LEIDEN UNIVERSITY MEDICAL CENTER

k-mer programming library and toolkit

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics

Introduction

Profiling

We frequently want to know something about raw datasets.

Introduction

Profiling

We frequently want to know something about raw datasets.

Within a dataset (no or unknown reference):

- Quality control.
- Coverage estimation.
- Quality of a de novo assembly.

Introduction

Profiling

We frequently want to know something about raw datasets.

Within a dataset (no or unknown reference):

- Quality control.
- Coverage estimation.
- Quality of a de novo assembly.

Between 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.

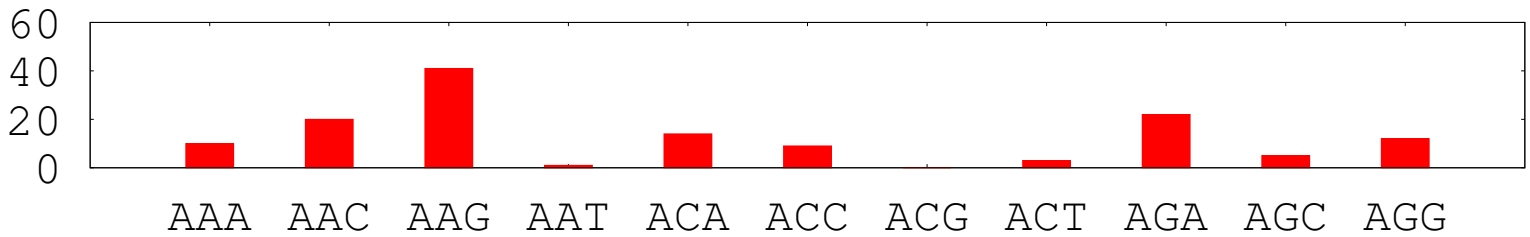

Figure 1: A profile of 3-mer counts.

In Figure 1 we see a part of 3-mer counts; **AAA** occurs 10 times, **AAC** occurs 20 times, etc.

Introduction

Why k-mers?

The usage of k-mers is appealing since:

- They are easy to work with.
- Fingerprinting of samples is possible (even for a small k-mer length (around 10)).
- Can be used to make a database to match against.
- Comparison between datasets takes little time compared to alignment.

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.
  - But, pattern growth needs also position information.
- 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.
  - But, pattern growth needs also position information.
- 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.
  - But, pattern growth needs also position information.
- 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.

This is solved by collapsing profiles.

kMer

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/

kMer

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/

kMer

Command line interface.

index	Make a k-mer profile from a FASTA file.
merge	Merge two k-mer profiles.
balance	Balance a k-mer profile.
showbalance	Show the balance of a k-mer profile.
positive	Only keep counts that are positive in both profiles.
scale	Scale profiles such that the total number of k-mers is equal.
smooth	Smooth two profiles by collapsing sub-profiles.
diff	Calculate the difference between two k-mer profiles.
matrix	Make a distance matrix any number of k-mer profiles.

kMer

Command line help.

All commands are documented.

```
> python kMer.py diff -h
usage: kMer.py 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.

kMer

Balancing.

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.

kMer

Balancing.

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.

Funcionality in **`kLib`**:

- We can split a profile into a forward and a reverse complement profile.
- We can calculate the balance between these sub-profiles.
- We can balance the profile by adding k-mer counts to their reverse complement and vice versa.

LUMC

kMer

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.

kMer

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.

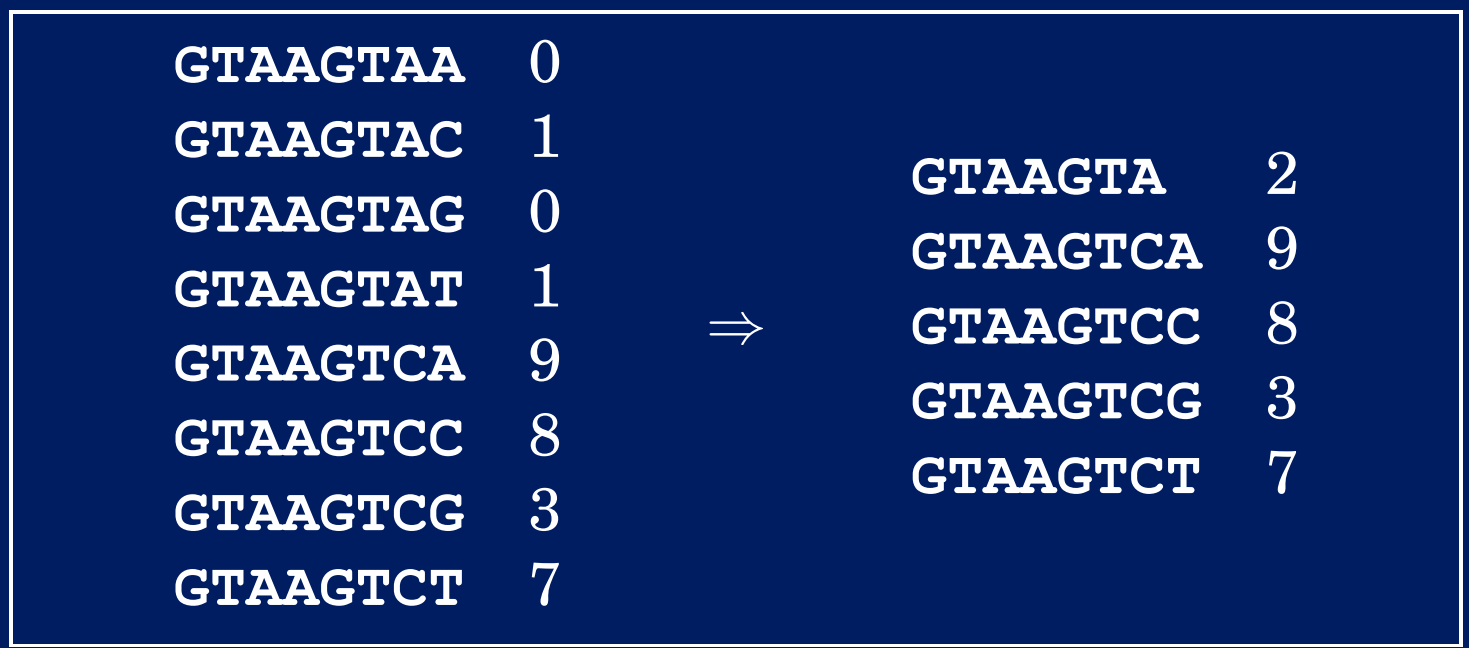

Figure 2: Smoothing by collapsing sub-profiles.

kMer

Smoothing.

The function to determine when to smooth is a parameter:

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

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

kMer

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.

LUMC

kLib

Library for k-mer profiles.

`kLib`:

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

kLib

Library for k-mer profiles.

`kLib`:

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

`kDiffLib`:

- Smoothing.
- Make a comparison recipe.

kLib

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.

LUMC

kLib

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.

kLib

API documentation.

Questions?

Acknowledgements:

Yahya Anvar

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