Sequence Classification

A Tutorial in Genetic Sequence Classification Tools and Techniques

Jake Drew
Data Mining CSE 8331
Southern Methodist University
jakemdrew@gmail.com
www.jakemdrew.com

SMU | Bobby B. Lyle School of Engineering

Sequence Characters

A - Adenine

C - Cytosine

G - Guanine

T - Thymine

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	any base
. or -	gap

http://www.bioinformatics.org/sms/iupac.html

Sequence File Formats

Plain sequence format

A sequence in plain format may contain only IUPAC characters and spaces (no numbers!).

Note: A file in plain sequence format may only contain **one** sequence, while most other formats accept several sequences in one file.

An example sequence in plain format is:

```
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

FASTA format

A sequence file in FASTA format can contain several sequences. Each sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line must begin with a greater-than (">") symbol in the first column.

An example sequence in FASTA format is:

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for prepro cortistatin like
peptide, complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```


http://www.genomatix.de/online_help/help/sequence_formats.html

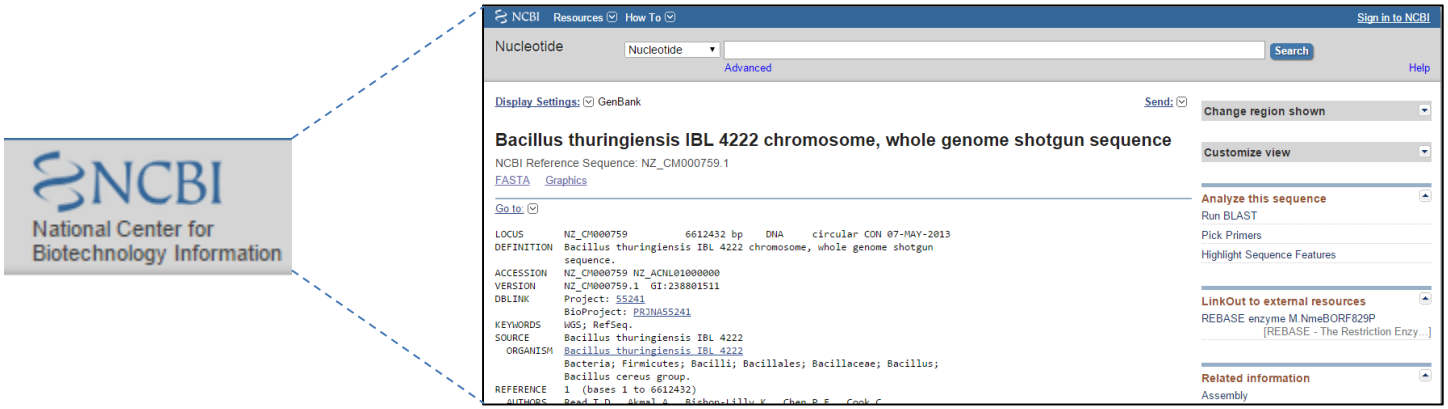
Sequence File Formats

GenBank format

A sequence file in GenBank format can contain several sequences. One sequence in GenBank format starts with a line containing the word LOCUS and a number of annotation lines. The start of the sequence is marked by a line containing "ORIGIN" and the end of the sequence is marked by two slashes ("//").

An example sequence in GenBank format is:

```
LOCUS AB000263 368 bp mRNA linear PRI 05-FEB-1999
DEFINITION Homo sapiens mRNA for prepro cortistatin like peptide,
complete cds.
ACCESSION AB000263
ORIGIN
1   acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg
61  ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
121 caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc
181 aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
241 gcgcacccca ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga
301 agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca
361 gacctgaa //
```



http://www.ncbi.nlm.nih.gov/nuccore/NZ_CM000759.1

SAM / BAM File Formats

SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments. SAM aims to be a format that:

- Is flexible enough to store all the alignment information generated by various alignment programs;
- Is simple enough to be easily generated by alignment programs or converted from existing alignment formats;
- Is compact in file size;
- Allows most of operations on the alignment to work on a stream without loading the whole alignment into memory;
- Allows the file to be indexed by genomic position to efficiently retrieve all reads aligning to a locus.

SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format.

BAM is the Binary Equivalent Format for SAM

Sequence Tokenization

Typically, sequences are compared for similarity using words of length k:

Token 1	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 2	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 3	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 4	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 5	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 6	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA
Token 47	TTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCCTAATGCATGCAAGTCGA

Simple Sequence Comparison

Two sequences S and T can be compared for similarity using Jaccard Similarity:

	Sequence S	Sequence T
1	ACCTGTAA	TTGGCCAA
2	GGGTAACA	CACACACA
3	AACCGGTT	ACCTGTAA
4	ACACACAC	ATGATGTG
5	GTGTAGTA	GGGTAACA
6	TTGGTGAG	GAGTGGTT

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|} = \frac{2}{10} = .2 = 20\%$$

This could be painfully slow when dealing with millions of sequence words.

Feature Space Size

The word length k chosen during tokenization can have impacts on classification performance and accuracy:

- When $k = 3$, there are $4^3 = 64$ possible unique words.
- When $k = 8$, there are $4^8 = 65{,}536$ possible unique words.
- When $k = 31$, there are $4^{31} = 4.61\ x\ 10^{18}$ possible unique words.
- When $k = 60$, there are $4^{60} = 1.33\ x\ 10^{36}$ possible unique words.

When $k = 3$, the features of two sequences can be represented in a 64 x 2 matrix:

	Sequence 1	Sequence 2
AAA	1	1
AAC	0	1
AAT	0	1
AAG	1	0
ACA	1	0
ACC	0	1
ACG	0	0
ACT	1	0
...	...	...

- Very small matrix
- All words fit in memory of any machine
- Performance optimal
- Poor accuracy as many sequences will share words.
- Complicated statistics must be used to calculate similarity.

Feature Space Size

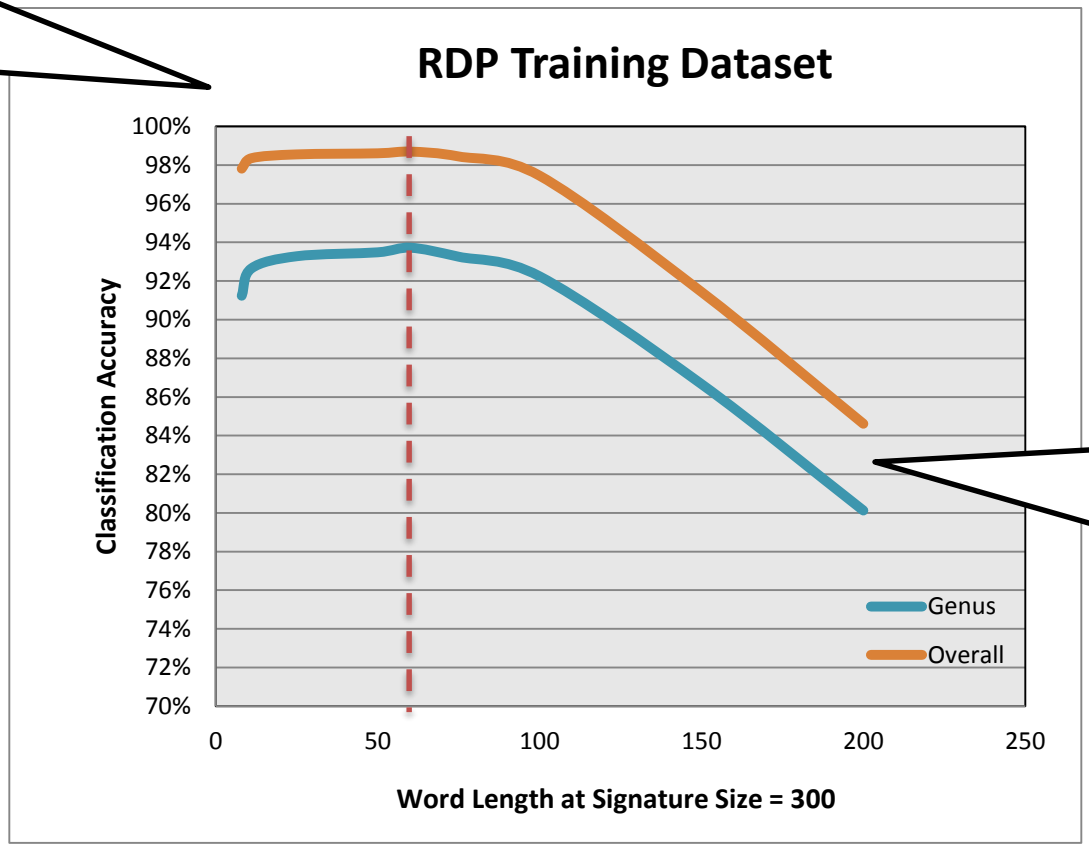
When $k = 60$, the features of two sequences can be represented in a matrix containing $1.33\ x\ 10^{36}$ possible rows.

- Performance challenges due to very large matrix.
- Matrix typically must be compressed using Locality Sensitive Hashing, random sampling, or some other technique.
- Highly accurate when collisions occur since the random probability of such a long match is very small.
- It is possible to use highly performance optimal Boolean similarity calculations since collisions are such strong indicators of similarity.
- Unique words are not included in the matrix unless they are identified within a sequence during training.

Optimal Word Length

Strand

Accuracy impacted by small feature space.



Accuracy impacted by lack of collisions.

Strand uses a word length of 60 characters with 4^{60} possible unique word values.

Sequence Alignment

"In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences."

Sequence 1

GGGGGCTAGCGTTATTCGGAATTACTG**G**GCGTAAAGCGCAC
GTAGGCGGA**T**CGGAAAGTCAGAGGTGAAATCCCAGGGCT

Sequence 2

GGGGGCTAGCGTTATTCGGAATTACTG**T**GCGTAAAGCGCAC
GTAGGCGGA**A**CGGAAAGTCAGAGGTGAAATCCCAGGGCT

NCBI BLAST

Sequence ID: lcl|60427 Length: 80 Number of Matches: 1

Range 1: 1 to 80 Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Identities	Gaps	Strand
137 bits(74)	2e-38	78/80(98%)	0/80(0%)	Plus/Plus

```
Query  1   GGGGGCTAGCGTTATTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGATCGGAAAGTC  60
           ||||||||||||||||||||||||||| |||||||||||||||||||| |||||||||
Sbjct  1   GGGGGCTAGCGTTATTCGGAATTACTGTGCGTAAAGCGCACGTAGGCGGAACGGAAAGTC  60

Query  61  AGAGGTGAAATCCCAGGGCT  80
           ||||||||||||||||||||
Sbjct  61  AGAGGTGAAATCCCAGGGCT  80
```

http://blast.ncbi.nlm.nih.gov/

Max score	Total score	Query cover	E value	Ident	Accession
137	137	100%	2e-38	98%	60427

http://en.wikipedia.org/wiki/Sequence_alignment

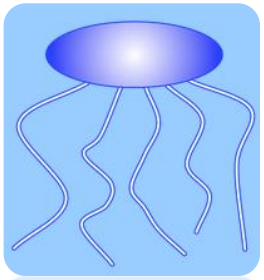
Sequence Alignment Tools

The following are some of the tools available online from NCBI:

- **BLAST** – Basic Local Alignment Search Tool.
- **Megablast** – Optimized for highly similar sequences.
- **BLASTN** - Search nucleotide subjects using a nucleotide query.
- **BLASTP** - Search protein databases using a protein query.
- **BLASTX** - search protein databases using a translated nucleotide query.
- **TBLASTN** - search translated nucleotide databases using a protein query.
- **TBLASTX** - search translated nucleotide databases using a translated nucleotide query.

Sequence alignment is very accurate but slow compared to alignment-free methods.

JELLYFISH Parallel K-mer Counter



JELLYFISH is a tool for fast, memory-efficient counting of k-mers in DNA.

- Tokenizes and counts words at a length specified by the user.
- Reads FASTA and multi-FASTA files containing DNA sequences.
- Outputs its k-mer counts in an binary format.
- Offers the "jellyfish dump" command for human readable outputs.
- Used by Kraken to tokenize and count sequence words from input data.
- Open source and freely available.
- Quickly implement custom classifiers without implementing tokenization and counting.

http://www.cbcb.umd.edu/software/jellyfish/

Alignment-Free Classifiers

Alignment-free use sequence tokenization and k-mer / word counting to estimate similarity between sequences.

The following are state of the art alignment-free classifiers :

- **RDP** - Naive Bayesian Classifier
- **Kraken** - Abundance Estimation Classifier
- **Strand** - MapReduce Style Classifier using Locality Sensitive Hashing

http://blast.ncbi.nlm.nih.gov/Blast.cgi

RDP

The Ribosomal Database Project (RDP) Classifier, a naıve Bayesian classifier, can rapidly and accurately classify bacterial 16S rRNA sequences:

- **Naıve Bayesian Classifier**
- **Predicts the taxonomy of gene sequences**
- **Uses a word length of 8 ($k = 8$)**
- **Average RDP training dataset record length = 1500 bases**
- **Tested shorter word lengths of 6 and 7 bases with low accuracy results**
- **RDP keeps a matrix of its training data with frequencies of the words it encounters by each training class**
- **RDP uses sampling to further reduce its feature space**
- **Due to sampling and a shorter word length RDP uses 100 bootstrap trials to make a class prediction (hurts performance)**
- **Must use more complex statistical methods to make predictions:**

P-value Estimate

$$Z = \frac{\frac{x}{N_1} - \frac{y}{N_2}}{\sqrt{\mu(1 - \mu)\left(\frac{1}{N_1} + \frac{1}{N_2}\right)}}$$

Membership Estimate

$$P(y|x) = \left(\frac{N_2}{N_1}\right)^y \frac{(x+y)!}{x!y!\left(1 + \frac{N_2}{N_1}\right)^{(x+y+1)}}$$

http://sourceforge.net/projects/rdp-classifier/

BioTools R

This package aims at using R and the Biostrings package as the common interface for several important tools for multiple sequence alignment as database driven sequence management for 16S rRNA:

- **Use R to execute**
  - **RDP**
  - **Blast**
  - **Clustalw**
  - **kalign**
- **Available using Library(BioTools) package in R**
- **Must install RDP and BLAST executables to use**

BioTools R Example

The following example shows using BioTools and RDP to classify sequences in R:

Reference Package

```
library(BioTools)

#Set working directory
setwd("D:/Strand/Benchmarks/RDP/TenFold")

foldsPath <- "D:/Strand/Benchmarks/RDP/RDPFolds/"
foldSummaryAll <- NULL

for (i in 1:10) {
    classifyPath <- paste(foldsPath, "foldClassify",i,".fasta", sep='')
    learnPath<- paste(foldsPath, "foldLearn",i,".fasta", sep='')

    #learning ...
    # Start the clock!
    ptm <- proc.time()
    learnFold<-readDNAStringSet(learnPath)
    foldRDP <- trainRDP(learnFold)
    # Stop the clock!
    learnTime<-proc.time() - ptm

    #classification...
    ptm <- proc.time()
    classifyFold<-readDNAStringSet(classifyPath)
    classifyPredict<-predict(foldRDP, classifyFold)
    classifyTime<-proc.time() - ptm
```

Read Data

Train Classifier

Read Data

Make Predictions

Kraken Classifier

Kraken is an ultrafast and highly accurate program for assigning taxonomic labels to metagenomic DNA sequences:

- **Kraken is an "Abundance Estimation" classifier**
- **Classifies sequence reads between 92-156 bases**
- **The goal is very rapid "shotgun classification"**
- **Uses a word length of 31 ($k = 31$)**
- **The full Kraken database is in memory and requires 70GB**
- **High precision (classification accuracy) >95%**
- **Low sensitivity (the ability to make a prediction)**
- **Only classified between 72% to 91% of its test files**
- **Saves words during training with the lowest common ancestry**
- **Classification speeds are measured in "reads per minute" (rpm)**
- **Rpm values vary between test datasets from 890,000 to 1.5 million**
- **Kraken is open source and available for download from the Kraken website.**

http://ccb.jhu.edu/software/kraken/

Kraken Classifier

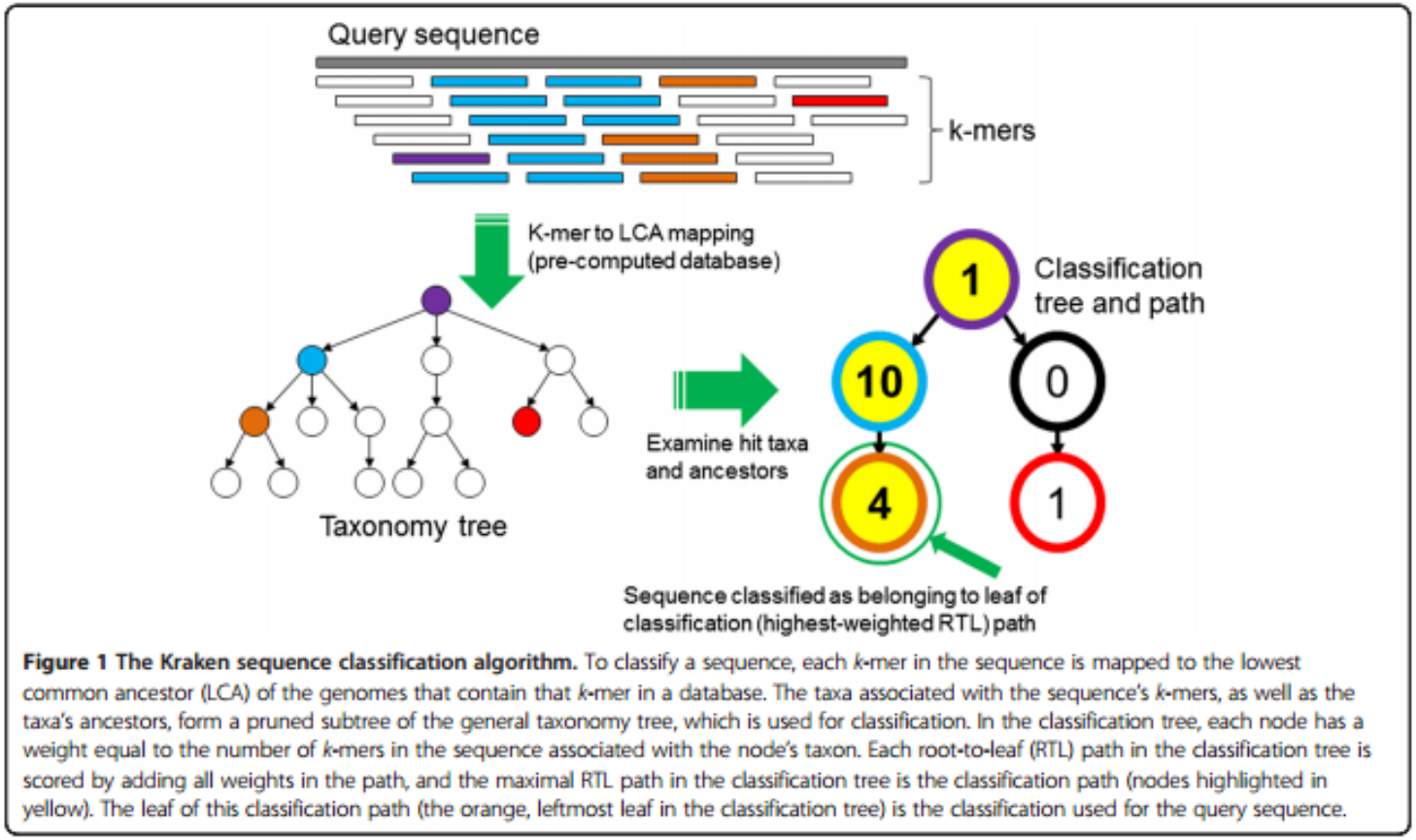


Figure 1 The Kraken sequence classification algorithm. To classify a sequence, each *k*-mer in the sequence is mapped to the lowest common ancestor (LCA) of the genomes that contain that *k*-mer in a database. The taxa associated with the sequence's *k*-mers, as well as the taxa's ancestors, form a pruned subtree of the general taxonomy tree, which is used for classification. In the classification tree, each node has a weight equal to the number of *k*-mers in the sequence associated with the node's taxon. Each root-to-leaf (RTL) path in the classification tree is scored by adding all weights in the path, and the maximal RTL path in the classification tree is the classification path (nodes highlighted in yellow). The leaf of this classification path (the orange, leftmost leaf in the classification tree) is the classification used for the query sequence.

http://www.biomedcentral.com/content/pdf/gb-2014-15-3-r46.pdf

Kraken Classifier

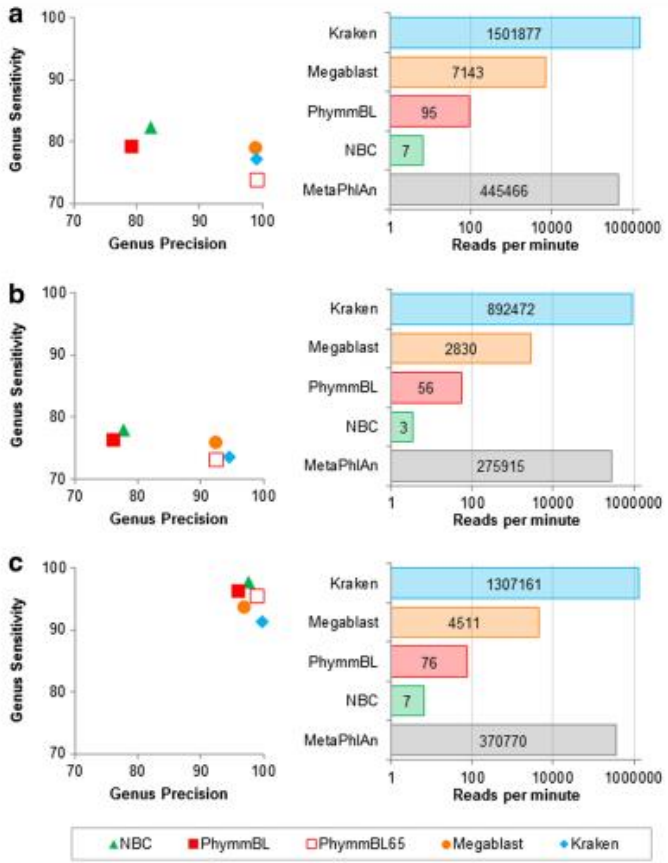


Figure 2 Classification accuracy and speed comparison of classification programs for three simulated metagenomes. For each metagenome, genus precision and sensitivity are shown for five classifiers, and speed is shown for five programs (PhymmBL65 is simply a confidence-filtered version of PhymmBL's results, and MetaPhlAn only classifies a subset of reads that map to one of its marker genes, as it is an abundance estimation program). Results shown are for: **(a)** the HiSeq metagenome, consisting of HiSeq reads (mean length $\mu = 92$ bp) in equal proportion from ten bacterial sequencing projects; **(b)** the MiSeq metagenome, consisting of MiSeq reads ($\mu = 156$ bp) in equal proportion from ten bacterial projects; and **(c)** the simBA-5 metagenome, consisting of simulated 100-bp reads with a high error rate from 1,967 bacterial and archaeal taxa. Note that the horizontal axes in all speed graphs have a logarithmic scale.

http://www.biomedcentral.com/content/pdf/gb-2014-15-3-r46.pdf

Strand Classifier

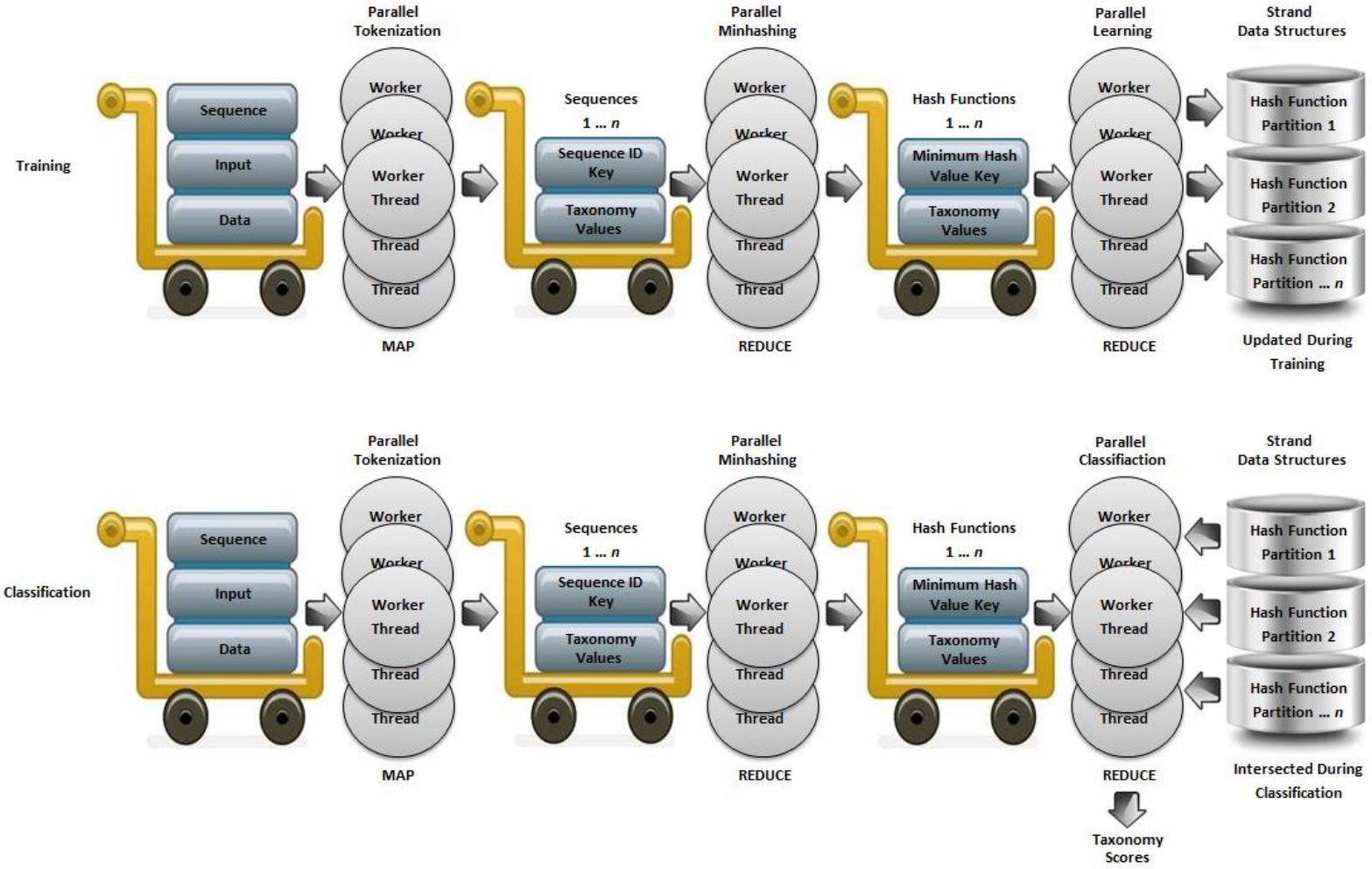
Strand

Performs Fast Sequence Comparison using MapReduce and Locality Sensitive Hashing:

- **General purpose classifier**
- **Uses a MapReduce style pipeline for highly parallel processing performance**
- **Uses Locality Sensitive Hashing to compress data by > 98%**
- **20 times faster than RDP**
- **Uses only around 4GB to store the same training data used in the Kraken database.**
- **More accurate than RDP.**
- **Comparable accuracy when compared to Kraken.**

http://www.biomedcentral.com/content/pdf/gb-2014-15-3-r46.pdf

Strand Multicore MapReduce



MinHash Signatures

Strand uses n randomly seeded hash functions to select n minimum hash values which represent a random permutation of all words in a sequence or taxonomy:

Strand Intersects n MinHash values between sequences and taxonomies to provide an accurate approximation of the Jaccard Index.

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|} \approx \frac{2}{6} \approx .33$$

Questions?

Additional References

1. **BLAST -** Altschul, Stephen F., et al. "Basic local alignment search tool." Journal of molecular biology 215.3 (1990): 403-410.
2. **JELLYFISH** - Guillaume Marcais and Carl Kingsford, A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics (2011) 27(6): 764-770.
3. **RDP** – Wang, Qiong, et al. "Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy." Applied and environmental microbiology 73.16 (2007): 5261-5267.
4. **Kraken** – Wood, Derrick E., and Steven L. Salzberg. "Kraken: ultrafast metagenomic sequence classification using exact alignments." Genome Biol 15.3 (2014): R46.
5. **Strand** - Drew, Jake, and Michael Hahsler. "Strand: fast sequence comparison using mapreduce and locality sensitive hashing." Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. ACM, 2014.
6. **R BioTools** - Hahsler, Michael, and Anurag Nagar. "BioTools: Tools based on Biostrings (alignment, classification, database)." (2013): 05-01.

Strand

Thank You!

Jake Drew

jakemdrew@gmail.com

www.jakemdrew.com