Probe Selection Problem: Structure and Algorithms

Daniel CAZALIS
and
Tom MILLEDGE
and
Giri NARASIMHAN
School of Computer Science, Florida International University,
Miami FL 33199, USA.

ABSTRACT

Given a set of clonal sequences, the Probe Selection Problem is that of designing the smallest set of oligonucleotide probes so that every pair of clones (or at least a maximum number of the pairs) can be differentiated by at least one probe. We have implemented several algorithms that give an optimal or near-optimal solution to the problem. The quality of the results is measured using quantities such as entropy, number of differentiated sets of clones, size of largest differentiated set of clones, and number of differentiated pairs of clones. Our heuristic methods take advantage of the characteristics of the problem instances. An empirical approach is used to select the most efficient techniques as well as to better understand the structure of the problem. Many of our methods are modifications of existing algorithms or are based on commonly used combinatorial optimization techniques. In many cases, the results are better than those presented in the literature.

Keywords: Clone, Probe, Entropy, Oligonucleotide, Algorithm

INTRODUCTION

A short sequence of nucleotides (probe) will *hybridize* to a given cloned DNA sequence (clone) if the reverse complement of the probe is present as a subsequence in the clonal sequence. For this paper, we relax this definition and make the model more realistic by assuming that the hybridization can occur even if the reverse complement of the probe is present as a subsequence with one mismatched nucleotide. Given two clonal sequences, a probe is said to differentiate the two clones if the probe hybridizes to one, but not the other.

Since hybridization tests are easily implemented in a laboratory, the method of hybridizing short synthetic oligonucleotide probes to cloned DNA sequences is commonly used for many purposes, such as oligonucleotide fingerprinting, and cDNA gene expression analysis. Each of these problems start with the design of a probe set for a given clone set. In order to reduce the cost and effort, the number and length of probes need to be small. In practice, the set of clonal sequences often have similar characteristics.

The probe selection problem is NP-hard, and is therefore well suited for solution by heuristic methods. Ideally, the selected set of probes should be able to differentiate all the clones, in the sense that all clones should have different sets of vectors (“fingerprints”) that characterize the hybridization relationships between the clones and the set of probes. We say that two clones are indistinguishable by a set of probes when they have the same fingerprint. In that case, we also say that they belong to the same cluster. Roughly speaking, a set of probes is informative if it distinguishes most clones by generating a distribution with many clusters, each with at most a small number of elements.

A number of heuristic methods have been previously proposed for probe selection. The GC-content in relation with the expected frequency in a training database was used to select probes by Cuticchia *et al.* [1], Fu *et al.* [2] and Drmanac *et al.* [3]. This approach for the selection of probes gives results that show an improvement over random selection. However these frequency-based methods lead to the selection of many highly similar probes without a significant gain in information.

The method of Meier *et al.* [4] is based on a greedy algorithm and was shown to give better results than the random- and frequency-based selection methods by Herwig *et al.* [5]. Borneman *et al*. [6] showed that sophisticated optimization techniques such as Simulated Annealing and Lagrangian Relaxation give marginally better probe set selections than the greedy algorithm.

We propose, implement and compare a wide range of algorithms that give approximate solutions to the probe selection problem for an input set of clones in a fair and uniform framework. In Rash and Gusfield [7], a method with a near-optimal guarantee was proposed for selecting the smallest set of probes that provided a unique signature, or “barcode”, for viral sized genomes. Their method used suffix-trees to identify the critical substrings and integer-linear programming to solve the minimization problem. We do not implement this suffix-tree method for several reasons: they use probes of variable lengths, from 15 to as many as 40 bases (and, thus, makes it difficult to compare with our results). The selection of a proper subset of probes is the core of this method, but we concentrate on the algorithms applied to the set of probes not on the selection of them and we work with a more common length of 8 bases. They argue that this length constraint is itself a form of pre-selection, however our fixed-length probe sets are not derived from the clones in the study, whereas the suffix-tree method samples the genome sequence in order to generate candidate subsequences.

PROBLEM FORMULATION

There are two basic formulations of the probe selection problem:

1. Find a set of probes of a specified length that distinguish all given clones, and
2. Find a set of k probes of a specified length that distinguish as many clones as possible.

In this paper, we address only the second formulation. A more precise mathematical formulation of the probe selection problem is given below.

Maximum Distinguishing Probe Set Problem (MDPS): Given a set of clonal sequences, $C = \{c_1, c_2, \dots c_m\}$, and a set of probes $P = \{p_1, p_2, ... p_n\}$ over the alphabet {**A, C, G, T**}, find a set of probes, $S \subseteq P, | S |= k$ that maximizes dm(S, C), a measure of the set of clonal pairs that are distinguished by the probes in S.

Set *C* is, in general, a collection of long nucleotide sequences. On the other hand, set *P* consists of short nucleotide sequences typically of the same length (8 to 16 nucleotides long); the length k is usually chosen as $k \approx \log_2(m)$, and a typical problem may have about 1000 clones of different lengths, with 8-12 probes each of length 8 to be selected from a large selection of over 2^{16} probes. Note that $\log_2(m)$ is a theoretical lower bound on the number of probes needed to differentiate all the clones.

THEORETICAL BACKGROUND

Combinatorial Optimization deals with finding an optimal solution set over a discrete, finite solution space. For many optimization problems, finding the exact solution is only feasible for instances of relatively small size, and the complexity of finding an exact solution is NP-Hard. As is well known, NP-hard problems differ widely in the number of hard instances that occur in practice and also in the existence of near-optimal solutions. Some difficult problems can be solved exactly for real life instances; for others, the approximate solution is sufficiently good for all practical purposes. For many others, even the computable approximate solutions are infeasible. The MDPS problem is an NP-hard problem [6]. Are the instances of NP-hard problems that we encounter in computational biology science too difficult for heuristics and approximation algorithms to provide reasonably good solutions most of the time? What approaches best solve these problems? Are any of the heuristics better than others all the time?

In practice, the probe selection problem appears to be tractable by heuristic methods. However, only extensive testing and evaluation can find definitive answers to these questions. There are several reasons why this problem is tractable: 1) The probes are typically short (8-12 bases), 2) The number of probes to be selected is small in comparison to the number of clones (5-10 probes to distinguish thousands of clones), and 3) The DNA clonal sequences are not completely random, but are statistically biased with presumably meaningful patterns.

GOODNESS MEASURES

There are several measures of the goodness of a designed probe set. The main property of a goodness measure is that it must be optimal for any probe set that distinguishes all clones. For example, we can ask ourselves which of two probe sets, A and B, better distinguishes a set of 10 clones. Suppose that probe set A distinguishes 5 sets of 2 clones each, while B distinguishes 6 sets with 1 clone each and one set with 4. From probe set A, the largest undistinguished set of clones has size 2, which is smaller than that for B. On the other hand, probe set B has 6 clones perfectly distinguished, while probe set A has no clone perfectly distinguished.

Here we propose four different goodness measures. The four measures are (a) entropy of the distinguished clone sets, (b) number of distinguished clone sets, (c) the size of the largest distinguished clone set, and (d) the number of distinguished pairs of clones. However, among the four, we favor the entropy measure as being the most comprehensive and general measure.

Entropy (E)

This is given by the formula,

$$E = -\sum_{i=1}^{k} \frac{| C_i |}{m} \log_2 \left(\frac{| C_i |}{m} \right)$$

where $C_1, C_2, \ldots, C_k$ are k sets of undistinguishable clones for the given probe set. In other words, clones in two different sets are distinguished by the probe set.

This measure has several advantages: it is fairly easy to calculate; it takes into consideration the number and sizes of the sets; it is used in many algorithms; its range of values is limited; and its value gives you an idea of how good the solution is in relation to the optimal.

Number of sets (NS)

This measures the number of sets of undistinguishable clones. In general, the larger this measure, the better is the probe set. The disadvantage is that if we have at least one set with a large number of clones, then the largeness of this value could be misleading.

Largest set (LS)

This is a measure that complements the number of sets. It too can be misleading if there are many sets with the largest value. It only takes into consideration the cardinality of one set ignoring all the other sets. It is clear that two different distributions with the same largest set may give completely different values when using other measures. For example, using this measure, 8 clone sets with size distribution of {1, 1, 1, 1, 1, 1, 2} will be judged to be of the same quality as 4 clone sets with a size distribution of {2, 2, 2, 2} even though the latter is a worse solution.

Number of pairs (NP)

This measure over the set $C \times C$ is important because it tells you how many clones can be distinguished by the selected set of probes. The range of this measure grows without bound and is dominated by the size of the largest set.

All the above measures have a unique optimal value for the optimal solution in which all clones are distinguishable by the selected set of probes. Thus, for the optimal solution, NP is 0, LS is 1, NS equals m, the number of clones, and finally, entropy E equals $\log_2 m$.

LOCAL OPTIMIZATION TECHNIQUES

The type of algorithms we study here start with an initial solution set S_0 (the empty set or a randomly generated set of probes), which is changed by a heuristically-guided iterative process, to a final solution set S_f which represents a local optimum. S_f is obtained when there is no neighbor solution with a better measure of goodness. Two kinds of neighborhood relationships are used in most of the algorithms presented here: augmented neighborhoods and pivot neighborhoods. We say that S_i, S_{i+1} are augmented neighbors iff $|S_i|+1=|S_{i+1}|$ and $S_i \subset S_{i=1}$. This is the kind of neighbor used by the so-called Greedy algorithm, in which a solution is built by successively adding probes.

A pivot neighbor is built by substituting a probe in the solution set by a probe that was not in the set. We say that S_i, S_{i+1} are pivot neighbors iff $|S_i|=|S_{i+1}|=k$ and $|S_i \cap S_{i=1}|=k-1$, i.e., the sets differ in exactly one element. Figure 1 below shows a schematic of the neighborhood relationship among solution sets.

ALGORITHMS IMPLEMENTED

Affinity-based Algorithm

A pairwise relation between the probes is used to devise a heuristic for a steepest descent algorithm.
Given a symmetric matrix A[n,n] find a set of index P={ p1, p2, p3…} given by the property vector xp={xi=1 iff i$\in P$, 0 otherwise} so that:

(1) $$\sum_{i,j\in P} A[i,j] = \sum_{i+1..n} x_{pi}(x_p{}'A[i])$$

is maximal

A greedy algorithm will select the probe with higher pairwise value with the previous set off probes until k sets are selected.

1. Build a square matrix of the affinity for every pair of probes in P from the {1,-1}-matrix $P \times P$.
2. Select the 2 probes with the best affinity.
3. Select a probe that together with the former ones generates a better sum over the affinity matrix until the size equals k.

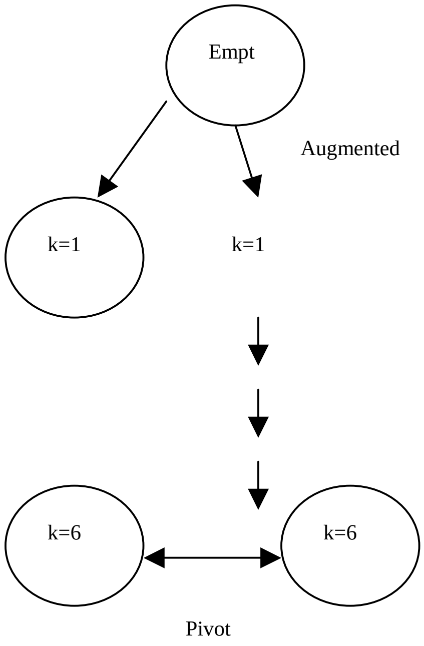

Figure 1: Solution Set Neighborhood

Greedy Algorithm

In the greedy algorithm, once a probe has been selected it stays in the solution set. A major disadvantage of this heuristic is that initial steps are taken when there are not many elements to validate the option. It only moves from a solution set to its augmented neighbors. No pivot neighbor solutions are considered.

1. Select a probe that best partitions the set of clones.
2. Select a probe that together with the previously selected ones partition the set better (with respect to the goodness measure).
3. If number of probes selected is less than k, go to 2.

Pivot Last Algorithm

In pivot last, the idea is to find a local minimum using pivot neighbor iteration after stopping from a greedy process. This sometimes improves the value of the final greedy solution but many times the final solution stays the same.

1. Apply the greedy algorithm first.
2. For every probe in the solution and for every probe not in the solution, exchange the probes if the resulting solution displays an improvement (pivot step).
3. If the goodness measure improves, change the selection and go to 2.

Pivot Greedy Algorithm

This is a cross between the greedy algorithm and the pivot algorithm. Every time we take an augmented step we look for a local minimum by repeated pivot steps. Selected probes may be removed in the pivot step but only if it leads to a better-valued solution.

1. Select a probe that best partitions the clone set
2. Select a probe that together with the previously selected ones partitions the clone set better.
3. For every probe selected for the solution, and for every probe not selected for the solution, exchange them and measure its goodness.
4. If the goodness measure improves, select the best improvement and go to 3.
5. If the number of probes is less than k, go to 2.

The algorithm stops because the loop must end for any of the proposed measures.

Genetic Algorithm

Start with a population of probe sets with k probes. In each generation, combine them to obtain many new probe sets, and select the best for survival in the next generation while eliminating the rest. Repeat for a large number of generations.

1. Randomly select a population of 1000 probes.
2. Let a small population with the highest goodness measures survive with no change.
3. Randomly combine pairs of probes in the population to make new recombined solutions until a large population is produced.
4. Kill the worst probes until a small population size is reached.
5. Repeat for a given number of generations

Random Pivot Algorithm

RP is a very simple algorithm that only takes random pivot steps and gives surprisingly good results. Take an initial random solution and select a random element from the solution set and one that is not in the solution set, pivot if the goodness measure improves, else keep trying until stopping condition is reached. We stop if many pivots result in no improvements.

```
1.  Select a random set of k probes
2.  Do a random pivot, and measure the goodness.
3.  If improvement results, change the set.
4.  If no improvement is seen in more than k*(size of set of
    probes) iterations, then exit; otherwise go to 2.
```

Simulated Annealing (Simula)

A temperature parameter oversees the algorithm. Random pivot steps that do not result in an improvement in the goodness measure may be taken when the temperature is still high in order to get out of local minima. At low temperatures it behaves like the greedy algorithm.

```
1.  Select a random set of k probes.
2.  Set the initial temperature.
3.  Do a random pivot, (set)=>newset.
4.  Change to the new pivot with the probability
        Min{1, exp( (dm(newset)-dm(set))/t)
5.  Diminish the temperature by a constant factor.
```

EXPERIMENTATION

We tested the above suite of algorithms on both real and synthetic data. All algorithms were applied to a set of 536 viral genomes we obtained from Rash and Gusfield from their string barcoding study [7]. We also tested the set of 579 Gyrase A sequences from Genbank, as well as to 1800 randomly Gyrase B sequences. The Gyrase data sets were interesting because the clones had much higher similarity than the viral sequences, making the design of a probe set a difficult problem. For the real data sets, the number of probes was varied from 10 through 14. For synthetic data we randomly constructed 20 problems with 128 clones and 300 probes and applied all the algorithms to each input data set. We generated all possible probes of length eight, of which there were 32896 we then selected 2000, and applied our algorithms to this dataset. All our results on the real and synthetic datasets were done with a fixed probe length of seven, eight, and nine we confined our search to all probes that appear in at least one clone accepting one missed. The results are as shown in the graphs and the table below.

RESULTS AND CONCLUSIONS

The objectives of this study were to compare and evaluate different heuristic algorithms for the probe selection problem using fixed length probes. The results from synthetic data sets and real data sets were consistent in the sense that the same algorithms performed comparably. For the real data sets, the running time varied considerably, since a relatively large number of probes (of the order of 2000) were generated for them. The number of iteration for the Genetic and Random Pivot algorithms were set to give similar running times than Greedy Pivot the slowest one, Greedy runs was faster than all of them and Pivot Last runs in top of Greedy with only one or two Pivot steps.

All algorithms use entropy as the goodness measure, but results using other measures of quality are given for comparison (Table 1).

In general good solutions also have neighboring good solutions for both augmented and pivot neighbors. Changing many elements of the solution did not seem to be particularly helpful. The inferior performance of the genetic algorithm corroborates this claim.

The pure Greedy method easily gets stuck in local minima. The Pivot Last algorithm is marginally better than pure greedy in this sense. But in many cases it is incapable of improving a solution just by pivoting. Pivot Last improved the Greedy algorithm solutions only when it was in a "small" local minimum. This happened in less than 50% of the cases and the improvement obtained by Pivot Last was relatively minor (Figure 1).

The alternation of Augmented Neighbor and Pivot Neighbor used in the Pivot algorithm consistently gives the best results with both synthetic and real data (Figure 2). In the synthetic datasets, Random Pivot performed better than Simulating Annealing which seems to suggest that the results obtained by S.A. are due more to the improving pivot steps in the low temperature section than to the worsening pivot steps in the high temperature section (Figure 1).

REFERENCES

1]. Cuticchia, A., J. Arnold, and W. Timberlake, *PCAP:* **Probe choice and analysis package--a set of programs to aid in choosing synthetic oligomers for contig mapping.** Comput. Appl. Biosci., 1993. **9**(2): p. 201-203.

[2]. Fu, Y.X., W.E. Timberlake, and J. Arnold, **On the Design of Genome Mapping Experiments Using Short Synthetic Oligonucleotides.** Biometrics, 1992. **48**(2): p. 337-359.

[3]. Drmanac, S., N.A. Stavropoulos, I. Labat, J. Vonau, B. Hauser, M.B. Soares, and R. Drmanac, **Gene-Representing cDNA Clusters Defined by Hybridization of 57,419 Clones from Infant Brain Libraries with Short Oligonucleotide Probes**. Genomics, 1996. **37**(1): p. 29-40.

[4]. Maier, E., S. Meier-Ewert, D. Bancroft, and H. Lehrach, **Automated array technologies for gene expression profiling.** Drug Discovery Today, 1997. **2**(8): p. 315-324.

[5.] Herwig, R., A.O. Schmitt, M. Steinfath, J. O'Brien, H. Seidel, S. Meier-Ewert, H. Lehrach, and U. Radelof, **Information theoretical probe selection for hybridisation experiments.** Bioinformatics, 2000. **16**(10): p. 890-898.

[6]. Borneman, J., M. Chrobak, G. Della Vedova, A. Figueroa, and T. Jiang, **Probe selection algorithms with applications in the analysis of microbial communities.** Bioinformatics, 2001. **17**(90001): p. 39S-48.

[7]. Rash, S. and D. Gusfield. *String* **Barcoding: uncovering optimal virus signatures.** in *RECOMB*. 2002.

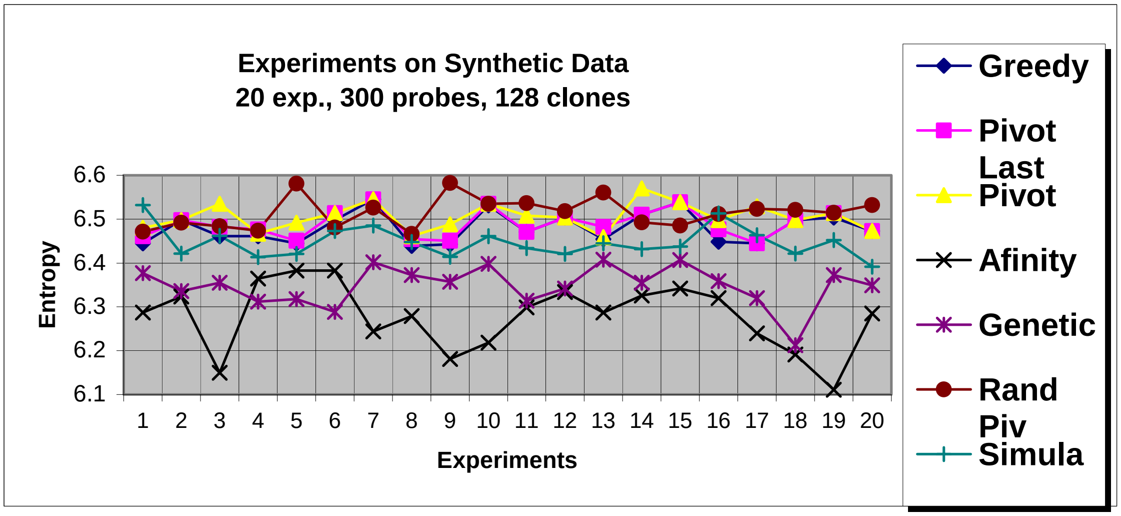

Figure 1: Overall results of 20 experiments of synthetic data: 128 clones, 300 probes.

PS	Results for Real Data (Length, Missed)									
	Algorithm	Gyrase B (8,1)			Gyrase A (7,1)			Viral Sequences (9,1)		
		Entropy	**NS**	**LS**	**Entropy**	**NS**	**LS**	**Entropy**	**NS**	**LS**
8	Genetic	6.51681	138	20	7.17872	240	92	7.24911	212	9
	Random Pivot	6.89692	167	17	7.42104	248	72	7.36812	225	7
	Greedy	6.82886	167	18	7.34425	243	60	7.38354	224	9
	Pivot Last	6.8737	172	20	7.40305	244	58	7.38354	224	9
	Greedy Pivot	6.97776	179	13	7.48133	249	51	7.41775	228	9
9	Genetic	6.94921	193	24	7.72637	401	85	7.8018	307	8
	Random Pivot	7.26134	209	13	8.08305	408	60	7.99156	333	5
	Greedy	7.21894	213	16	7.9279	402	58	7.93615	328	7
	Pivot Last	7.29441	221	16	8.04343	422	53	7.93615	328	7
	Greedy Pivot	7.33412	220	11	7.98138	432	38	7.96286	327	7
10	Genetic	7.25508	225	14	8.16058	510	58	8.05224	358	6
	Random Pivot	7.59323	264	12	8.54307	590	51	8.25645	387	5
	Greedy	7.50505	254	14	8.40456	581	53	8.27827	396	5
	Pivot Last	7.55605	262	12	8.54915	599	45	8.27827	396	5
	Greedy Pivot	7.60498	267	11	8.60101	606	37	8.32861	413	6
11	Genetic	7.45709	256	17	8.58009	680	49	8.28232	410	6
	Random Pivot	7.77116	292	12	8.95783	735	45	8.49102	438	4
	Greedy	7.70693	285	14	8.76439	720	47	8.48211	444	4
	Pivot Last	7.7441	293	16	8.89523	766	40	8.48211	444	4
	Greedy Pivot	7.8441	303	12	8.97972	777	33	8.5198	451	6
12	Genetic	7.61145	271	16	8.74391	722	47	8.43178	442	4
	Random Pivot	7.97221	330	13	9.14348	840	39	8.62003	465	4
	Greedy	7.85252	308	14	9.01763	832	40	8.60109	468	4

	Pivot Last	7.9002	317	14	9.12942	880	37	8.61369	470	4
	Greedy Pivot	8.01836	341	11	9.21264	897	31	8.64187	475	5
13	Genetic	7.60645	285	19	9.14016	882	38	8.55773	458	7
	Random Pivot	8.1117	351	9	9.33875	964	31	8.66456	481	4
	Greedy	7.96623	330	14	9.20693	927	38	8.6743	482	4
	Pivot Last	8.00999	340	16	9.30069	962	31	8.709	480	3
	Greedy Pivot	8.13328	357	9	9.36887	981	29	8.7468	490	4

Table 1: Results on Data sets for Viral Sequences, Gyrase A and Gyrase B (PS – probes selected, NS – number of sets, LS – largest set).

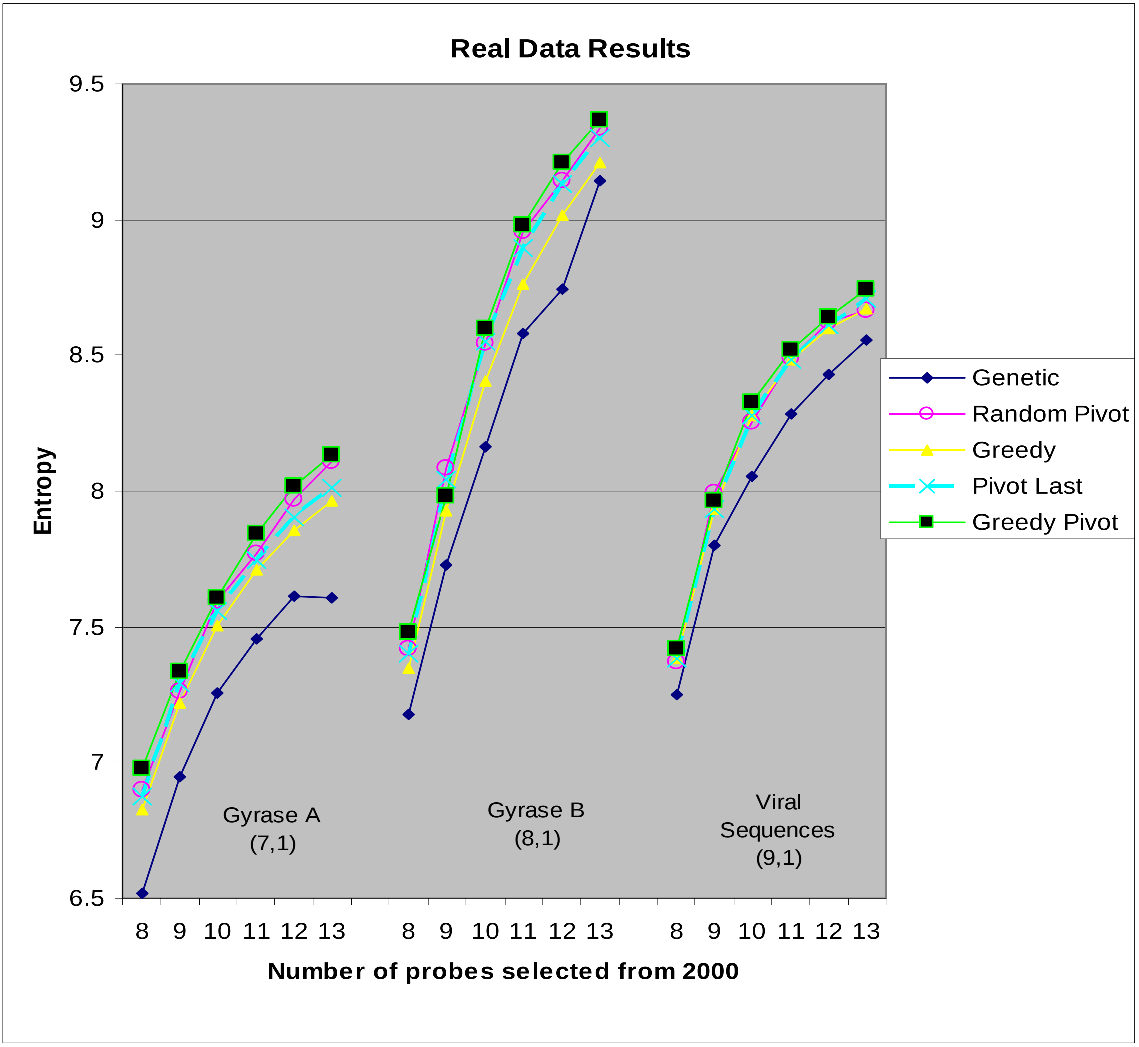

Figure 2: Overall results for three data sets.