Prediction of Lab Origins From Feature Analysis of Sequencing Read Data

Bowen Xue/ Andy Lin/ Fan Zhang

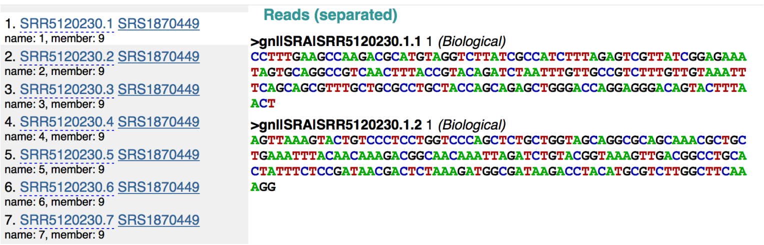
Background

For forensic investigation, it’s important to determine where a DNA sample was sequenced.Datasets from different sequencing institutions may have informative features which makes it possible to use machine learning to do the classification. Our project is working on the classification from features analysis of the reads data to determine which lab origin it comes from.

Data

We will focus on institutions that have sequenced E.coli using Illumina MiSeq. There are 359 paired-end sequencing runs downloaded from the National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA). 174 of the 359 datasets were submitted to the SRA by the Centers for Disease Control and Prevention Enteric Diseases Laboratory Branch. The remaining datasets were submitted by the Statens Serum Institut in Denmark.

The SRA Toolkit was used to download SRA files and convert them into FASTQ files. Custom Python scripts were used to do batch preprocessing. Below is what the raw data looks like.



1. SRR5120230.1 SRS1870449
name: 1, member: 9
2. SRR5120230.2 SRS1870449
name: 2, member: 9
3. SRR5120230.3 SRS1870449
name: 3, member: 9
4. SRR5120230.4 SRS1870449
name: 4, member: 9
5. SRR5120230.5 SRS1870449
name: 5, member: 9
6. SRR5120230.6 SRS1870449
name: 6, member: 9
7. SRR5120230.7 SRS1870449
name: 7, member: 9

Reads (separated)

```
>gnl|SRA|SRR5120230.1.1 1 (Biological)
CCTTTGAAGCCAAGACGCATGTAGGTCTTATCGCCATCTTTAGAGTCGTTATCGGAGAAA
TAGTGCAGGCCGTCAACTTTACCGTACAGATCTAATTTGTTGCCGTCTTTGTTGTAAATT
TCAGCAGCGTTTGCTGCGCCTGCTACCAGCAGAGCTGGGACCAGGAGGGACAGTACTTTA
ACT
>gnl|SRA|SRR5120230.1.2 1 (Biological)
AGTTAAAGTACTGTCCCTCCTGGTCCCAGCTCTGCTGGTAGCAGGCGCAGCAAACGCTGC
TGAAATTTACAACAAAGACGGCAACAAATTAGATCTGTACGGTAAAGTTGACGGCCTGCA
CTATTTCTCCGATAACGACTCTAAAGATGGCGATAAGACCTACATGCGTCTTGGCTTCAA
AGG
```

Methodology

The methodology was designed with two parts: supervised learning including logistic regression and SVM method and unsupervised learning using kmean. 60% train, 20% test, 20% validation.

For unsupervised learning, since the dimension of the features can be up to several hundreds, the first intuition is to do dimension reduction using two methods first: T-SNE (t-distributed stochastic embedding) and SIMLR (Single-cell Interpretation via Multi-kernel Learning) which uses multiple kernels to give the best estimate the dimension reduction result.

T-SNE: In this framework, each single point has a distribution of potential neighbors on all other points, defined as Pj|i which translates as probability that j is i’s neighbor, which means each data has its internal view about all other points.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \quad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \quad q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}.$$

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

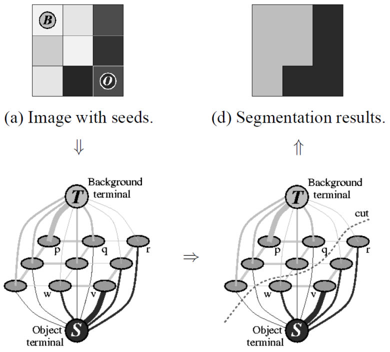
After some rough try with Kmean++ using these two dimension reduction methods, the accuracy was just 0.573 with T-SNE and 0.615 with SIMLR, which is quite unacceptable. We simplified this unsupervised part to “semi-supervised learning”. Inspired by the idea from image segmentation, the Graphcut Algorithm does the following cost minimization:

$$E(A) = \lambda \cdot R(A) + B(A)$$

$$\begin{aligned} R(A) &= \sum_{p \in \mathcal{P}} R_p(A_p) \\ B(A) &= \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q) \end{aligned}$$

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

Where the R(A) represents the data term which try to fit the optimal model for the measurement; B(A) represents the smoothness term, which comes from some prior knowledge.   A is a vector with each element being a node, and the node is either an object or a background; P denotes all pixels in an image; R is a metrics which measures the fitness of a node Ap if we know the category of the node; B is a metrics which measures the similarity between nodes.



(a) Image with seeds.

(d) Segmentation results.

(b) Graph.

(c) Cut.

edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{“bkg”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{“obj”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

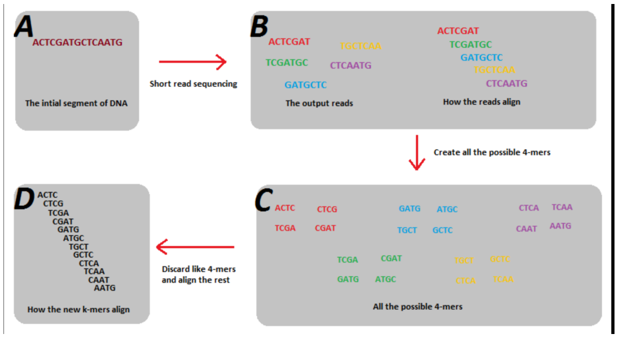
Since there is not a natural edge definition in the general graph, we define the neighbor edge, n-link, to be the neighbor probability in t-SNE section and data used here is high dimensional data. To reduce, the computation and noise, we only take highest 20 probable neighbor for each node. Hence the function B(p,q) has been defined. For data fitness part, Rp(), it is unreasonable to fit a histogram in high-dimension data due to curse of dimensionality and we don’t have enough data. Therefore R(p) is built from dimension reduced data, i.e. compute a histogram based on labelled object/ background data, and when encountered a new data, we fit the location info into that histogram and generate the scores.

Features

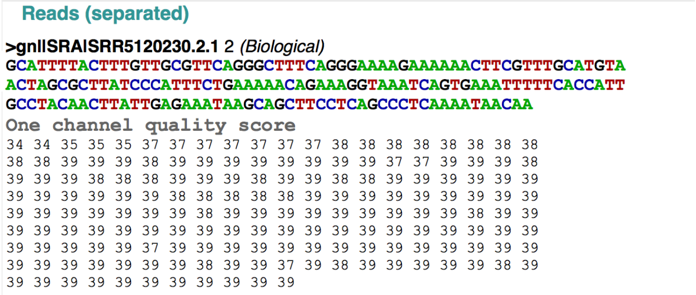
We tried three different feature types to infer the laboratory origin of sequencing runs:

The first set of features was the proportion of every 4-mer in the reads of a sequencing run. Prior to sequencing, genomic DNA must randomly sheared into smaller pieces. There are several types of methods for shearing DNA, which include physical, enzymatic, and chemical.

These biases may be ascertained by comparing the proportion of different 4-mers across sequencing runs. To calculate the proportion of 4-mers in a sequencing run, we slid a four DNA base size window down each read. At every position in the reads, we then would increment the corresponding 4-mer. Following normalization, this results in the proportion of each 4-mer in each sequencing run. There are either 256 or 625 features that result from this process, depending on whether 'N' is considered a unique nucleotide or not.



The second set of features we used is related to the quality of the sequencing reads. Depending on the protocol used and the skill of the technician processing the sample, there can be differences in the quality of the sequencing run. Each base in a sequencing read has an associated quality score with it. The distribution of quality scores and the drop in quality as a function of read length may differ between laboratories as a result of technician idiosyncrasies and protocol differences.



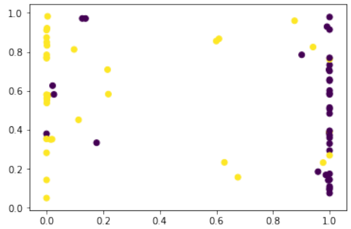
Reads (separated)

```
>gnl|SRA|SRR5120230.2.1 2 (Biological)
GCATTTTACTTTGTTGCGTTCAGGGCTTTCAGGGAAAAGAAAAAACTTCGTTTGCATGTA
ACTAGCGCTTATCCCATTTCTGAAAAACAGAAAGGTAAATCAGTGAAATTTTTCACCATT
GCCTACAACTTATTGAGAAATAAGCAGCTTCCTCAGCCCTCAAAATAACAA
One channel quality score
34 34 35 35 35 37 37 37 37 37 37 37 38 38 38 38 38 38 38 38
38 38 39 39 39 38 39 39 39 39 39 39 39 39 37 37 39 39 39 38
39 39 39 38 38 38 39 39 39 38 39 39 38 38 39 39 39 39 39 39
39 39 39 39 39 39 38 38 38 38 38 39 39 39 39 39 39 39 39 39
39 39 39 39 39 39 39 38 39 39 39 39 39 39 39 39 39 38 39 39
39 39 39 39 39 39 39 39 39 39 39 39 39 39 39 39 39 39 39 39
39 39 39 39 39 37 39 39 39 39 39 39 39 39 39 39 39 39 39 39
39 39 39 39 39 39 39 38 39 39 37 39 38 39 39 39 39 39 38 39
39 39 39 39 39 39 39 39 39 39 39
```

The third set of features used as input into the machine learning algorithms is the fragment size distribution of each sequencing run. Due to the various methods for fragmenting DNA prior to sequencing, it is possible that there is a signature each laboratory imparts on fragment length distribution. To determine whether fragment length distribution can differentiate between laboratory sources, we aligned paired-end reads together against a reference genome to determine fragment length. Each fragment length bin contains fragment lengths in a 200 base-pair window. The smallest bin size is -1,000 and the largest bin size is 1,000. Any fragment length that falls into a bin smaller than -1,000 is assigned to the -1,000th bin. Any fragment length that falls into a bin larger than 1,000 is assigned to the 1,000th bin.

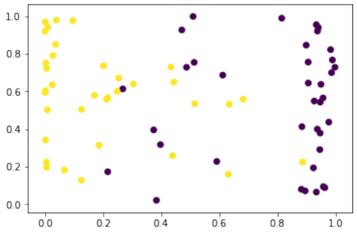
Project Results

1. logistic regression with single feature of quality score distribution:



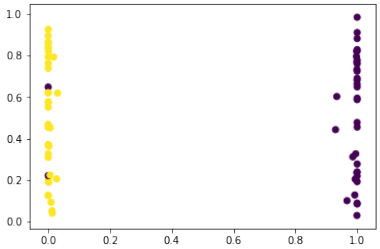
```
accuracy:  0.777777777778
precision:  0.756097560976
recall:  0.837837837838
```

2. logistic regression with 4-mer features



```
accuracy:  0.833333333333
precision:  0.857142857143
recall:  0.810810810811
```

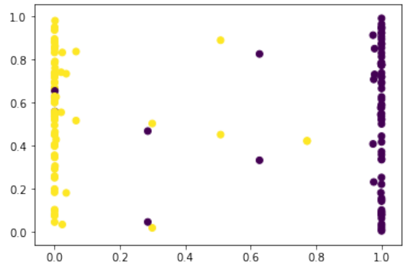
3. logistic regression with combined features：



```
accuracy:  0.972222222222
precision:  1.0
recall:  0.945945945946
```

Analysis：combined features accuracy is less than just 4-mer features which implies that the quality score distribution feature maybe “noise”. For semi-supervised learning, it may be more reasonable to just consider about 4-mer features.

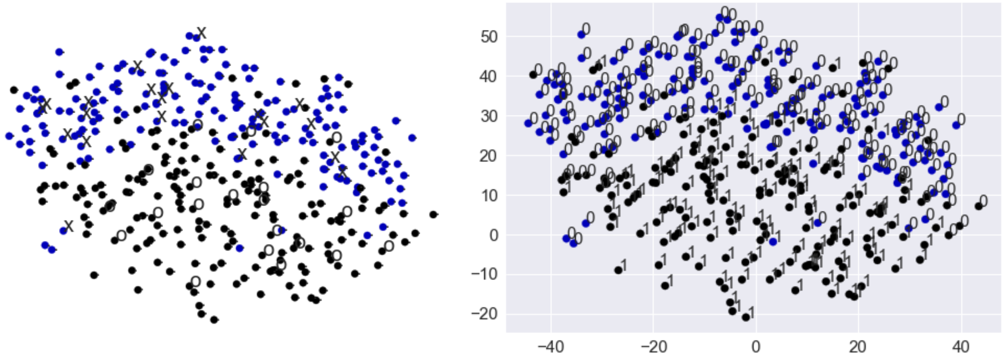
4. SVM with combined features:



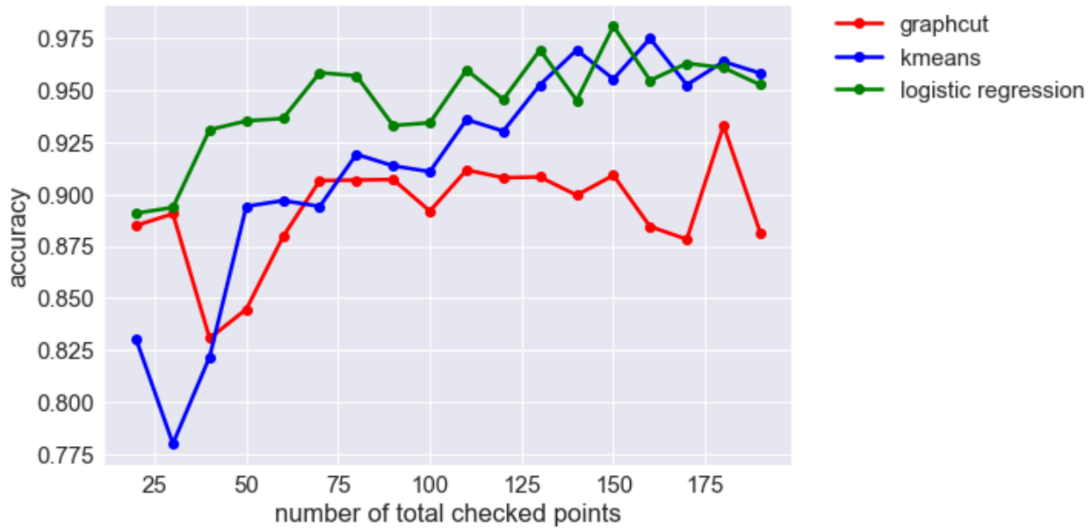
```
accuracy:  0.958333333333
precision:  0.972222222222
recall:  0.945945945946
```

SVM method turned out to have better performance than simply logistic regression with combined features which is reasonable by knowledge.

5. semi_supervised learning with 4-mer features:



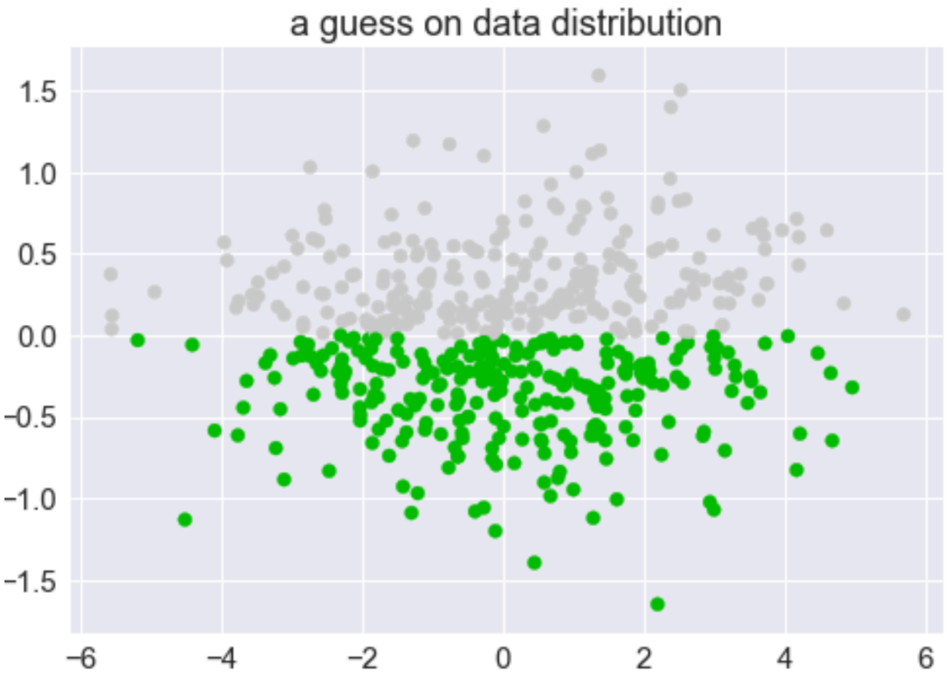
accuracy: 87.7%



Two semi-supervised learning algorithms comparison: it turned out the developed algorithm does not perform well, and the algorithm like dimension reduction does not improve the accuracy compared to simple supervised learning method.

Since logistic regression can make fairly accurate prediction based on a few points. Data in high dimension space is easily separable and there is only a few mixing there. On the other hand, in the unsupervised part, it turned out it is hard to cluster them without a few known points.

We also tried the ICA to try to cluster based on "non-gaussianity", but the result is also poor. The data might follow the similar distribution as the following graph, which data are easily classified once some points are known, but in general hard to cluster. Although in the following case it would be easy, by considering the second PCA component.



Future Work

Additional datasets;

Datasets from different experiments from the same laboratory;

Datasets with different organisms;

More features can be added to quality scores: Mean, Sd, Full distribution, Average decrease in quality as a function of base.

References

http://www.nature.com/nrg/journal/v11/n10/full/nrg2825.html
https://academic.oup.com/nar/article/42/21/e161/2903156/svaseq-removing-batch-effects-and-other-unwanted
https://www.nature.com/articles/srep39921
https://genomemedicine.biomedcentral.com/articles/10.1186/gm208
https://cancergenome.nih.gov/researchhighlights/tcgainaction/AkbaniBatchEffectsCaseStudy
https://bmcbiol.biomedcentral.com/articles/10.1186/s12915-014-0087-z