Biocomputational Puzzles

Dennis Shasha

Courant Institute of Mathematical Sciences

New York University

shasha@cs.nyu.edu

Close collaborators in biology: Gloria Coruzzi and Ken Birnbaum.

Close collaborator in DNA computing: Martyn Amos

What Does Biological Computing Mean?
Part I

- Algorithmist: "I think of algorithm. Biologists will implement. I will enjoy fame and glory." Paradigm for string and graph comparison algorithms. Combinatorial Pattern Matching conf; low ratio utilty/cleverness

- Computer architect/physicist: There is a foundational problem in biology that can be solved only through massive computation. Protein folding based on molecular dynamics (femtosecond n-body steps that must scale up to millisecond duration) as in David Shaw's machine. Protein folding based on machine learning (Rosetta).

What Does Biological Computing Mean? Part II

- Biologist: could some computer scientist help me analyze data through anova and tell me that my gene is important? This is bioinformatics. What database people have done: work on query languages for ordered data for tree and graph data (e.g. Jignesh Patel/Rosalba Giugno), provenance (Peter Buneman, Susan Davidson, Juliana Freire), visualization (Spotfire from Shneiderman/Sungear by Chris Poultney). Good and useful.

- Computer science/biologist: computer scientist helps identify promising experiments; develops new hypotheses, applies machine learning and active learning (Daphne Koller).

Power of Active Learning

1. Given a large search space (e.g. many different input levels of carbon, nitrogen, light, and hormones), propose a well-separated subset of those condition-values (using combinatorial design) as an experiment and then, depending on the results of the experiments, propose a new set. Goal is to find the conditions that are most important. Active learning and sampling with guarantees.

2. Biological data comes in two flavors: lots of "omic" scale data (species-wide) but of poor quality; a tiny bit of high quality data obtained by laborious techniques knockouts/overexpressors. Goal: use machine learning/data mining to predict precise results from low quality data. Can't always do this because trained machine learning model may have too low a precision based on cross-validation on the training data. Active learning in that case entails figuring out which other high quality data to ask for.

What Does Biological Computing Mean? Part III

- Computer scientist/engineer: Think about biology as a computational substrate and try to compute with it (Leonard Adelman, Richard Lipton, DNA computing conference).

This is what I propose to talk about.

DNA Hash Pooling and its Applications

Here I'd like to talk about a database-inspired algorithm that one could use to solve some important biological problems and to do so IN A WET LAB.

Joint work with:

Martyn Amos
Department of Computing and Mathematics
Manchester Metropolitan University
M.Amos@mmu.ac.uk

The Challenge

In ecological systems (human gut, lakes), want to answer questions such as

- (Genome Detection) Is some known gene sequence present at least in part in an environmental sample?
  Ex: is a pathogen present?

- (Similarity Discovery) How similar are two unsequenced samples in terms of sequence content?
  Motivation: try to find the sequences in the two samples that cause the same effect.

Techniques Available

- If you have intact cells, look for common "16S rRNA" to find common species.
  Difficulties: (i) you may not have intact cells; (ii) some cell types may be much more frequent than others; (iii) extraction and analysis requires effort.

- If you know beforehand which sequences may be of interest, you may use a microarray (prepared in advance) which can test for thousands of sequences simultaneously.
  Difficulty: you may not know in advance what to look for.

- Break the DNA you are given into small pieces and find the sequence of random pieces, hoping to find a match in a database.
  Difficulties: (i) may repeatedly sequence pieces from the most frequent genomes; (ii) genome of interest may be absent from database.

Our Goals

- Don't count on intact cells at least not for all species of interest.

- Don't have time or resources to prepare microarrays for all possibilities or don't know in advance.

- Want to detect rare species as well as frequent ones.

Fundamental Observation

The first step to naming/sequencing what is in common between two mixtures is to discover common subcomponents of mixtures (i) without regard to frequency and *(ii) without an a priori idea of what they are*.

Our approach: "hash" the mixtures into buckets (here called pools) characterized by labels and then compare the sequences having similar labels.

In What Sense Hashing?

Consider a set S1 of 10,000 integers ranging in values from 0 to 1 billion and another set S2 of 20,000 integers in the same range. Both sets could contain many duplicates.

Suppose we want to discover which integers are common to the two sets.

Approach 1: Design a hash function h that maps each integer to say 100 different values. Take each member s of S1 and put s in a "pool" labeled ("S1", h(s)). Similarly for each member t of S2, put t in ("S2", h(t)). To find the intersection, we need only compare "S1" pools and "S2" pools such that the hash values are the same, e.g. ("S1", 17) with ("S2", 17).

Difficulty: there will be many distinct values (collisions) in each pool.

How to Reduce Collisions

Approach 2: Design two different hash functions h1 and h2 that each maps an integer to 100 values. Now characterize each s in S1 by "pool" ("S1", h1(s), h2(s)) and each t in S2 by ("S2", h1(t), h2(t)).

The pools have fewer distinct values.

Smaller pools have fewer duplicates.

Our technique closely follows this model and shares the essential properties, though the functions are chemical not mathematical.

Technology We Use

A *restriction enzyme* cuts a DNA strand at a certain letter sequence called a "recognition site" usually of length between 4 and 8. For example, the restriction enzyme SmaI cuts in the middle of CCCGGG.

Our Basic Algorithm

- Cut the DNA in each sample with one 6 site restriction enzyme. This gives a set of strand lengths. The cut is analogous to the hash function and the length to the hash value. The result is a set of "pools" of DNA of purity.

- Then do this again (but on the strands in each pool) with a 4 site restriction enzyme. This gives a new set of pools. Each pool is labeled with the size of its strands from the first restriction enzyme and the size from the second e.g. (13065, 432).

- Look for common sequences in similarly labeled pools.

- Rare sequences are likely to be in smaller pools.

- The method requires minimal DNA sequencing.
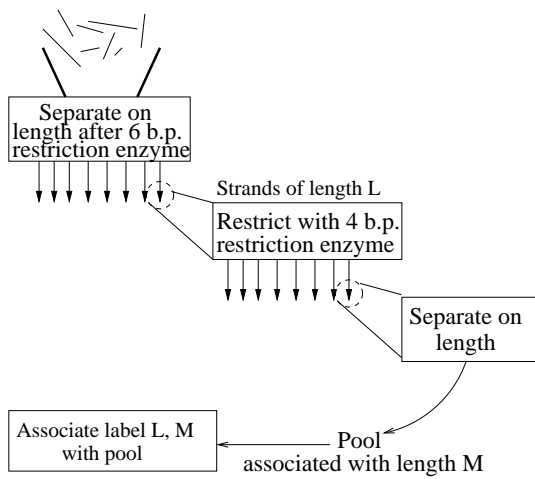
Example

Consider E coli.

Cut using the enzyme SmaI (recognising CCCGGG).

Take the pool corresponding to length 264.

Cut that pool with RsaI (GTAC) and take the pool of length 31.

We get a pool having label (264, 31). It happens to have a single member with the sequence CTATCCGCTCAATGAGTCGGTCGCCATTGCC.

By contrast, the pool with label (770, 207) has three different sequences.

Graphical Illustration of the Algorithm

Separate on
length after 6 b.p.
restriction enzyme

Strands of length L

Restrict with 4 b.p.
restriction enzyme

Separate on
length

Associate label L, M
with pool

Pool
associated with length M

Two-stage hash pooling

Error Considerations

Possible Objection: Separating fragments by length entails a certain inaccuracy imposed by the laboratory technique (e.g. electrophoresis).

A reasonable estimate of this error is plus or minus 10 base pairs.

In fact, this goes up slightly with length, but this is a conservative bound. So, we'll use this for convenience.

This inaccuracy may lead one to believe that two pools have the same label when in fact the labels are different. We will call these "neighbors."

The first cut may bring two pools together which will then be cut together into more pools.

Solution: 10 base pair separation

Labels L and L' are 10 base pair-neighbors if (i) the first component of L and the first component of L' are different but differ by 10 or less ($0 <| L[0] - L'[0] |\leq 10$); or (ii) the first component of L and L' are the same but the second components differ by 10 or less ($0 <| L[1] - L'[1] |\leq 10$).

Note that if the second components are close but the first ones are very different, we have no problem because the two pools will be separated by the first restriction enzyme cut.

For *E. coli* K12, the labels (188, 59) and (188, 106), for example, have no 10 base pair-neighbors and of course they won't be confused with one another.

Experiment 1: genome detection

Given a tube, T, of unknown DNA (perhaps from an environmental sample) and a known genome E Coli, are "reasonably sized" portions of that genome present in T, even if in small concentrations?

A "reasonably sized" portion is a sequence of length at least 200,000 base pairs (or roughly 5% of the length of a bacterial genome such as E Coli which is 4.6 million base pairs).

Experiment 1: Protocol

Question: Are "reasonably sized" portions of a known genome present in a sample?

1. Compute the *candidate set* of E Coli, meaning pools having no 10 base pair neighbors.
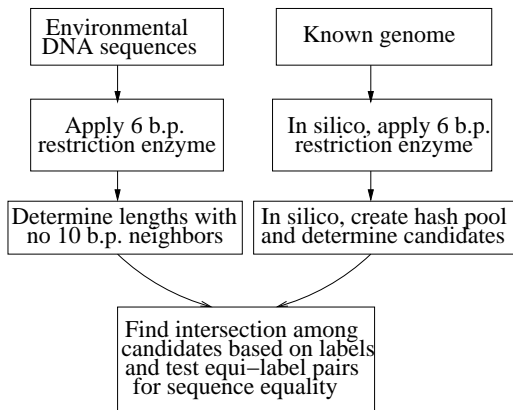   There are 3,567 candidates for E Coli using the 6 site restriction enzyme cutting at CCCGGG and the 4 site restriction enzyme cutting at AGCT. This gives us a "signature" of pools labels
   NB. This step is done entirely in silico. It can be computed just once for any combination of known genome and restriction enzyme set.

2. Cut the unknown sample T with the same two restriction enzymes (cut sites: CCCGGG and AGCT).

3. Label the resulting candidate set of hash pools from T. Sequence the found E Coli labels.

Protocol 1 in Pictures

Environmental DNA sequences	Known genome
↓	↓
Apply 6 b.p. restriction enzyme	In silico, apply 6 b.p. restriction enzyme
↓	↓
Determine lengths with no 10 b.p. neighbors	In silico, create hash pool and determine candidates

Find intersection among candidates based on labels and test equi–label pairs for sequence equality

Genome sequence detection

Experiment 1: Results

Question: Are "reasonably sized" portions of a known genome present in a sample?

Setup: take a 200,000 base pair subsequence of E Coli and enclose it in a sequence four times the length of E Coli but with the same GC ratio as E Coli.

On average (over 20 repeats of this setup and protocol) there are 70 labels produced that are the same as the E Coli signature. Virtually all of them (over 99%) come from that 200,000 base pair subsequence.

So very little sequencing is necessary to confirm a hypothesis of similarity.

Experiment 1: Issue

Will one have to generate many pools? This could be labor-intensive.

Fortunately, no. After the first cut with the six site restriction enzyme, one can filter out pools that have 10 base pair neighbors. Generally, this leaves very few (on the order of 5 or 6) to cut with the four site restriction enzyme.

Experiment 2: Challenge

Given two tubes of DNA, do they contain strands that are the same or very similar?

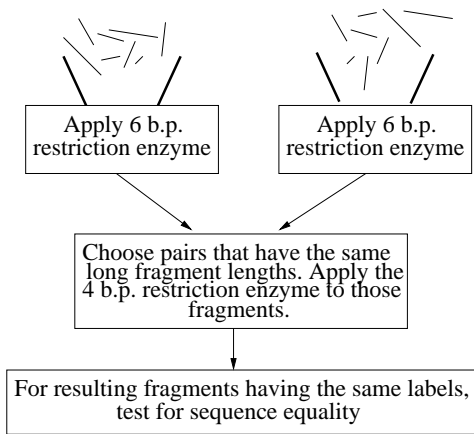Useful when comparing samples of unsequenced genomes.

Most samples are like this.

Experiment 2: Protocol

Question: Given two tubes of DNA, do they contain strands that are the same or very similar?

1. Cut each sample with the six base pair restriction enzyme, then find all lengths that are the same (within an accuracy of 10 base pairs).

2. On the upper dectile of those lengths (approximately 94 of them), apply the four base pair restriction enzyme.

3. Compare sequences for pools having the same labels.

Protocol 2 in pictures

Apply 6 b.p.
restriction enzyme

Apply 6 b.p.
restriction enzyme

Choose pairs that have the same
long fragment lengths. Apply the
4 b.p. restriction enzyme to those
fragments.

For resulting fragments having the same labels,
test for sequence equality

Sample comparison

Experiment 2: Results

Question: Given two tubes of DNA, do they contain strands that are the same or very similar?

Setup: Take 200,000 base pairs from E Coli. Call that E.

Create a sample S1 consisting of E embedded in 19 million randomly generated base pairs having the same GC ratio as E Coli.

Create a second sample S2 consisting of E embedded in 19 million (different) randomly generated base pairs having the same GC ratio as E Coli. Net effect of setup: common sequence E is just 1% of each sample.

After applying the protocol, an average of 14% of the pools having the same labels consist of subsequences of E. Roughly 1 in 7 pools sequenced will locate common strands REGARDLESS OF HOW INFREQUENT THOSE STRANDS ARE. (Choosing cuts resulting from the six base pair restriction having size near the expected value would have yielded the common strands only 0.4% of the

Conclusions

1. DNA Hash pooling is a method to simplify many problems in metagenomics.

2. Gives the experimenter the ability to query for known sequences and genomes in a sample or to find common sequences from unknown genomes in two or more samples *even if the identified sequences are rare*.

3. Basic protocol: given a sample of DNA, apply a restriction enzymes to it and if the right size apply a second smaller restriction enzyme.

4. Label the resulting "pools" with the lengths from each restriction enzyme cut. Compare sequences from pools having the same label.