LEIDEN UNIVERSITY MEDICAL CENTER

k-mer profiling
Theory and Implementation

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics

Introduction

Raw sequencing data

When do we work with *raw data*:

- Unknown reference.
- No time for analysis.

Introduction

Raw sequencing data

When do we work with *raw data*:

- Unknown reference.
- No time for analysis.

If the reference sequence is unknown, we can still do:

- Quality control.
- Coverage estimation.

Introduction

Raw sequencing data

When do we work with *raw data*:

- Unknown reference.
- No time for analysis.

If the reference sequence is unknown, we can still do:

- Quality control.
- Coverage estimation.

Compare raw datasets:

- Quality control.
- Phylogeny.
- Metagenomics.

Introduction

Counting k-mers

We choose a k and count all occurrences of substrings of length k.

Introduction

Counting k-mers

We choose a k and count all occurrences of substrings of length k.

The counts of these substrings serve as a fingerprint of the dataset.

Introduction

Counting k-mers

We choose a k and count all occurrences of substrings of length k.

The counts of these substrings serve as a fingerprint of the dataset.

But, these counts contain a lot more information.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	0	TA	0
AC	1	CC	0	GC	0	TC	0
AG	0	CG	0	GG	0	TG	0
AT	0	CT	0	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA . . .

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	0	TA	0
AC	1	CC	0	GC	0	TC	0
AG	0	CG	0	GG	0	TG	0
AT	0	CT	1	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA . . .

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	0	TA	1
AC	1	CC	0	GC	0	TC	0
AG	0	CG	0	GG	0	TG	0
AT	0	CT	1	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	0	TA	1
AC	1	CC	0	GC	0	TC	0
AG	1	CG	0	GG	0	TG	0
AT	0	CT	1	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA . . .

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	1	TA	1
AC	1	CC	0	GC	0	TC	0
AG	1	CG	0	GG	0	TG	0
AT	0	CT	1	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

A 2-mer profile

ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...

Figure 1: Indexing.

We use a sliding window to find all k-mers ($k = 2$).

AA	0	CA	0	GA	1	TA	1
AC	2	CC	0	GC	0	TC	0
AG	1	CG	0	GG	0	TG	0
AT	0	CT	1	GT	0	TT	0

Table 1: 2-mer profile.

Observations are stored in a table.

Introduction

Counting k-mers

We choose a k and count all occurrences of substrings of length k.

Introduction

Counting k-mers

We choose a k and count all occurrences of substrings of length k.

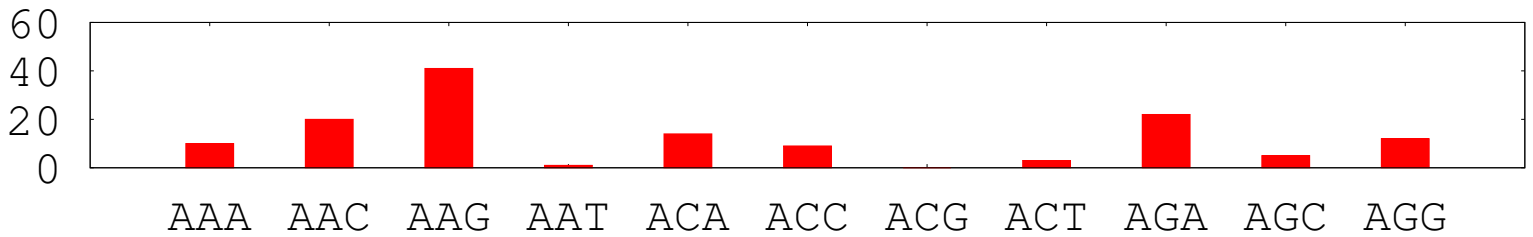


Figure 2: A profile of 3-mer counts.

In Figure 2 we see a part of 3-mer counts; **AAA** occurs 10 times, **AAC** occurs 20 times, etc.

Introduction

Choosing k

k can not be too small:

- $k = 1$ will result in loss of all subsequence information.
- $k = 2$ will give you information about di-nucleotides.
- There is only one unique 10-mer in Human (hg18).

Introduction

Choosing k

k can not be too small:

- $k = 1$ will result in loss of all subsequence information.
- $k = 2$ will give you information about di-nucleotides.
- There is only one unique 10-mer in Human (hg18).

But, k can not be too large either:

- Almost all 18-mers are unique in the Human genome.
- Since this is not error tolerant:
  - Read errors.
  - Assembly errors.

Introduction

Choosing k

k can not be too small:

- $k = 1$ will result in loss of all subsequence information.
- $k = 2$ will give you information about di-nucleotides.
- There is only one unique 10-mer in Human (hg18).

But, k can not be too large either:

- Almost all 18-mers are unique in the Human genome.
- Since this is not error tolerant:
  - Read errors.
  - Assembly errors.

We have a solution for this (explained later in this presentation).

k-mer profiles

Indexing

nucleotide	binary	decimal
`A`	`00`	0
`C`	`01`	1
`G`	`10`	2
`T`	`11`	3

Table 2: Nucleotide encoding table.

We use an additional trick to store profiles efficiently.

k-mer profiles

Indexing

nucleotide	binary	decimal
`A`	`00`	0
`C`	`01`	1
`G`	`10`	2
`T`	`11`	3

Table 2: Nucleotide encoding table.

We use an additional trick to store profiles efficiently.

First notice that we can encode a nucleotide in two bits.

k-mer profiles

Indexing

substring	binary	decimal
AAAA	00 00 00 00	0
AAAC	00 00 00 01	1
⋮	⋮	⋮
GATC	10 00 11 01	141
⋮	⋮	⋮
TTTT	11 11 11 11	255

Table 3: Encoding strings.

We can concatenate the *binary encoding*.

k-mer profiles

Indexing

substring	binary	decimal
AAAA	00 00 00 00	0
AAAC	00 00 00 01	1
⋮	⋮	⋮
GATC	10 00 11 01	141
⋮	⋮	⋮
TTTT	11 11 11 11	255

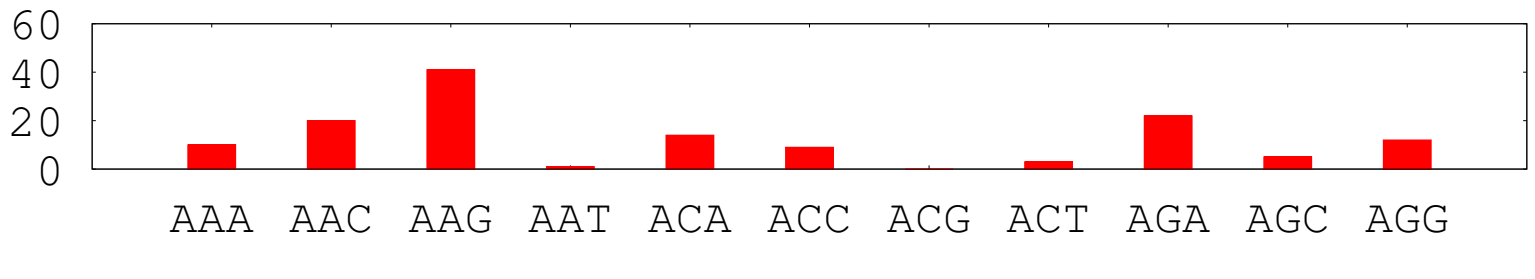
Table 3: Encoding strings.

We can concatenate the *binary encoding*.

There is no need to store the substrings.

k-mer profiles

Comparing k-mer profiles



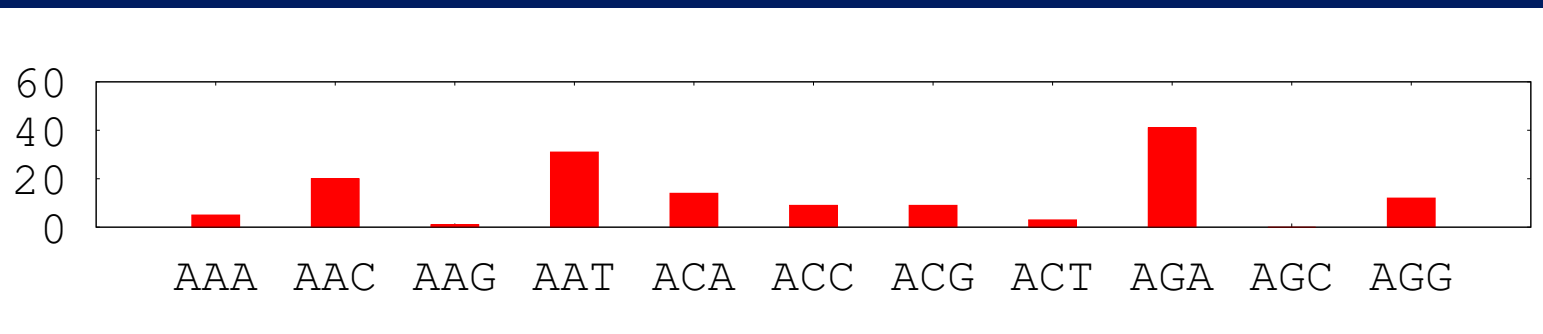


Figure 3: Two profiles of k-mer counts.

k-mer profiles

Comparing k-mer profiles

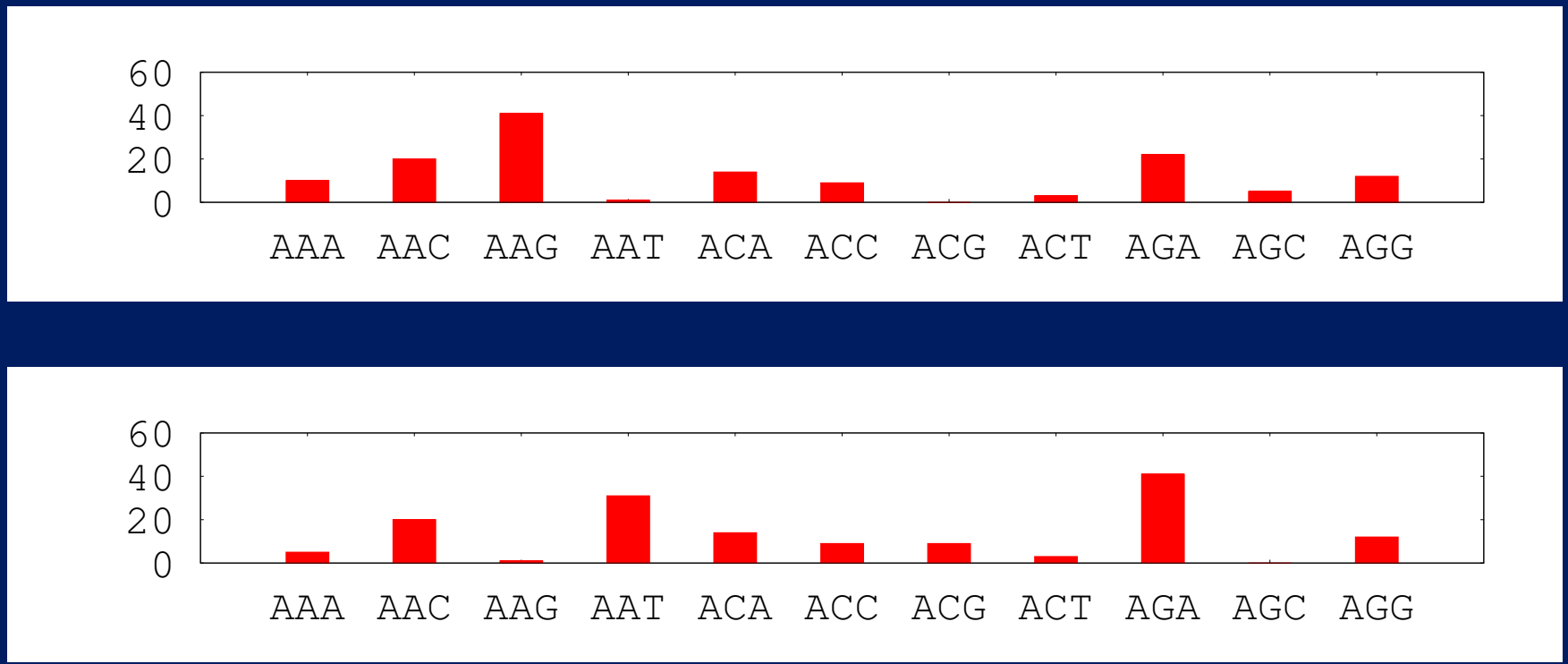


Figure 3: Two profiles of k-mer counts.

How to express this difference with one value.

k-mer profiles

Pairwise distance function

We use the following function:

$$f(x,y) = \frac{|x-y|}{(x+1)(y+1)}$$

k-mer profiles

Pairwise distance function

We use the following function:

$$f(x,y) = \frac{|x-y|}{(x+1)(y+1)}$$

Properties:

- $f(0,1) = \frac{1}{2}$
- $f(0,1) > f(7,8)$

k-mer profiles

Pairwise distance function

We use the following function:

$$f(x,y) = \frac{|x-y|}{(x+1)(y+1)}$$

Properties:

- $f(0,1) = \frac{1}{2}$
- $f(0,1) > f(7,8)$

This is desirable:

- The fact that a k-mer is not present is more important than the number of times it is present.
- Differences in the low end of the spectrum are more important than ones at the high end.

k-mer profiles

Multiset distance function

Let f be a function $f : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ with finite supremum M and the following properties:

$$
\begin{array}{lcll}
f(x,y) & = & f(y,x) & \text{for all } x,y \in \mathbb{R}_{\geq 0} \\
f(x,x) & = & 0 & \text{for all } x \in \mathbb{R}_{\geq 0} \\
f(x,0) & \geq & M/2 & \text{for all } x \in \mathbb{R}_{>0} \\
f(x,y) & \leq & f(x,z) + f(z,y) & \text{for all } x,y,z \in \mathbb{R}_{\geq 0}
\end{array}
$$

k-mer profiles

Multiset distance function

Let f be a function $f : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with finite supremum M and the following properties:

$$
\begin{array}{lcll}
f(x,y) & = & f(y,x) & \text{for all } x,y \in \mathbb{R}_{\geq 0} \\
f(x,x) & = & 0 & \text{for all } x \in \mathbb{R}_{\geq 0} \\
f(x,0) & \geq & M/2 & \text{for all } x \in \mathbb{R}_{>0} \\
f(x,y) & \leq & f(x,z) + f(z,y) & \text{for all } x,y,z \in \mathbb{R}_{\geq 0}
\end{array}
$$

For a multiset X, let $S(X)$ denote its underlying set. For multisets X, Y with $S(X), S(Y) \subseteq \{1, 2, \ldots, n\}$ we define

$$
d_f(X,Y) = \frac{\sum_{i=1}^{n} f(x_i, y_i)}{|S(X) \cup S(Y)| + 1}
$$

k-mer profiles

Strand balance

When analysing a dataset:

- We either see a k-mer or its reverse complement (50% chance of either).
- If sequenced in sufficient depth, we expect a balance between forward and reverse complement k-mers.

k-mer profiles

Strand balance

```
ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...
TGATCTGGTGAATGATCTCTGATCTGGTGGTGATCTGGT...
```

Figure 4: Double stranded DNA.

k-mer profiles

Strand balance

```
ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...
TGATCTGGTGAATGATCTCTGATCTGGTGGTGATCTGGT...
```

Figure 4: Double stranded DNA.

k-mer profiles

Strand balance

```
ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...
TGATCTGGTGAATGATCTCTGATCTGGTGGTGATCTGGT...
```

Figure 4: Double stranded DNA.

k-mer profiles

Strand balance

```
ACTAGACCACTTACTAGAGACTAGACCACCACTAGACCA...
TGATCTGGTGAATGATCTCTGATCTGGTGGTGATCTGGT...
```

Figure 4: Double stranded DNA.

So, the number of times we see **TAGAGACTAGA** must be roughly the same as for **TCTAGTCTCTA**.

k-mer profiles

Strand balance

How to calculate it:

- We can split a profile into a forward and a reverse complement profile.
- We can calculate the balance between these sub-profiles.

k-mer profiles

Strand balance

How to calculate it:

- We can split a profile into a forward and a reverse complement profile.
- We can calculate the balance between these sub-profiles.

This is an estimation of "sufficient coverage":

- If the balance is good, the coverage is good.
  - Does not work for strand specific protocols.

k-mer profiles

Strand balance

How to calculate it:

- We can split a profile into a forward and a reverse complement profile.
- We can calculate the balance between these sub-profiles.

This is an estimation of "sufficient coverage":

- If the balance is good, the coverage is good.
  - Does not work for strand specific protocols.

We can balance the profile by adding k-mer counts to their reverse complement and vice versa.

k-mer profiles

Shrinking

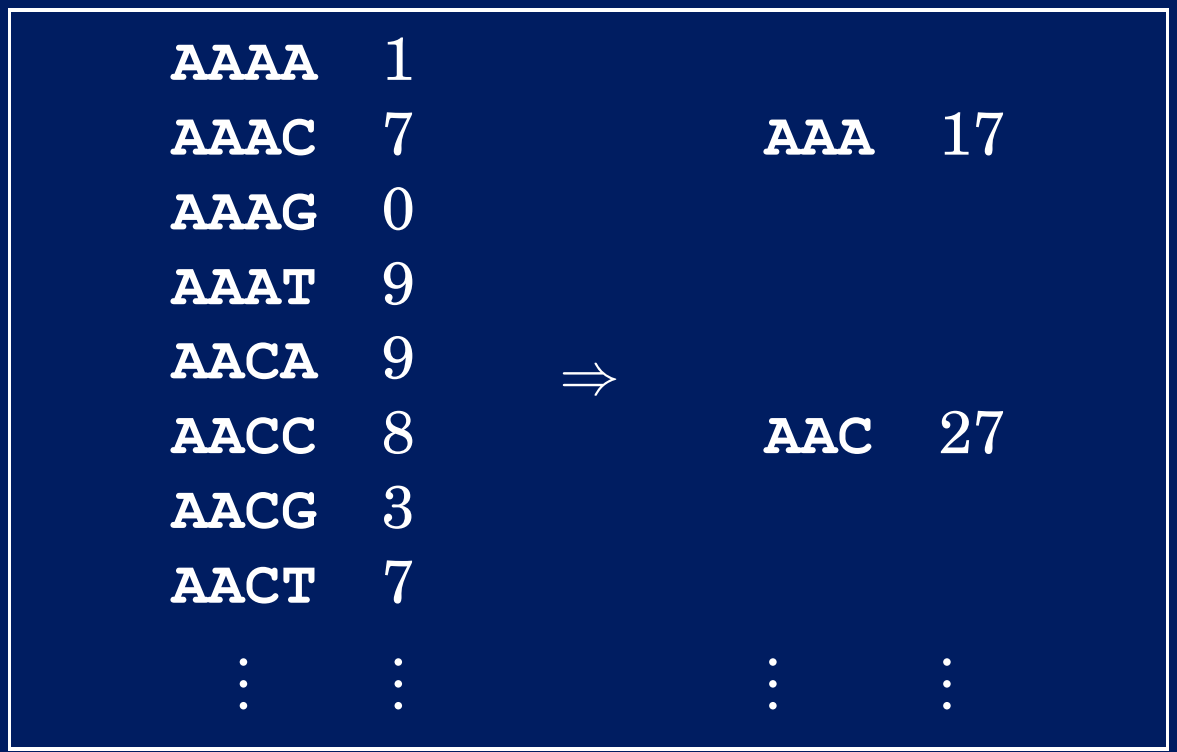


Figure 5: Shrinking a profile.

k-mer profiles

Shrinking

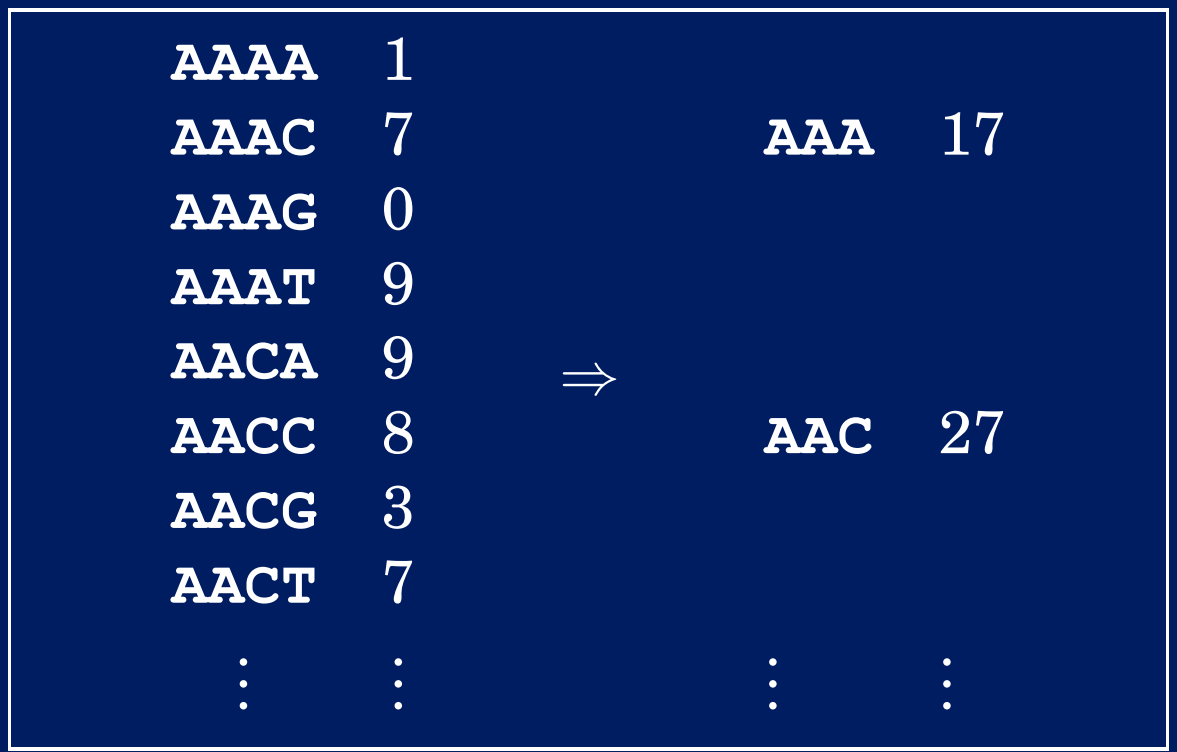


Figure 5: Shrinking a profile.

Works fine if the indexed sequences are large compared to k.

k-mer profiles

Smoothing

How do we compare multiple k-mer profiles?

One sample might be sequenced deeper than the other, so the optimal k might differ between comparisons.

k-mer profiles

Smoothing

How do we compare multiple k-mer profiles?

One sample might be sequenced deeper than the other, so the optimal k might differ between comparisons.

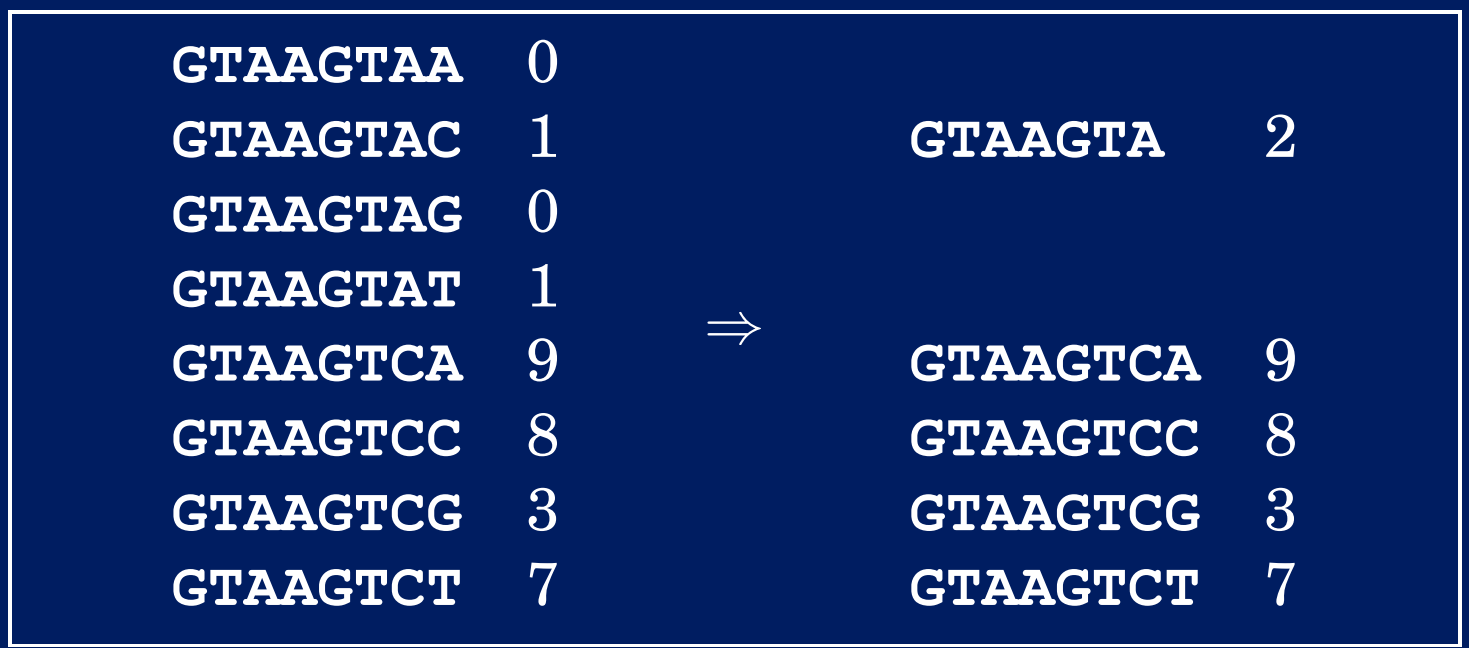


Figure 6: Smoothing by collapsing sub-profiles.

k-mer profiles

Smoothing

The function to determine when to smooth is a parameter:

- Median.
- Minimum.
- Average.
- ...

This function has a threshold, which is also a parameter.

k-mer profiles

Smoothing

The function to determine when to smooth is a parameter:

- Median.
- Minimum.
- Average.
- ...

This function has a threshold, which is also a parameter.

We can index with a large k-mer size, this method automatically uses the optimal size when comparing.

Implementation: command line interface

A programming library and command line interface

Uses a binary encoding of a k-mer as index:

- Only counts need to be stored.

https://humgenprojects.lumc.nl/svn/k-mer/

Implementation: command line interface

A programming library and command line interface

Uses a binary encoding of a k-mer as index:

- Only counts need to be stored.

Stored profiles can be:

- Loaded.
- Saved.
- Manipulated.
- Compared.
- ...

https://humgenprojects.lumc.nl/svn/k-mer/

Implementation: command line interface

Indexing and querying

Generic functions:

- index — Make a profile from a FASTA file.

Implementation: command line interface

Indexing and querying

Generic functions:

- index — Make a profile from a FASTA file.

Summaries, querying, etc.:

- info — Generic info (k-mer length, total/non-zero counts).
- meanstd — Mean and standard deviation of the count distribution.
- distr — Histogram of the count distribution.
- showbalance — Show the balance of a profile.
- getcount — Get the count for a specific k-mer.

Implementation: command line interface

Profile manipulation

The following manipulaiton functions are implemented:

- merge — Merge two profiles.
- balance — Balance a profile.
- positive — Only keep counts that are positive in both profiles.
- scale — Scale profiles such that the total number counts are equal.
- shrink — Calculate a smaller profile.
- smooth — Smooth two profiles by collapsing sub-profiles.
- shuffle — Permute a profile.

Implementation: command line interface

Comparison

For pairwise and multiple sample comparison:

- diff — Calculate the difference between two profiles.
- matrix — Make a distance matrix any number of profiles.

Implementation: command line interface

Comparison

For pairwise and multiple sample comparison:

- diff — Calculate the difference between two profiles.
- matrix — Make a distance matrix any number of profiles.

The resulting *distance matrix* can be visualised by a *phylogenetic tree*, or by doing *multidimensional scaling* (e.g., PCA).

Implementation: command line interface

Command line help

All commands are documented.

```
> python kMer.py diff -h
usage: kMer diff [-h] -i INPUT INPUT [-d] [-s SUMMARY] [-t THRESHOLD] [-b]
                 [-p] [-S] [-m] [-e] [-P PAIRWISE] [-n PRECISION]

Calculate the difference between two k-mer profiles.

optional arguments:
  -h, --help        show this help message and exit
  -i INPUT INPUT    pair of input files
  -d                scale down (default=False)
  -s SUMMARY        summary function for dynamic smoothing (int default=0)
  -t THRESHOLD      threshold for the summary function (int default=0)
  -b                balance the profiles (default=False)
  -p                use only positive values (default=False)
  -S                scale the profiles (default=False)
  -m                smooth the profiles (default=False)
  -e                use the euclidean distance metric (default=False)
  -P PAIRWISE       paiwise distance function for the multiset distance (int
                    default=0)
  -n PRECISION      number of decimals (int default=3)
```

Listing 1: Help for the diff option.

Programming library

Library for k-mer profiles

`kLib`:

- Analyse a fasta file.
- Loading and saving of profiles.
- Merging.
- Balancing / splitting.

Programming library

Library for k-mer profiles

`kLib`:

- Analyse a fasta file.
- Loading and saving of profiles.
- Merging.
- Balancing / splitting.

`kDiffLib`:

- Smoothing.
- Make a comparison recipe.

Programming library

Library for k-mer profiles

`kLib`:

- Analyse a fasta file.
- Loading and saving of profiles.
- Merging.
- Balancing / splitting.

`kDiffLib`:

- Smoothing.
- Make a comparison recipe.

`kMer`:

- Command line interface.

Programming library

A more general library

These functions are not limited to k-mer profiles.

`metrics`:

- Scaling.
- Extracting positive counts.
- Multiset distance function.
  - Pairwise distance functions.
- Euclidean distance function.
- Summary functions for the smoothing algorithm.

Programming library

API documentation

Figure 7: API documentation.

Questions?

Acknowledgements:

Yahya Anvar
Lusine Khachatryan
Irina Pulyakhina
Michiel van Galen
Jaap van der Heijden
Ken Kraaijeveld
Walter Kosters
Hendrik Jan Hoogeboom
Johan den Dunnen

https://humgenprojects.lumc.nl/svn/k-mer/