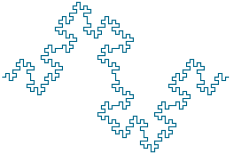
Classification of metagenomic data

Shuya Chu, Stephen Cristiano, Bing He, Zhou Ye
Johns Hopkins University



December 2, 2013

METAGENOMICS

"The application of modern genomics techniques to the study of communities of microbial organisms directly in their natural environments, bypassing the need for isolation and lab cultivation of individual species". - Lior Patcher and Kevin Chen, 2005

BIOLOGICAL MOTIVATION

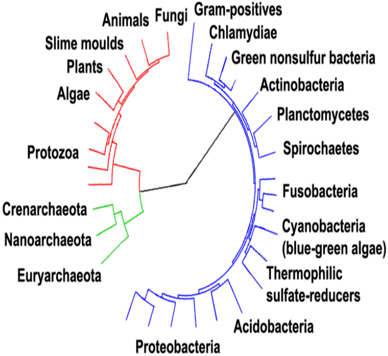
- Potential impact on public health (the human microbiome project), agriculture, biofuels, and other fields.
- If we take an environmental sample (soil, gut, . . .) and sequence the contained DNA, we will generate reads from many microbial organisms.
- We wish to classify each read to its correct phylogenic origin and gain some insight to the abundance of each organism within the sample.
- Complicated by the hierarchical structure of the classifications and since in a metagenomic sample, many reads will come from *"de novo"* genomes.

EXISTING METHODS

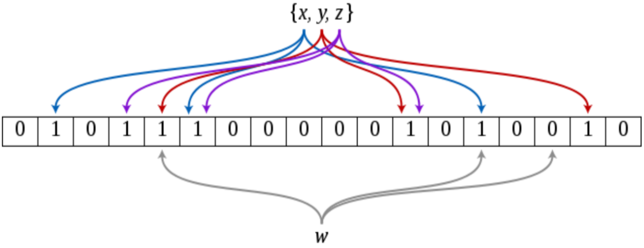
- PhymmBL
  - Arthur Brady and Steven Salzberg
  - Uses Interpolated Markov Model together with BLAST to classify DNA taxonomically classify metagenomic reads.
  - Very accurate, but local alignment with BLAST is very slow!
- FACS
  - Henrik Stranneheim and Joakim Lundeberg
  - Bloom Filter approach - very space-efficient.
  - Has performance issues and primary purpose is identifying DNA contamination in metagenome sample.
- Kraken
  - Derick Wood and Steven Salzberg
  - Novel classification algorithm utilizing exact alignment of k-mers.
  - Achieves very high speed in exchange for a slight sacrifice of sensitivity.
- MetaPhlAn, MEGAN, metaphyler, . . .

DATA

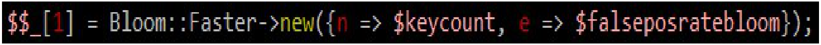
- RefSeq for whole genomes of bacteria (obtained from NCBI)
- Due to vastness of RefSeq, only considered proteobacterium phylum.
- Generated simulated FASTA files from Metasim and some of our own scripts.



Bloom Filter



Recall: a Bloom filter is a sketch data structure that is extremely compact, but fails sometimes.
We use the PERL module Bloom::Faster to build bloom filters from our reference genomes.



```
$$_[1] = Bloom::Faster->new({n => $keycount, e => $falseposratebloom});
```

Bloom Filter

General Process

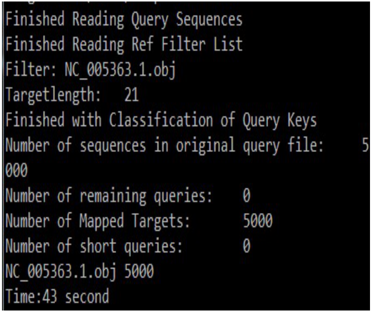
- **First**, split query seq into K-mers (with the same length we used to build the filter). **Second**, query K-mers against each Bloom Filter for each Ref Genome.
- **Coarse Scan**: Query Bloom Filter with Non-overlapping K-mers.
- **Detailed Scan**: If query seq passes the Coarse Scan, we query Bloom Filter all (overlapping) K-mers.

Scoring

- Match score proportional to matched bp in each query seq.
- **Species level**: combine max and threshold for match score.
- **Higher level**: if species-level classification cannot be made, use mutual information criteria for classifying at genus (or higher) level.
- Incorporate false positive rate into match score?

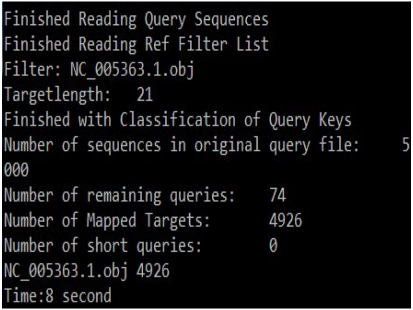
PRELIMINARY RESULTS

Query simulated data with Exact model:
Takes 43 seconds



```
Finished Reading Query Sequences
Finished Reading Ref Filter List
Filter: NC_005363.1.obj
Targetlength:   21
Finished with Classification of Query Keys
Number of sequences in original query file:     5
000
Number of remaining queries:    0
Number of Mapped Targets:       5000
Number of short queries:        0
NC_005363.1.obj 5000
Time:43 second
```

Query simulated data with 454 error model:
Takes 8 seconds



```
Finished Reading Query Sequences
Finished Reading Ref Filter List
Filter: NC_005363.1.obj
Targetlength:   21
Finished with Classification of Query Keys
Number of sequences in original query file:     5
000
Number of remaining queries:    74
Number of Mapped Targets:       4926
Number of short queries:        0
NC_005363.1.obj 4926
Time:8 second
```

INTERPOLATED MARKOV MODEL

For modeling DNA, we need $O(4^{n+1})$ parameters for an n-th order Markov model.

Interpolated Markov model combines the merits of both low and high order Markov model.

Basic Model:

$$\begin{aligned}&P_{IMM}(x_i|x_{i-1},\cdots,x_{i-n})\\&=\lambda_0 P(x_i)+\lambda_1 P(x_i|x_{i-1})+\cdots+\lambda_n P(x_i|x_{i-1},\cdots,x_{i-n})\end{aligned}$$

where $\sum_i \lambda_i = 1$

Improved Model (Use History):

$$\begin{aligned}&P_{IMM,n}(x_i|x_{i-1},\cdots,x_{i-n})\\&=\lambda_n(x_i)P(x_i|x_{i-1},\cdots,x_{i-n})\\&+(1-\lambda_n(x_i))P_{IMM,n-1}(x_i|x_{i-1},\cdots,x_{i-n+1})\end{aligned}$$

PARAMETER LEARNING

We learn the probability $P(x_i|x_{i-1},\cdots,x_{i-n})$ by counting.

$$P(x_i|x_{i-1},\cdots,x_{i-n}) = \frac{f(x_i,x_{i-1},\cdots,x_{i-n})}{f(x_{i-1},\cdots,x_{i-n})}$$

where f is the number of occurrences of input string
λ is determined by a χ^2 hypothesis test. If there are sufficient number of strings c to compute $P(x_i|x_{i-1},\cdots,x_{i-n})$, we set $\lambda_n(x_i)$ as 1; otherwise, we compare the distribution of n-th order history and n-1-th by χ^2 hypothesis test. After doing this test, we will obtain a p-value d. We determine $\lambda_n(x_i)$ by:

$$\lambda_n(x_i) = \begin{cases} 1.0 & \text{if } c > \mathit{threshold} \\ \frac{c}{400}\times d & \text{if } d \geq 0.05, c \leq \mathit{threshold} \\ 0.0 & \text{if } d < 0.05, c \leq \mathit{threshold} \end{cases}$$

The threshold is determined by trial and error.

IMPROVEMENT WITH BOWTIE

What is Bowtie?

- Bowtie is a sequence aligner wrote by professor Langmead.
- It indexes the genome with an FM Index to align short reads.
- For each alignment, Bowtie also scores them

Idea for Improvement:

- Using score for previous alignment, skip some bad alignment
- Details: If the sample data get a 'bad' score when aligning to a species, then it definitely will have bad scores for the species in the same genus.

Improvement with Bowtie

Set the Threshold:

- Give more penalties for mismatch(gap)
- Use the average score

Ongoing work:

- Still thinking about how to get the best threshold
- Comparison and analysis between this and bowtie

Ongoing work

- Benchmark all the methods against each other on the same machine and input.
- Memory footprint of bloom filter vs Markov Model.
- Instead of absolute classification, return a confidence score instead.
- Investigate how well a method deals with *de novo* genomes.

THANKS

- Ben Langmead
- The developers of the open source software which aided us in this project.
- Hongkai Ji and Kasper Hansen