeld
DSB

13 Jun 2022

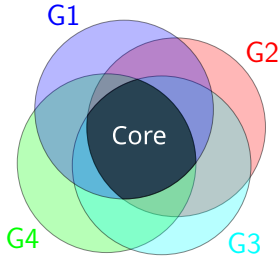
Pangenome (Tettelin et al., 2005)

- **Pangenome:** the set of all distinct genes present in a species
  - **Core genes**: present in all genomes
  - **Dispensable genes**: present in some genomes

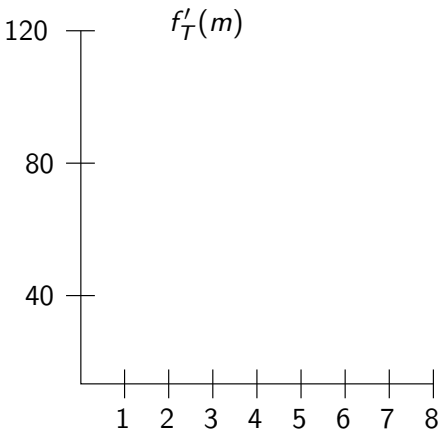


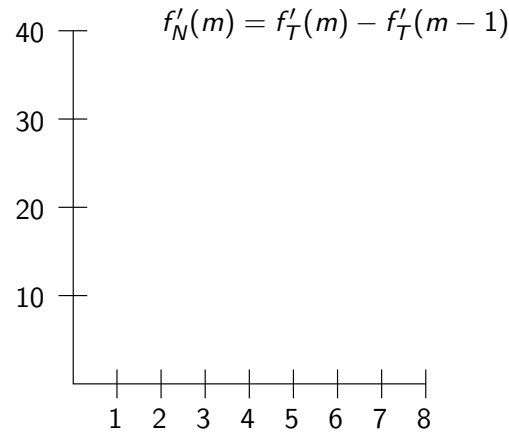
Pangenome (Tettelin et al., 2005)

- **Pangenome:** the set of all distinct genes present in a species
  - **Closed pangenome**
    The number of distinct genes is asymptotic
  - **Open pangenome**
    The number of distinct genes keeps increasing

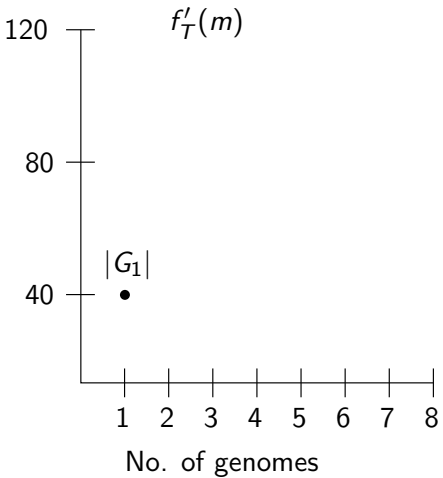


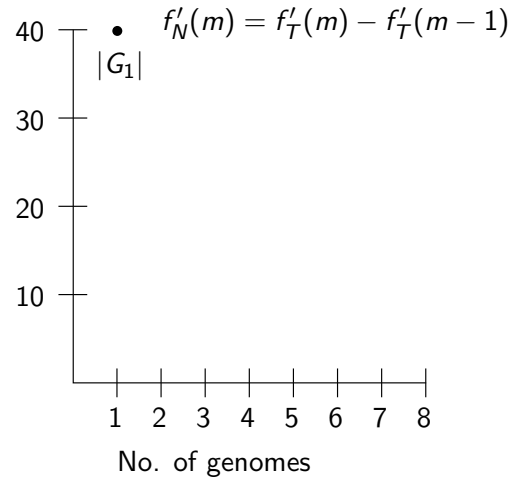
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8



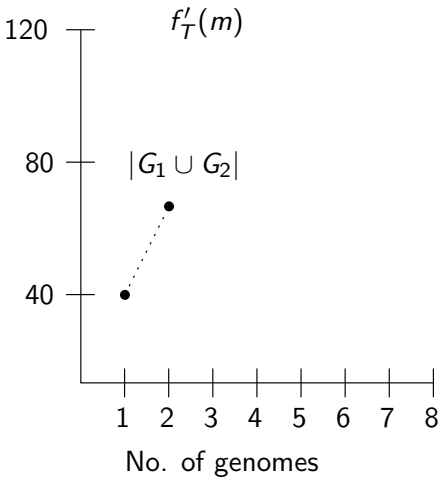


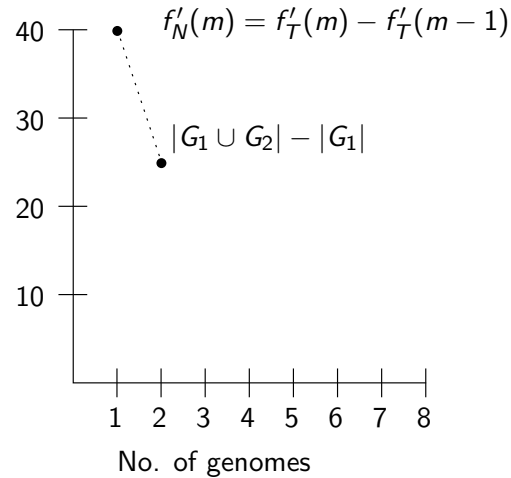
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8





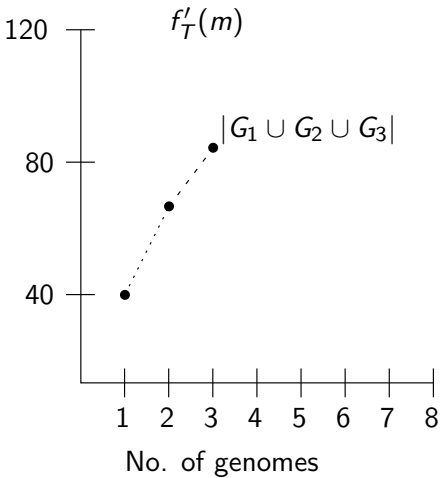
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8

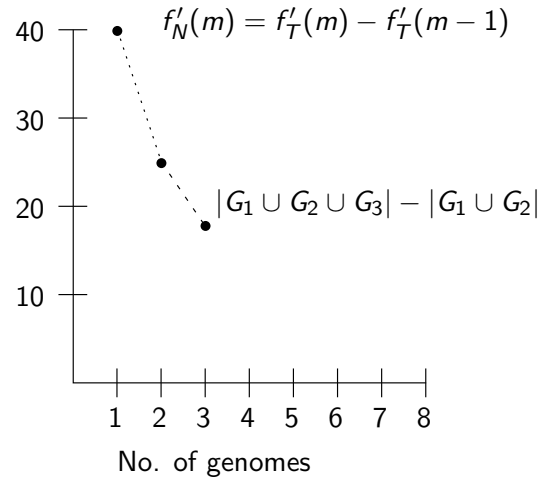




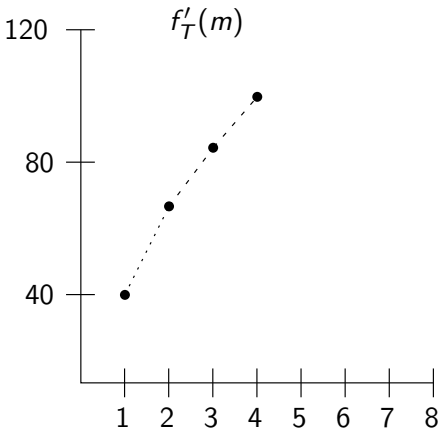
Pangenome

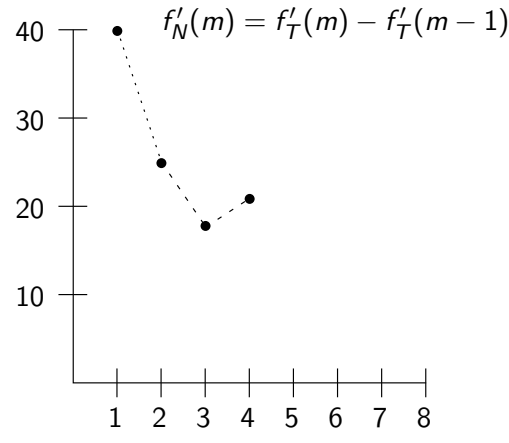
$G_1 \quad G_2 \quad G_3 \quad G_4 \quad G_5 \quad G_6 \quad G_7 \quad G_8$



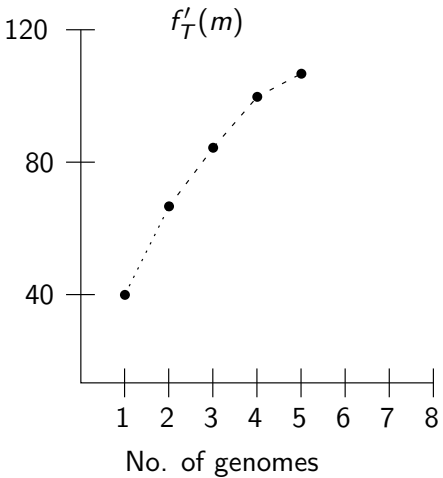


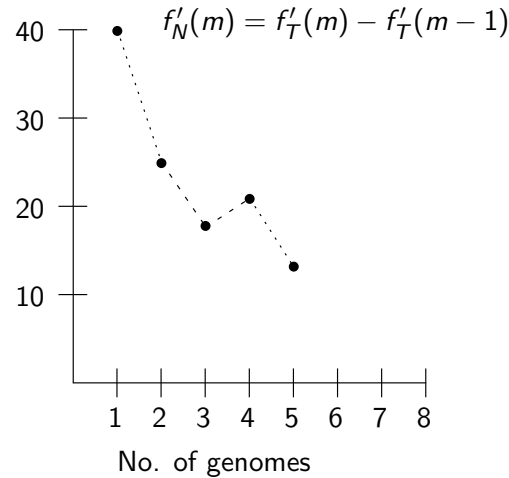
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8



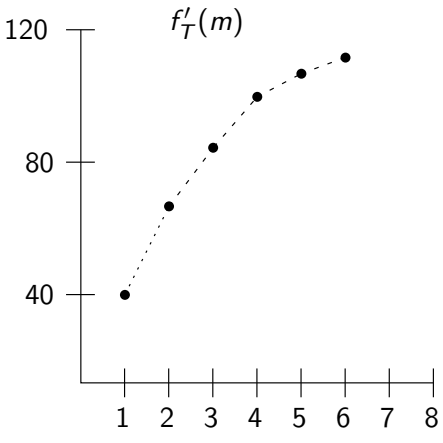


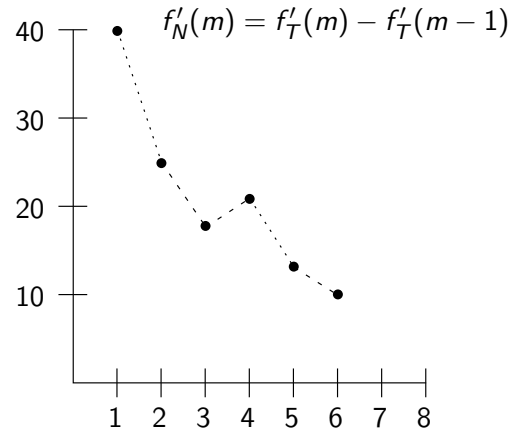
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8





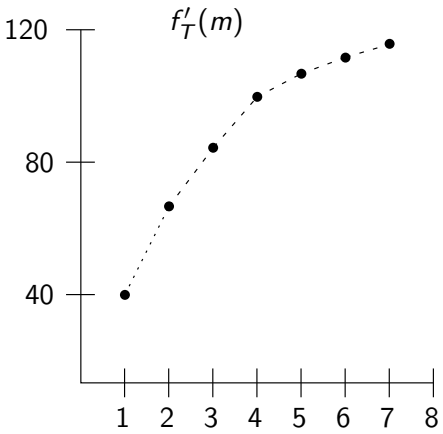
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8

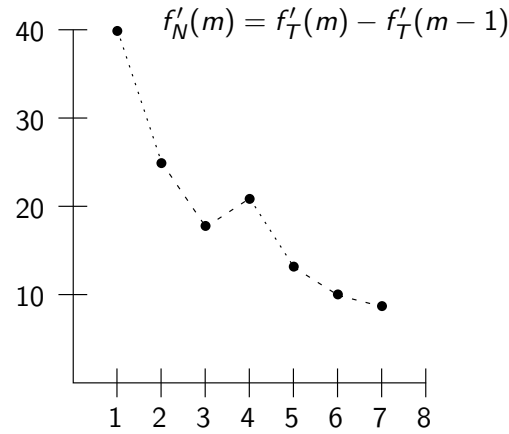




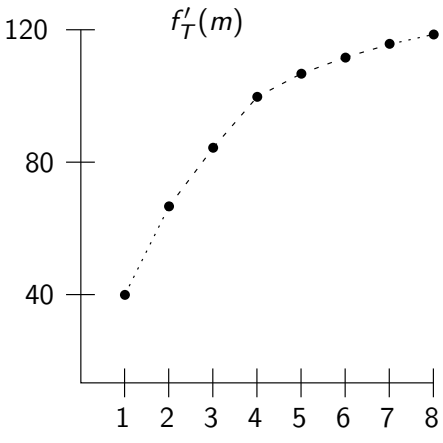
Pangenome

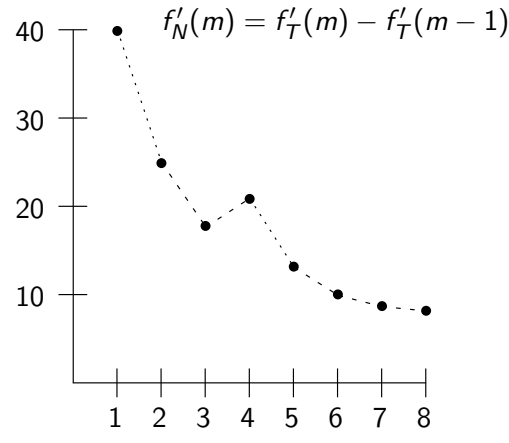
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8



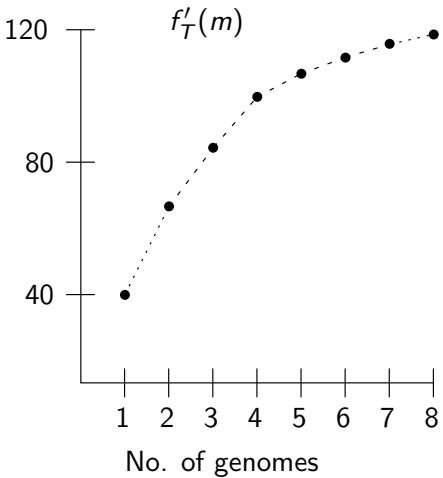


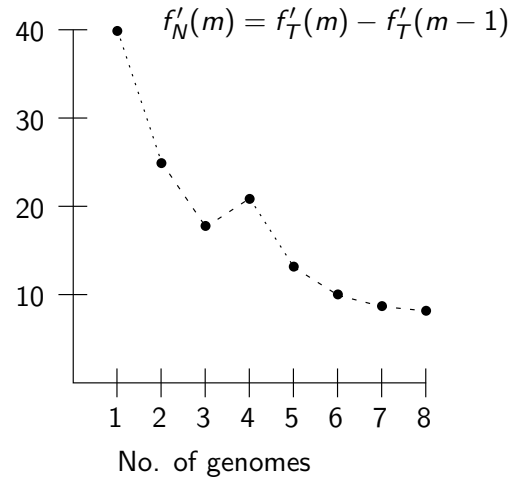
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8



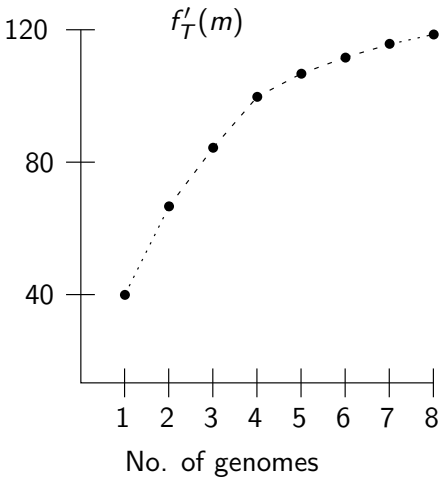


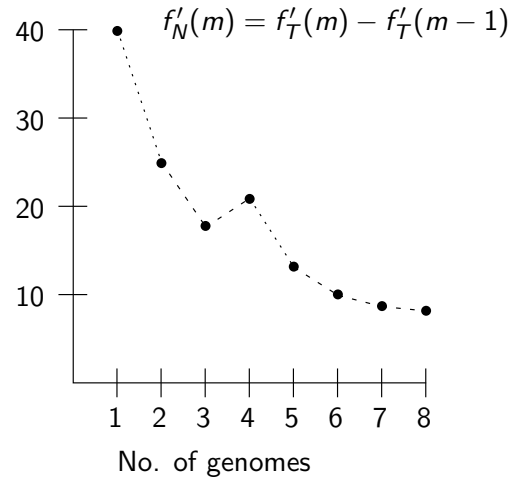
G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8



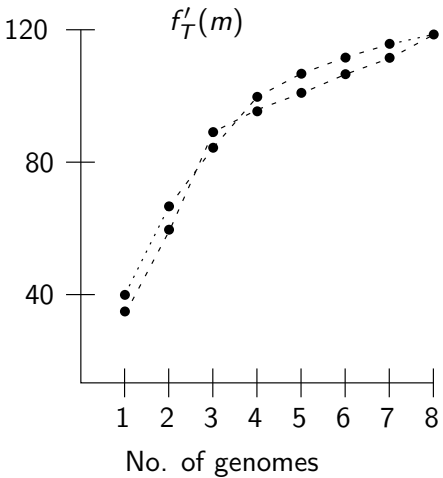


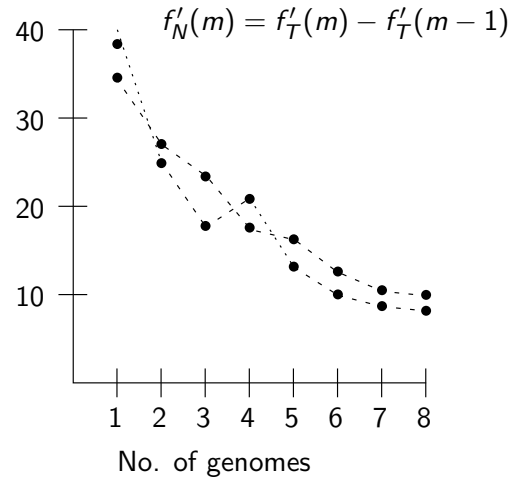
G_4 G_1 G_7 G_5 G_8 G_2 G_3 G_6



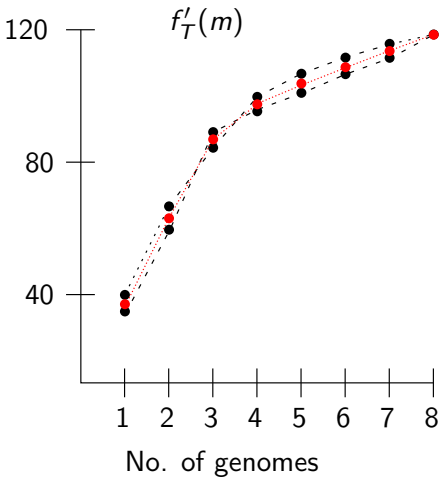


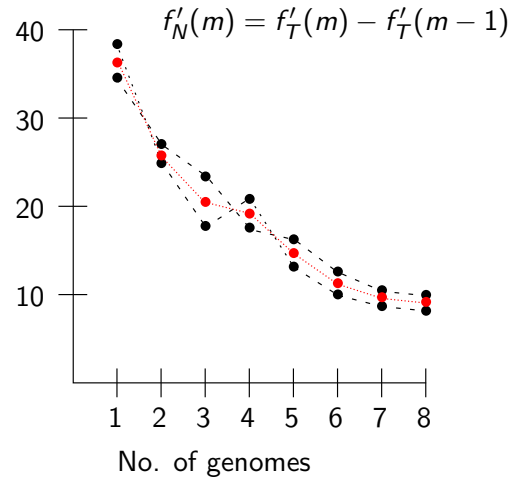
G_4 G_1 G_7 G_5 G_8 G_2 G_3 G_6



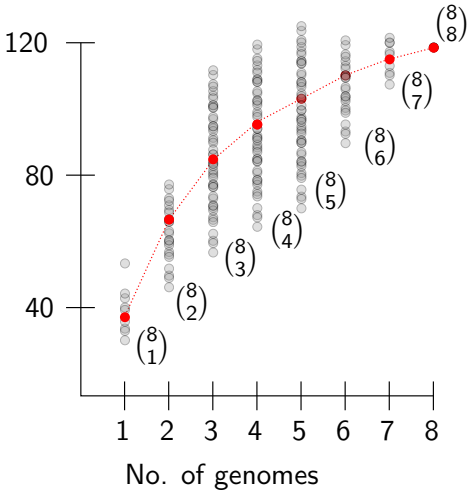


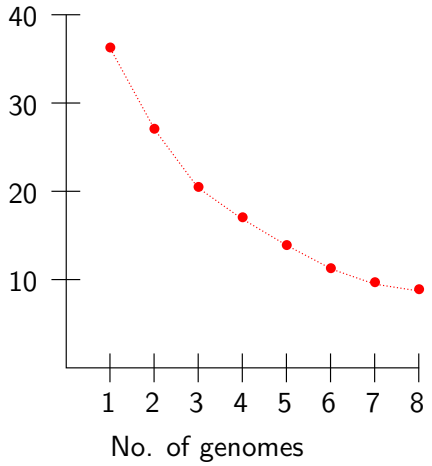
$G_4 \quad G_1 \quad G_7 \quad G_5 \quad G_8 \quad G_2 \quad G_3 \quad G_6$





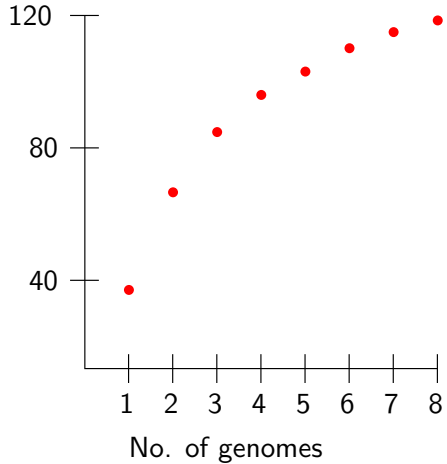
Pangenome

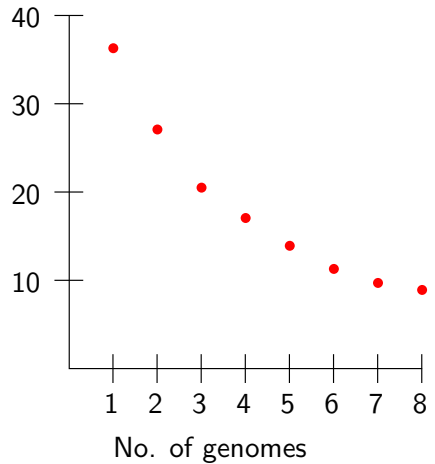

120
80
40
1
2
3
4
5
6
7
8
No. of genomes


40
30
20
10
1
2
3
4
5
6
7
8
No. of genomes

Tettelin et al., 2008 - Open closed Pangenome

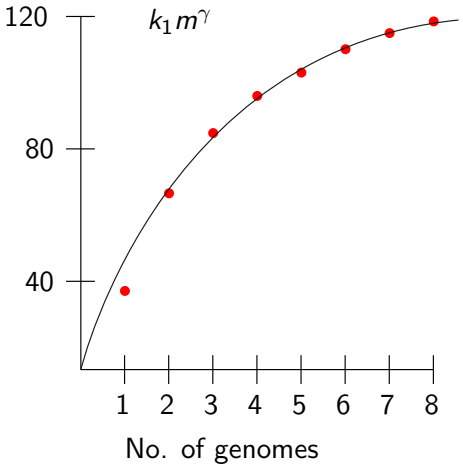
Heap's law

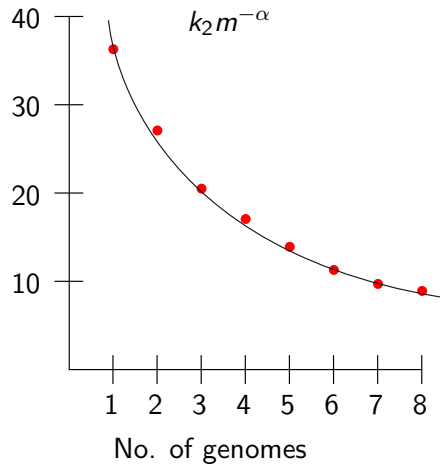




Tettelin et al., 2008 - Open closed Pangenome

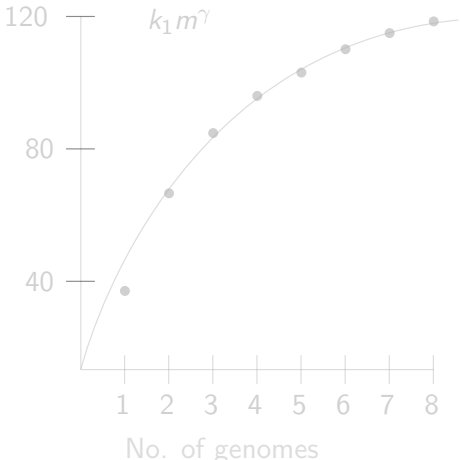
Heap's law

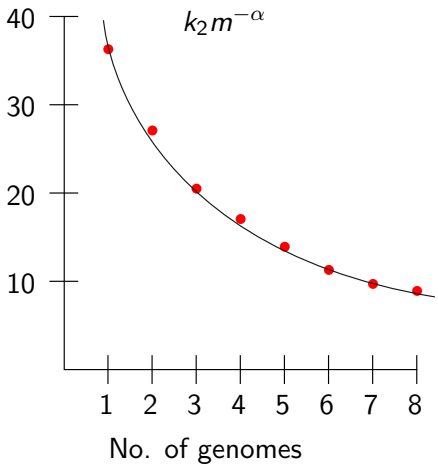




Tettelin et al., 2008 - Open closed Pangenome

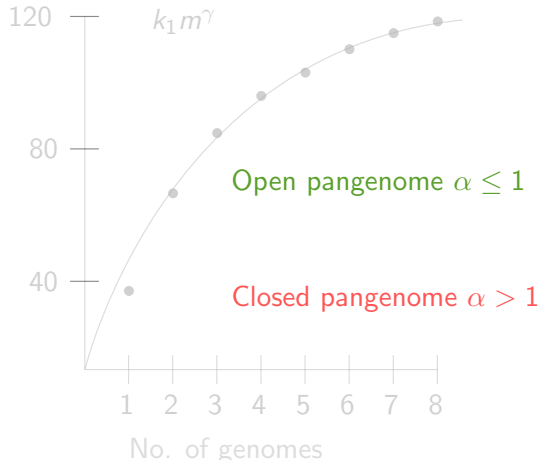
Heap's law

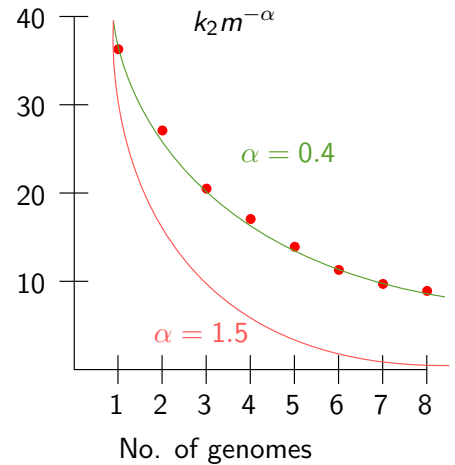




Tettelin et al., 2008 - Open closed Pangenome

Heap's law





Definitions

- Let a genome, G be a set of items (genes, k-mers).

Definitions

- Let a genome, G be a set of items (genes, k-mers).
- Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a set of genomes

Definitions

- Let a genome, G be a set of items (genes, k-mers).
- Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a set of genomes
- $\mathcal{P}(\mathcal{G})$ is the power set representing all possible subsets of $\mathcal{G}$

$$\mathcal{P}(\{G_1, G_2, G_3\}) = \{\emptyset, \{G_1\}, \{G_2\}, \{G_3\}, \{G_1, G_2\}, \{G_1, G_3\}, \{G_2, G_3\}, \{G_1, G_2, G_3\}\}$$

Definitions

- Let a genome, G be a set of items (genes, k-mers).
- Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a set of genomes
- $\mathcal{P}(\mathcal{G})$ is the power set representing all possible subsets of $\mathcal{G}$

$$\mathcal{P}(\{G_1, G_2, G_3\}) = \{\emptyset, \{G_1\}, \{G_2\}, \{G_3\}, \{G_1, G_2\}, \{G_1, G_3\}, \{G_2, G_3\}, \{G_1, G_2, G_3\}\}$$

- $\mathcal{G}_m = \{S \in \mathcal{P}(\mathcal{G}) \mid |S| = m\}$ the set of subsets of $\mathcal{G}$ of cardinality m

$$\mathcal{G}_2 = \{\{G_1, G_2\}, \{G_1, G_3\}, \{G_2, G_3\}\}$$

Definitions

- Let a genome, G be a set of items (genes, k-mers).
- Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a set of genomes
- $\mathcal{P}(\mathcal{G})$ is the power set representing all possible subsets of $\mathcal{G}$

$$\mathcal{P}(\{G_1, G_2, G_3\}) = \{\emptyset, \{G_1\}, \{G_2\}, \{G_3\}, \{G_1, G_2\}, \{G_1, G_3\}, \{G_2, G_3\}, \{G_1, G_2, G_3\}\}$$

- $\mathcal{G}_m = \{S \in \mathcal{P}(\mathcal{G}) \mid |S| = m\}$ the set of subsets of $\mathcal{G}$ of cardinality m

$$\mathcal{G}_2 = \{\{G_1, G_2\}, \{G_1, G_3\}, \{G_2, G_3\}\}$$

- The computation of $f_T(m)$ requires taking the average of $\binom{n}{m}$ values,

$$f_T(m) = \frac{1}{\binom{n}{m}} \sum_{S \in \mathcal{G}_m} \left| \bigcup_{G \in S} G \right|$$

$$\text{e.g., } f_T(2) = \frac{|G_1 \cup G_2| + |G_1 \cup G_3| + |G_2 \cup G_3|}{\binom{3}{2}}$$

- Pangenome growth:

$$f_T(m) = \frac{1}{\binom{n}{m}} \sum_{S \in \mathcal{G}_m} \left| \bigcup_{G \in S} G \right|$$

- Average number of new genes that are added when adding the m-th genome:

$$f_N(m) = \begin{cases} 0 & \text{if } m = 0 \\ f_T(m) - f_T(m-1) & \text{otherwise} \end{cases}$$

Why using k-mer for pangenome openness

PRO

Why using k-mer for pangenome openness

PRO

- Including non-coding region

Why using k-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)

Why using k-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans
- No need for gene homology

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans
- No need for gene homology
- Simple and fast

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans
- No need for gene homology
- Simple and fast

CON

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans
- No need for gene homology
- Simple and fast

CON

- Less informative

Why using *k*-mer for pangenome openness

PRO

- Including non-coding region
- Can be performed directly on sequencing reads (no assembly)
- Does not need any annotation
  - Genes, trust in silico predicition without curation
  - Missed, un-annotated genes, or ORFans
- No need for gene homology
- Simple and fast

CON

- Less informative
- Choice of *k*

Obtaining f_T

Genes

- Pan-matrix

	G_1	G_2		...			G_n
gene_1	0	0	1	1	0	1	1	1
gene_2	1	1	0	1	1	0	0	0
...				...				
gene_l	0	1	0	0	1	0	0	0

- Sampling with different orderings of $\mathcal{G}$

Obtaining f_T **efficiently**

- Without *approximating* f_T (i.e., sampling)

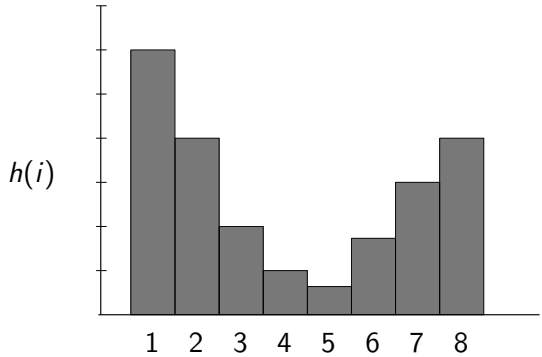
Obtaining f_T **efficiently**

- Without *approximating* f_T (i.e., sampling)
- Without considering multiple *orderings* of $\mathcal{G}$

Obtaining f_T **efficiently**

- Without *approximating* f_T (i.e., sampling)
- Without considering multiple *orderings* of $\mathcal{G}$
- How:

$$h(i) = \text{number of items occuring in } \textbf{exactly } i \text{ genomes}$$



Obtaining f_T efficiently

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

$$f_T(3) = (|G_1 \cup G_2 \cup G_3| + |G_1 \cup G_2 \cup G_4| + |G_1 \cup G_2 \cup G_5| + |G_1 \cup G_3 \cup G_4| + |G_1 \cup G_3 \cup G_5| + |G_1 \cup G_4 \cup G_5| + |G_2 \cup G_3 \cup G_4| + |G_2 \cup G_3 \cup G_5| + |G_2 \cup G_4 \cup G_5| + |G_3 \cup G_4 \cup G_5|) \frac{1}{\binom{5}{3}}$$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

$$f_T(3) = (|G_1 \cup G_2 \cup G_3| + |G_1 \cup G_2 \cup G_4| + |G_1 \cup G_2 \cup G_5| + |G_1 \cup G_3 \cup G_4| + |G_1 \cup G_3 \cup G_5| + |G_1 \cup G_4 \cup G_5| + |G_2 \cup G_3 \cup G_4| + |G_2 \cup G_3 \cup G_5| + |G_2 \cup G_4 \cup G_5| + |G_3 \cup G_4 \cup G_5|) \frac{1}{\binom{5}{3}}$$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

$$f_T(3) = (|G_1 \cup G_2 \cup G_3| + |G_1 \cup G_2 \cup G_4| + |G_1 \cup G_2 \cup G_5| + |G_1 \cup G_3 \cup G_4| + |G_1 \cup G_3 \cup G_5| + |G_1 \cup G_4 \cup G_5| + |G_2 \cup G_3 \cup G_4| + |G_2 \cup G_3 \cup G_5| + |G_2 \cup G_4 \cup G_5| + |G_3 \cup G_4 \cup G_5|) \frac{1}{\binom{5}{3}}$$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

$$f_T(3) = (|G_1 \cup G_2 \cup G_3| + |G_1 \cup G_2 \cup G_4| + |G_1 \cup G_2 \cup G_5| + |G_1 \cup G_3 \cup G_4| + |G_1 \cup G_3 \cup G_5| + |G_1 \cup G_4 \cup G_5| + |G_2 \cup G_3 \cup G_4| + |G_2 \cup G_3 \cup G_5| + |G_2 \cup G_4 \cup G_5| + |G_3 \cup G_4 \cup G_5|) \frac{1}{\binom{5}{3}}$$

- Percent of $S \in \mathcal{G}_m$ that **do not have item** x

$$\binom{n-i}{m} \Big/ \binom{n}{m}$$

Obtaining f_T efficiently

- $\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}$
- The item x is present in G_1, G_2

$$f_T(3) = (|G_1 \cup G_2 \cup G_3| + |G_1 \cup G_2 \cup G_4| + |G_1 \cup G_2 \cup G_5| + |G_1 \cup G_3 \cup G_4| + |G_1 \cup G_3 \cup G_5| + |G_1 \cup G_4 \cup G_5| + |G_2 \cup G_3 \cup G_4| + |G_2 \cup G_3 \cup G_5| + |G_2 \cup G_4 \cup G_5| + |G_3 \cup G_4 \cup G_5|) \frac{1}{\binom{5}{3}}$$

- Percent of $S \in \mathcal{G}_m$ that **do not have item** x

$$\binom{n-i}{m} \Big/ \binom{n}{m} = \frac{(n-i)^{\underline{m}}}{n^{\underline{m}}}$$

$$n^{\underline{m}} = \overbrace{n(n-1)\dots(n-m+1)}^{m \text{ factors}}$$

Obtaining f_T efficiently

$$f_T(m) = \sum_{i=1}^{n} h(i)\left(1 - \frac{(n-i)^{\underline{m}}}{n^{\underline{m}}}\right)$$

Obtaining f_T efficiently

$$
\begin{aligned}
f_T(m) &= \sum_{i=1}^{n} h(i)\left(1 - \frac{(n-i)^{\underline{m}}}{n^{\underline{m}}}\right) \\
&= \sum_{i=1}^{n} h(i) - \frac{1}{n^{\underline{m}}} \sum_{i=1}^{m} (n-i)^{\underline{m}}
\end{aligned}
$$

Obtaining f_T efficiently

$$
\begin{aligned}
f_T(m) &= \sum_{i=1}^{n} h(i)\left(1 - \frac{(n-i)^{\underline{m}}}{n^{\underline{m}}}\right) \\
&= \sum_{i=1}^{n} h(i) - \frac{1}{n^{\underline{m}}} \sum_{i=1}^{m} (n-i)^{\underline{m}}
\end{aligned}
$$

- $(n-i)^{\underline{j+1}} = (n-i-j+1)(n-i)^{\underline{j}}$

Obtaining f_T efficiently

$$
\begin{aligned}
f_T(m) &= \sum_{i=1}^{n} h(i)\left(1 - \frac{(n-i)^{\underline{m}}}{n^{\underline{m}}}\right) \\
&= \sum_{i=1}^{n} h(i) - \frac{1}{n^{\underline{m}}} \sum_{i=1}^{m} (n-i)^{\underline{m}}
\end{aligned}
$$

- $(n-i)^{\underline{j+1}} = (n-i-j+1)(n-i)^{\underline{j}}$
- Time complexity: $O(n^2)$

How to obtain $h(i)$

Genes

Pan-matrix

	G_1	G_2			…			G_n
gene_1	0	0	1	1	0	1	1	1
gene_2	1	1	0	1	1	0	0	0
…					…			
gene_l	0	1	0	0	1	0	0	0

How to obtain $h(i)$

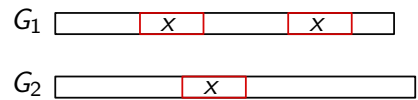
Genes

Pan-matrix

	G_1	G_2		...		G_n
gene_1	0	0	1	1	0	1	1	1
gene_2	1	1	0	1	1	0	0	0
...				...				
gene_l	0	1	0	0	1	0	0	0

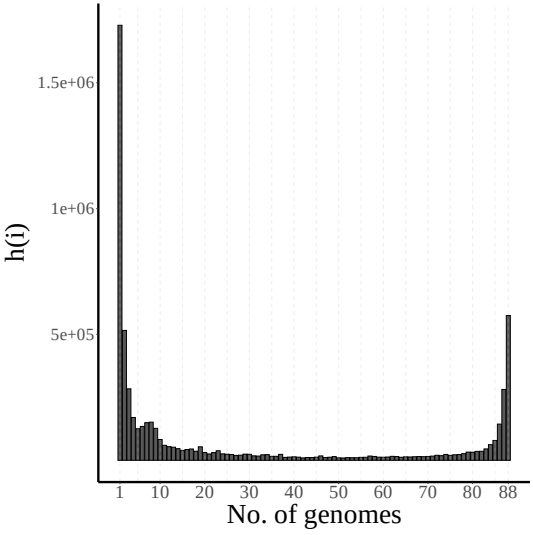
k-mers

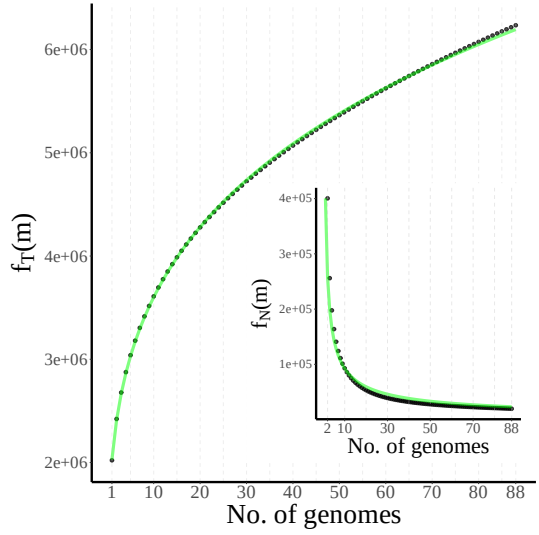
Modified version of YAK[1](yak-hist)



- k-mer x has multiplicity 2

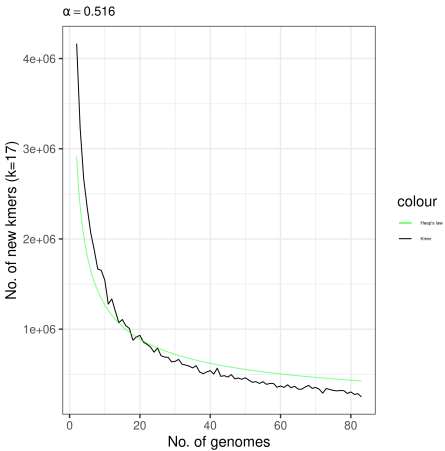
[1]https://github.com/lh3/yak


h(i)
1.5e+06
1e+06
5e+05
1
10
20
30
40
50
60
70
80
88
No. of genomes

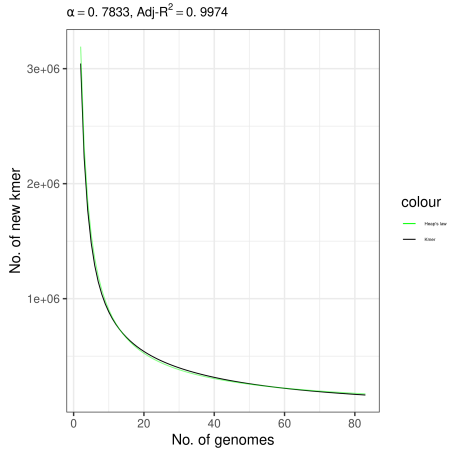

$f_T(m)$
6e+06
5e+06
4e+06
3e+06
2e+06
1
10
20
30
40
50
60
70
80
88
No. of genomes
$f_N(m)$
4e+05
3e+05
2e+05
1e+05
2
10
30
50
70
88
No. of genomes

Results

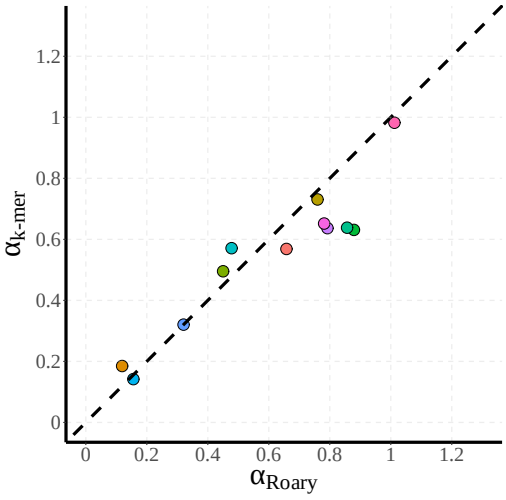
Permutation



Without permutation



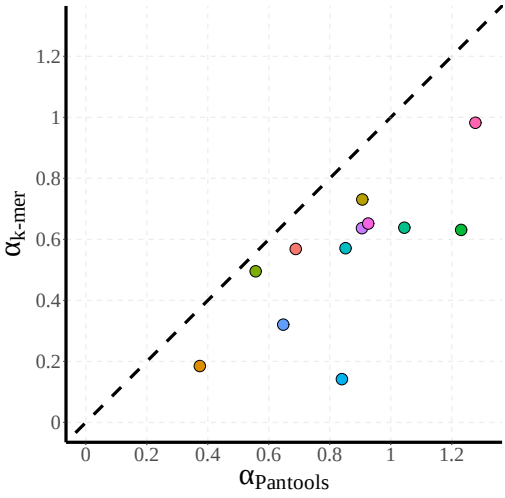
Results

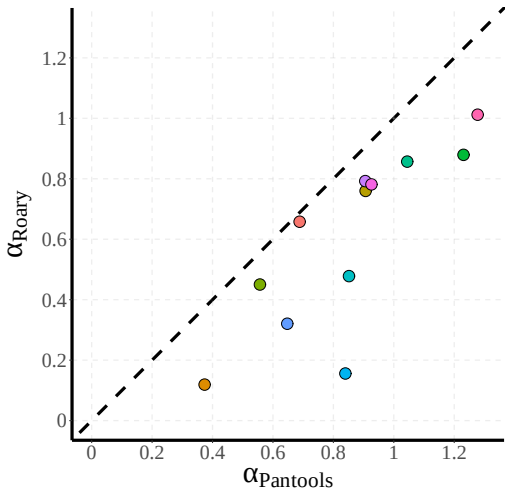


species

- Bacillus_cereus
- Buchnera_aphidicola
- Campylobacter_jejuni
- Clostridium_botulinum
- Coxiella_burnetii
- Francisella_tularensis
- Helicobacter_pylori
- Prochlorococcus_marinus
- Rhodopseudomonas_palustris
- Streptococcus_pneumoniae
- Streptococcus_pyogenes
- Yersinia_pestis

Results


1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.2
$\alpha_{\text{k-mer}}$
α_{Pantools}

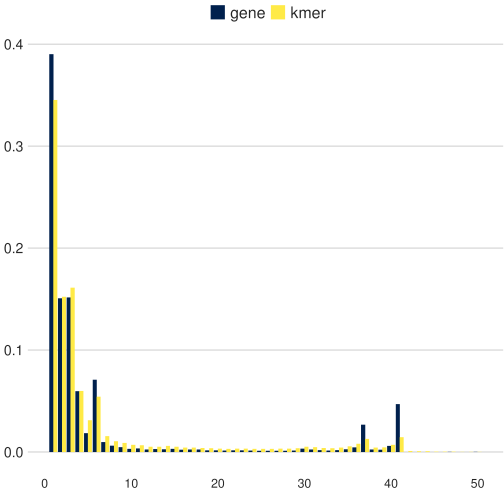

1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.2
α_{Roary}
α_{Pantools}

Normalized Kendal Tau distance

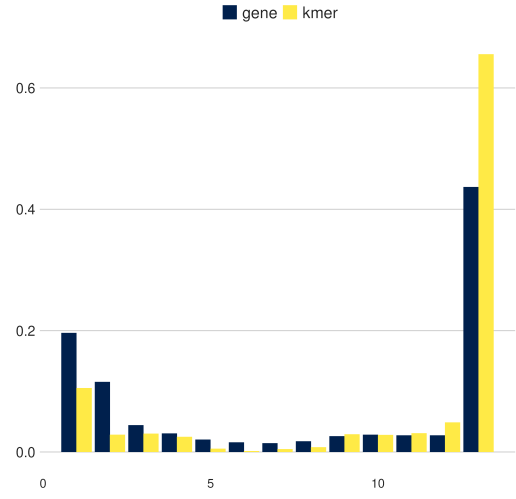
Counts the number of **pairwise disagreements** between two ranking lists divided by the total number of pairwise comparison, $n(n-1)/2$.

- Roary vs. k-mer: 0.167
- Roary vs. Pantools: 0.106
- k-mer vs. Pantools: 0.182

Histrogram, $h(i)$ of genes (Roary) and k-mers



Clostridium botulinum



Coxiella burnetii

Jensen-Shannon divergence

Species	**JSD**
Francisella tularensis	0.065
Coxiella burnetii	0.061
Yersinia pestis	0.11
Streptococcus pneumoniae	0.019
Helicobacter pylori	0.048
Clostridium botulinum	0.023
Streptococcus pyogenes	0.042
Prochlorococcus marinus	0.0048
Campylobacter jejuni	0.054
Buchnera aphidicola	0.023
Bacillus cereus	0.026
Escherichia coli	0.014

Core

$$f_C(m) = \frac{1}{\binom{n}{m}} \sum_{S \in \mathcal{G}_m} \left| \bigcap_{G \in S} G \right|$$

Compute f_C efficiently

$$f_C(m) = \frac{1}{n^{\underline{m}}} \sum_{i=m}^{n} h(i) i^{\underline{m}}$$

Quorum, q

$$f_C(m,q) = \frac{1}{\binom{n}{m}} \sum_{\lceil q*m \rceil}^{n} h(i) \sum_{j=\lceil q*m \rceil}^{i} \frac{\binom{i}{j}}{\binom{n-i}{m-j}}$$

Thank you for your attention