

Polypeptide Input Cardinality and its Effect on the Quality of MAFFT-generated Multiple Sequence Alignments

Nithya Chandrasekaran and Kyriacos Pavlou

May 2, 2007

1 Introduction

Multiple sequence alignment is a well-studied algorithmic problem of particular importance in Biology. In general, a sequence alignment is a way of arranging sequences of proteins to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. This similarity due to shared ancestry between different proteins is termed *homology* [1]. Hence, homology among proteins is often concluded on the basis of sequence similarity. This paper investigates the effects of varying the cardinalities of input protein sequences of the same family, on the quality of a multiple sequence alignment. In the next section we give a brief introduction to the biological context of the research question and discuss relevant terminology. Section 3 features the theoretical formulation of the multiple sequence alignment problem along with associated software solutions. In section 4 we formulate a theory regarding the relationship between cardinality of input sequences and the quality of the resulting alignment, and propose three hypotheses to test the validity of this theory. In section 5 we describe the design and experimental setup developed to test our hypotheses. Section 6 presents and discusses the experimental results for each hypothesis, while section 7 concludes with proposed future work.

2 Biological Background

Proteins are one of the most important and abundant molecules in nature. They play a crucial structural and metabolic role in living organisms because they constitute the building blocks of tissues and can regulate metabolism through their enzymic activity. These biological macromolecules are comprised of many amino acid molecules linked together with special chemical bonds called *peptide bonds* [4], hence the alternative name of *polypeptide chain* used for protein molecules. In each sequence a single letter can be used to represent a single amino acid in the chain. A single protein, can be made up of many amino acids, of which there are 20, and at each position in the chain repetition is allowed. For this reason, we can conceptually think of proteins as finite strings over an amino acid alphabet Σ .

Even though proteins have a complex three-dimensional structure, which determines their function, in this paper we are only concerned with their *primary structure*. The primary structure of a protein molecule is the succession of amino acids linked together in the polypeptide chain, *without* any regard to spatial arrangement. Figure 1 shows an example of the primary structure of a segment of a protein sequence.

As we have mentioned in the introduction, the similarity between different proteins due to shared ancestry in the process of evolution is called homology. In order to capture the evolutionary history of several proteins scientists construct *phylogenetic trees*. The leaves of these trees are the current forms of the proteins while internal nodes are hypothesized common ancestors [4].

Moreover, different proteins are classified into *protein families*. Proteins which exhibit similar phylogeny, which can manifest as structural or functional similarities between different sequences are said to belong to the same protein family.

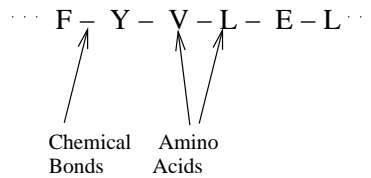


Figure 1: Primary Structure of a Protein Sequence

3 Multiple Sequence Alignment (MSA) and Software Tools:

Multiple sequence alignment is an algorithmic technique used in order to detect homology among protein sequences. Usually, it is preferable to align sequences which belong to the same protein family since the resulting multiple alignment is more informative from a biological perspective. The problem of multiple alignment is formulated as follows.

Given a collection of strings S_1, S_2, \dots, S_k and a cost function $\delta : (\Sigma \cup \{\epsilon\})^k \rightarrow \mathbb{R}$, the *General Multiple Alignment Problem* is to find an alignment $\mathbb{A} = (a_{ij})$ where $1 \leq i \leq k$ and $1 \leq j \leq l$, that minimizes

$$\sum_{1 \leq j \leq l} \delta(a_{1j}, a_{2j}, \dots, a_{kj}).$$

The objective function is a sum over all columns of a column cost function. This formulation of the problem is NP-complete.

In essence MSA tries to arrange sequences next to one another with similar elements juxtaposed. An arrangement of sequences where the cost of converting one sequence to another, for all sequences, is minimized is the desired multiple alignment. The conversion of one sequence to another is achieved through three different transformations, as implied by the cost function δ . These transformations are:

1. *Substitution*: The replacement of one letter with itself (identity) or with a different letter.
2. *Insertion*: The insertion of a letter in order to produce a longer sequence. A dash appears in the corresponding position of other sequences which have not been augmented in the alignment.
3. *Deletion*: The deletion of a letter in one sequence of the alignment. A dash at that position denotes its absence. Note that a deletion in one sequence is symmetric with an insertion in another.

```

S1:      ... F G V L E Q ...
S2:      ... - G F L - E ...
S3:      ... S - V E Q M ...
position: 1 2 3 4 5 6
  
```

Figure 2: Example of Multiple Alignment of three Sequences

Multiple software tools have been developed to provide fast solutions to the above problem. One such tool is MAFFT, which is a multiple sequence alignment program for amino acid or nucleotide sequences. It was developed at Kyoto University, it is open-source and it can be run in UNIX-like operating systems, accepting input files in FASTA format [3] [2]. We have chosen to use MAFFT in our experiments, over other available tools because there are several advantages associated with MAFFT. Apart from the the fact it is free, MAFFT is fast, accurate and ease to install and use.

In conjunction with MAFFT, our experiments will utilize the PALI benchmark. PALI is a database of phylogeny and alignment of homologous protein structures [5]. Each member in a protein family has been

structurally aligned with every other member in the same family (pair wise alignment) and all the members in the family are also aligned using simultaneous super-position (multiple alignment). PALI is a useful resource to help in analyzing the relationship between input cardinality and MSA quality, hence it is used as the benchmark for our experiments.

4 Proposed Theory and Hypotheses

We would like to see the effect of varying the number of sequences, participating in multiple sequence alignment performed by MAFFT, on the quality of the resulting alignment. More specifically, we expect that when we align sequences belonging to the *same* family, the higher the number of sequences aligned, the better the quality of the MAFFT-generated alignment.

Each PALI file comprises of protein sequences of the same family. The ancestral homology between these proteins is modeled using a phylogenetic tree. Addition of sequences belonging to the same family will be as branches to the existing tree without changing its root. Figure 3(a) shows a tree with only five sequences, while Figure 3(b) shows the structure of the tree after addition of more protein sequences (S_2, S_4, S_7, S_8). It is precisely because of this topological constraint that the addition of sequences is expected to provide more information during MSA and consequently result in a better multiple alignment. An objective way to quantify the quality of a multiple alignment is to calculate the *percentage recovery* of the MAFFT-generated alignment compared to the PALI benchmark alignment of the same sequences. It is calculated by looking at each position and comparing the letter present in the MAFFT alignment with that in the benchmark. The higher the agreement between the two alignments the higher the % recovery.

Furthermore, we will use *percentage identity* in our experiments as a metric of the evolutionary distance between two protein sequences. Percentage recover is calculated as the number of identity substitutions divided by the number of non-gaps.

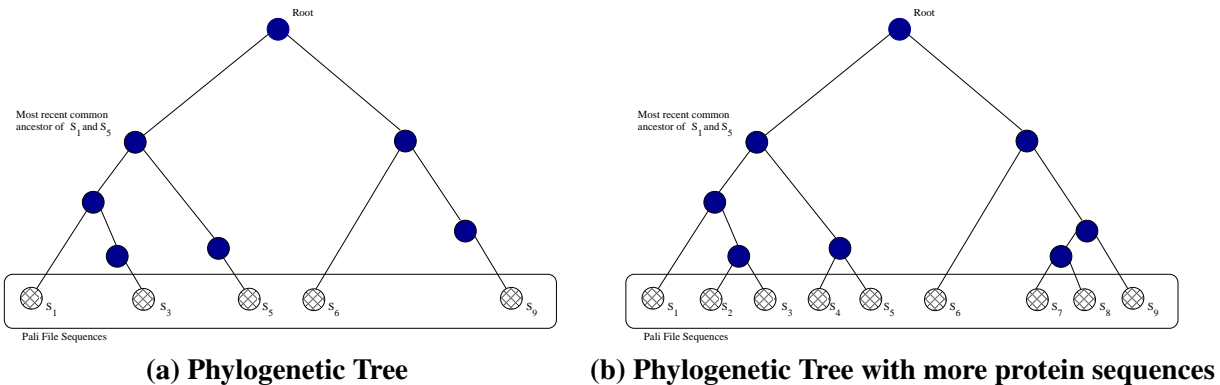


Figure 3: Change in the Structure of a Phylogenetic Tree with the addition of proteins of the same family

In order to test our theory we propose the following hypotheses.

Hypothesis 1. *An increase in the number of sequences of the same family used by MAFFT to obtain an alignment, results in an improvement of the generated alignment, i.e., it leads to an increase in % recovery.*

This hypothesis can be tested by checking if there exists a positive correlation between sample size and % recovery. Hypothesis 1 tests directly the impact of cardinality on the % recovery of the multiple alignment.

Hypothesis 2. *An increase in the number of sequences of the same family included in a benchmark alignment will increase the percentage identity among all the sequences in the benchmark.*

This hypothesis can be tested by checking if there exists a positive correlation between sample size and % identity. We expect hypothesis 2 to be true since the addition of more protein sequences of the same family will increase the % identity among all sequences because of the topological constraint, described above.

Hypothesis 3. *An increase in the mean % identity for a benchmark alignment of proteins of the same family leads to an increase in the mean % recovery of the MAFFT-generated alignment of the same set of sequences.*

The rationale for hypothesis 3 is that we expect more divergent sequences (i.e., sequences with smaller % identity) to be harder to recover (i.e., their alignment has low % recovery).

Note that hypotheses 2 and hypothesis 3 imply hypothesis 1. More precisely, if an increase in the cardinality of the input sets leads to an increase % identity and an increase in % identity leads to an increase in the % recovery then we can infer that an increase in cardinality will lead to an increase in the % recovery.

5 Experimental Design and Setup

The sequence alignment tool used in the experiment is MAFFT version 6.237beta. The whole experimental process was automated using Perl scripts (approximate code length is 700 lines) and the scripts were repeatedly called for all the PALI files using a wrapper script. Each file has sequences from the same family. Eighteen files were selected from the PALI benchmark, each having a cardinality higher than or equal to 19. For each such file, subsets were created, with each element in the subset selected at random. The implemented scripts and their corresponding function is given below.

- `630wrapperScript.pl`

Figure 4 shows the flow of control across the experimental process. The unaligned input file and its corresponding benchmark file (for each family) is fed to a script (`630CreateInputFiles.pl`), which generates the subset of sequences. The files containing the unaligned sequences are fed into `630runMAFFTcalcRecovery.pl`, which run MAFFT and generates the alignment for those sequences. Then it uses the `Qscore` executable to score the MAFFT output with the benchmark (aligned input files) and generates the recovery score (output into `recoveryScore`). The percentage identity score (`630calculateIdentity.pl`) for the aligned input files is generated and written into a text file (`identityScore`). The `input.dat` file contains the `identityScore` and `recoveryScore` data for any set of input files. The results are used to compute the correlation coefficient of the data points (tuples of percentage recovery and percentage identity sets for each input file, varying in cardinalities, from the same family). It reads the recovery scores and identity scores from the respective output files and prints it into the format as read by the `GNUplot` script. It also executes a `gnu` plot on the set of resultant scores and does scatter plots of % identity (y-axis) vs cardinality (x-axis) and % recovery (y-axis) vs cardinality. This is to be done across a number of files so that a conclusion can be drawn about the behavior of percentage identity and percentage recovery of the sequences by varying the cardinality of sequence subsets for different families. The supporting scripts are described `630wrapperScript.pl` in detail below.

- `630CreateInputFiles.pl`

This script generates subsets of sequences into different files, from the unaligned input file, each with a different cardinality such that the sequences, for each file, are picked in random from the comprehensive input file. The script also generates the corresponding benchmark subset files for the same sequences as in the sample picked during the random input selection (using a random seed). The subsets are created in such a way that their cardinalities change in increments of 2, starting from 7 and ending up to the cardinality of the entire file. For example for files with 20 or sequences, subsets of cardinalities 7, 9, 11, 13, 15, 17, 19 are created.

- 630runMAFFTcalcRecovery.pl

Given an input file and its corresponding benchmark alignment, this script takes the unaligned input sequence file and feeds it as input to MAFFT in order to create a multiple sequence alignment. It then compares the resulting alignment with a preexisting benchmark alignment available for the specific input. The degree of similarity between the MAFFT-generated alignment and the benchmark alignment is measured as a percentage recovery. Percentage recovery is calculated by using an executable called Qscore to score the MAFFT aligned and benchmark sequences.

- 630calculateIdentity.pl

Given a MAFFT-generated multiple alignments, the script for calculating percentage identity, parses the input file, compares two sequences at a time (pair wise comparison) and generates the identity percentage between the sequences. The degree of similarity between the two sequences is measured as the percentage identity. Identity between two sequences is calculated by checking whether an amino acid at a specific position of one aligned sequence is the same as its counterpart in the other sequence. This comparison is performed for all positions of the two alignments. All those that match are counted and divided by the length of the sequences. The final percentage identity is computed as the sum of percentage identities divided by the number of pairs compared.

- 630corCoefficient.c

This program reads from the input file containing a set of % recoveries and % identities and calculates the correlation coefficient between the data points and the variance of each set.

- Qscore

Executable used to score the MAFFT-generated multiple alignment.

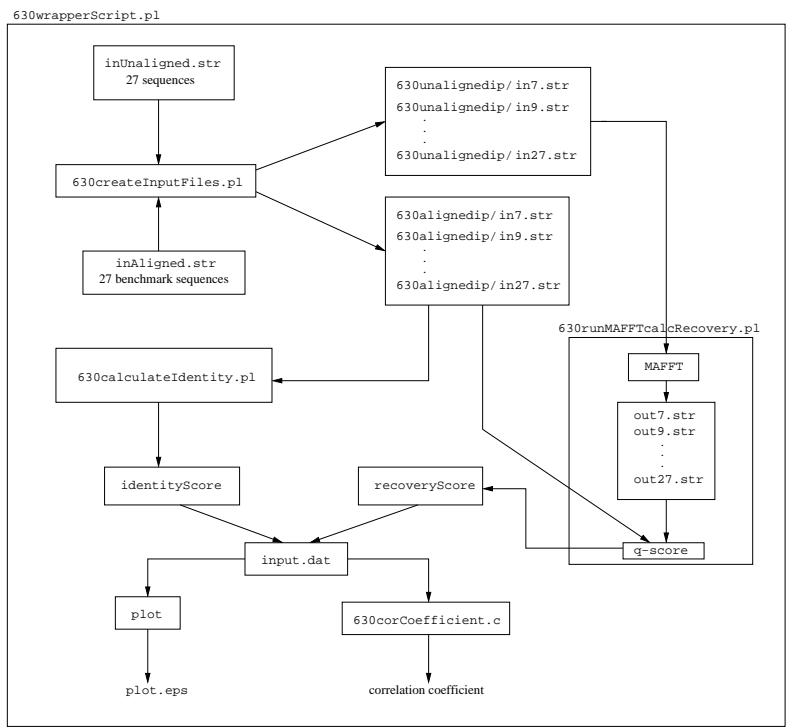


Figure 4: Perl scripts Interaction

We proceed to describe the three experiments we performed in order to test each one of our three hypotheses. The correlation between measured quantities is captured by correlation coefficients r . A r closer to 0 indicates low correlation, a r closer to 1 indicates high positive correlation, while a r closer to -1 indicates high negative correlation.

Experiment 1: For hypothesis 1, scatter plots are constructed with cardinality (of subsets) as x-axis and % recovery as y-axis. Each point in the plot represents a subset of sequences and each scatter plot contains data points from the subsets of the same family. Correlation coefficient r is computed for the data points in each file. As a result of this, we will have 18 scatter plots of % recovery vs cardinality, and their corresponding r 's.

Experiment 2: For hypothesis 2, scatter plots are constructed with cardinality (of subsets) as x-axis and % identity as y-axis. Each point in the plot represents a subset of sequences and each scatter plot contains data points from the subsets of the same family. Correlation coefficient r is computed for the data points in each file. As a result of this, we will have 18 scatter plots of % identity vs cardinality, and their corresponding r 's.

Experiment 3: For hypothesis 3, a scatter plot is constructed with mean %recovery as the x-axis and mean %identity as the y-axis (mean of the corresponding field in all the files). A regression line is drawn to visually understand the correlation of the points.

6 Experimental Results and Discussion

The results of the experiments are presented below.

6.1 Experiment 1 Results

The table in Figure 5 shows all 18 PALI files tested. For each such file a scatter plot was created of % recovery versus the cardinality of each subset. All scatter plots are given in Appendix A. For each plot the product moment correlation coefficient r_{rec} between % recovery and cardinality was calculated. Figure 5 shows that 8 out of 18 scatter plots exhibit a negative correlation while the majority exhibits a positive correlation. This seems to suggest that hypothesis 1 is not always true. In order to see whether these two quantities are always positively correlated we calculate 95% confidence limits ($z_{95\%} = 1.96$) for the correlation coefficient of entire protein family population ρ . The limits are given by following formula.

$$L_1 L_2 = \tanh^{-1} r_{rec} \pm z_{95\%} \cdot \sqrt{\frac{1}{n-3}}$$

$$\tanh L_1 \leq \rho \leq \tanh L_2$$

The results suggest that there exists a correlation between % recovery and input cardinality but the evidence are not as conclusive as to the positive nature of the correlation. Thus, testing hypotheses 2 and 3 will provide us with more evidence concerning the effect of the input cardinality.

6.2 Experiment 2 Results

The table in Figure 6 shows the PALI files, their cardinalities and the correlation coefficient between % identity and cardinality.

File	Corr.Coeff. r_{rec}	Cardinality n	95% Confidence Limits for ρ
Pali552	-0.111053	23	$-0.500357 \leq \rho \leq 0.315603$
Pali1092	0.168559	20	$-0.296052 \leq \rho \leq 0.568669$
Pali1392	0.370748	23	$-0.048940 \leq \rho \leq 0.679163$
Pali578	0.004843	27	$-0.375869 \leq \rho \leq 0.384156$
Pali1093	-0.509069	21	$-0.771267 \leq \rho \leq -0.099169$ (-)
Pali1444	-0.101387	41	$-0.396670 \leq \rho \leq 0.212910$
Pali19	-0.398237	21	$-0.708182 \leq \rho \leq 0.040403$
Pali792	-0.628951	23	$-0.826803 \leq \rho \leq -0.292602$ (-)
Pali1250	-0.134759	28	$-0.483532 \leq \rho \leq 0.250940$
Pali1486	-0.915625	19	$-0.967476 \leq \rho \leq -0.789937$ (-)
Pali475	-0.683076	29	$-0.839432 \leq \rho \leq -0.422286$ (-)
Pali809	0.165167	28	$-0.221569 \leq \rho \leq 0.507008$
Pali1254	0.929702	21	$0.831881 \leq \rho \leq 0.971491$ (+)
Pali1521	0.858814	19	$0.663370 \leq \rho \leq 0.944567$ (+)
Pali477	0.040561	24	$-0.368878 \leq \rho \leq 0.436817$
Pali815	0.371683	36	$0.049143 \leq \rho \leq 0.624023$ (+)
Pali1360	0.088908	22	$-0.345664 \leq \rho \leq 0.492078$
Pali1539	0.325098	37	$0.001200 \leq \rho \leq 0.587261$ (+)

Figure 5: The PALI files tested, their correlation coefficients (r_{rec}) and the corresponding 95% Confidence Interval on the population correlation coefficient ρ . Four protein files with negative correlation are marked with (-), while another four files having a positive correlation are marked with (+). In all other cases ρ can be positive or negative.

File	Corr.Coeff. r_{id}	Cardinality n
Pali552	0.794841	23
Pali1092	0.934524	20
Pali1392	-0.927521	23
Pali578	0.869510	27
Pali1093	-0.665508	21
Pali1444	-0.263146	41
Pali19	0.891321	21
Pali792	-0.185568	23
Pali1250	-0.579727	28
Pali1486	-0.385856	19
Pali475	-0.936684	29
Pali809	0.891794	28
Pali1254	0.907760	21
Pali1521	0.883946	19
Pali477	0.205365	24
Pali815	-0.548219	36
Pali1360	0.800224	22
Pali1539	0.902746	37

Figure 6: The PALI files tested, their correlation coefficients (r_{id}) of % identity vs cardinality

The scatter plot given in Appendix B suggest that there is a non-linear relationship between the two quantities. For this reason, the calculated r 's exhibits either positive or negative correlation. Even though this result, at first glance, disproves hypothesis 2, we believe that it is an artifact of our sampling technique. Plots which exhibit negative correlation such as Pali815 start with a subset of sequences that are closer together in the phylogenetic tree. Subsequent additions of new sequences will reduce the percentage identity to a stable value which is percentage identity of the entire Pali815 file. On the other hand, a plot such as Pali552 which shows a positive correlation corroborates the validity of hypothesis 2 because the initial subset contains proteins that are far apart in the phylogenetic tree. Subsequent additions of new sequences will increase the percentage identity up to a stable value which is the percentage identity of the entire Pali552 file.

6.3 Experiment 3 Results

In the third experiment we calculate the mean % identity of all 18 PALI files and plot it against their corresponding mean % recovery. The scatter plot is shown in Figure 7. As can be observed by the regression line in the scatter plot, the two quantities have a negative correlation. This seems to suggest that hypothesis 3 is false. However, we believe that the positive correlation predicted by hypothesis 3 is masked by the sampling technique which gave unexpected results as described in the results of hypothesis 2 as well.

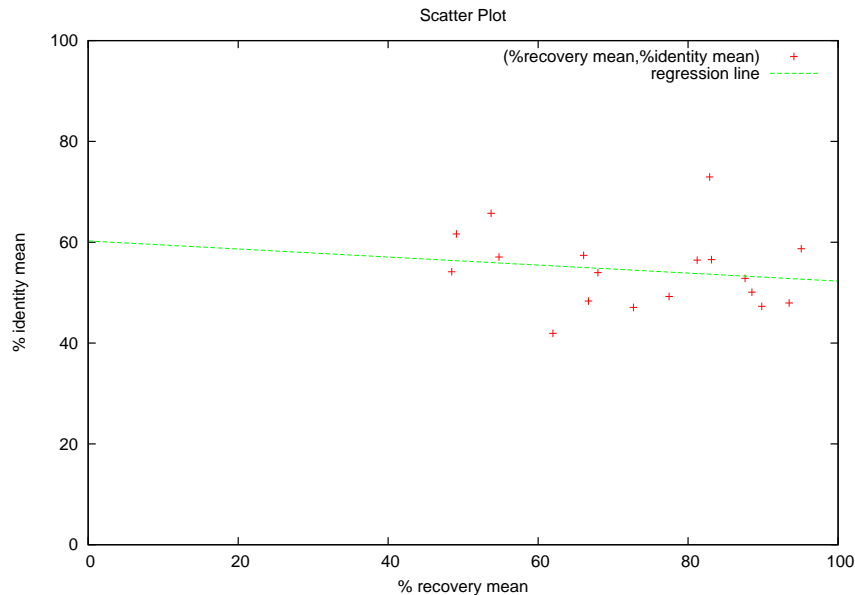


Figure 7: % Identity mean vs. % Recovery mean

7 Conclusion and Future Work

The future work proposed involves alternative techniques for sampling the sequences for the same experimental setup, using a different benchmark for running the tests and using a tool other than MAFFT to run the same experiments to test whether any pattern is observed across other alignment tools.

The first alternative is to use a different method to generate the input sequence subsets for each family. Currently, the sequences constituting each subset (from the same family) are picked at random from the parent file. The cardinality of the input subsets is the only parameter that is varied. Due to the inconclusive results for hypothesis 2, which we suspect, might have been influenced by the sampling technique, we propose a new sampling technique. The input subsets are created by starting with the entire PALI file and

iteratively removing one sequence at a time in order to create a subset. The choice of the sequence to be removed is determined by looking at the % identity between all the participating proteins. More precisely, a sequence in the pair with the smallest percentage identity/evolutionary distance is discarded at each iteration. The percentage identity for each pair of sequences can be easily maintained in a matrix. As a result of this sampling, we expect to see a uniform pattern in the resulting scatter plots of hypothesis 2. We expect the percentage identity to increase with the cardinality and be bounded above by the percentage identity of the parent file in all cases.

The second alternative is to use a different benchmark like BALiBase or SABmark. This could affect the percentage recovery score (hypothesis 1) as the scoring method used in different benchmarks are different.

The third alternative is to use a different software tool, other than MAFFT, in order to run the same set of experiments. The results of the comparative study might help draw conclusions on the way the alignment problem is handled across various tools.

Acknowledgment

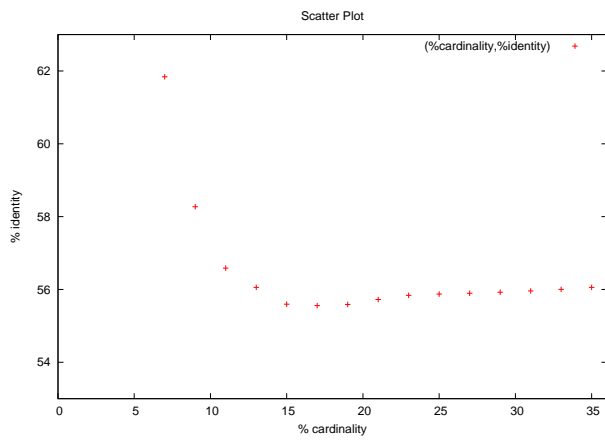
We would like to thank Dr. Snodgrass, Dr. Kececioglu and Travis Wheeler for all their help and invaluable input.

References

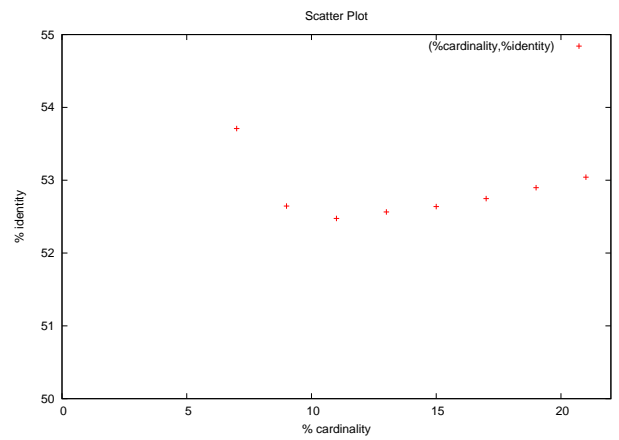
- [1] Dan Gusfield, **Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology**, 1999.
- [2] MAFFT version 6.237beta, Multiple alignment program for amino acid or nucleotide sequences. <http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/>
- [3] K. Katoh, K. Misawa, K. Kuma and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform." *Nucleic Acids Research* 30, 3059-3066, 2002.
- [4] J. Setubal and J. Meidanis, **Introduction to Computational Molecular Biology**, 1997.
- [5] S. Balaji, S. Sujatha, S.S. Kumar, N. Srinivasan, "PALI – a database of Phylogeny and ALIGNment of homologous protein structures." *Nucleic Acids Res.* 1;29(1):61-5, Jan 2001. <http://pauling.mbu.iisc.ernet.in/~pali/>

8 APPENDIX A:

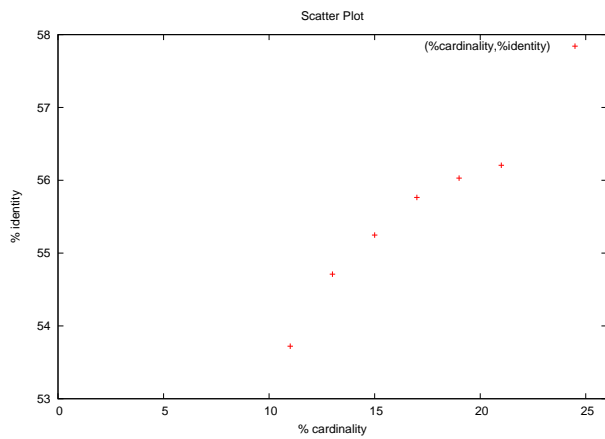
The following graphs are the scatter plots of Percentage Identity against Cardinality of the input subsets.



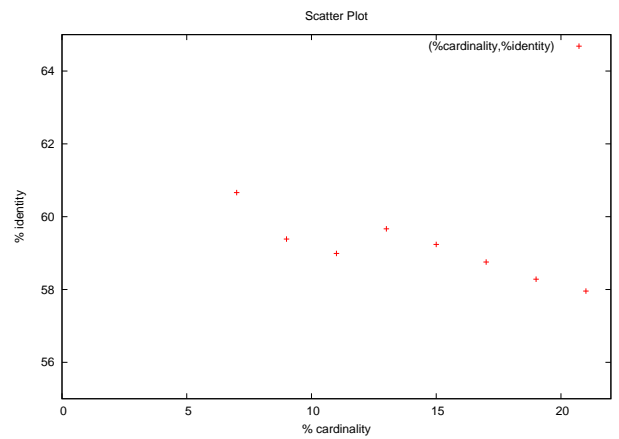
(a) Pali815 file



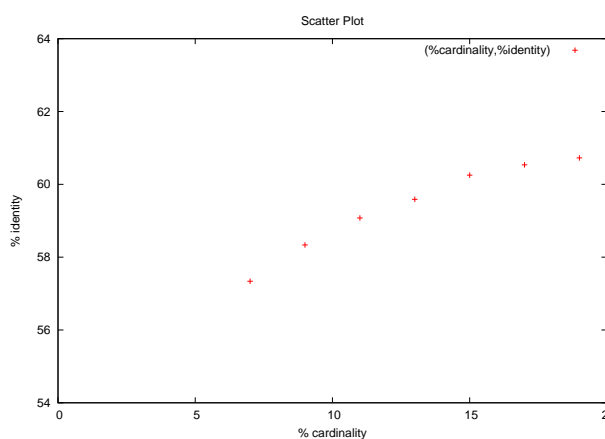
(b) Pali792 file



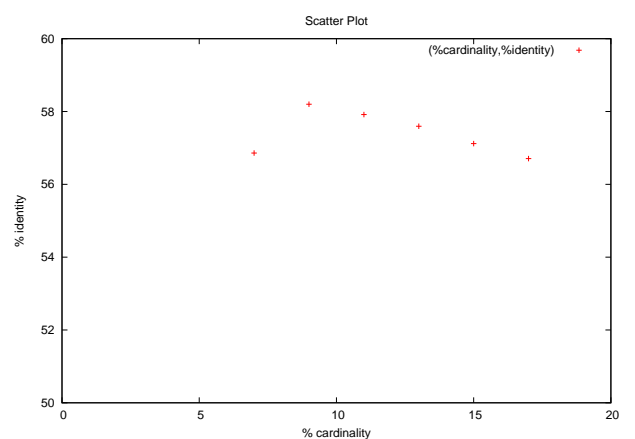
(c) Pali552 file



(d) Pali475 file

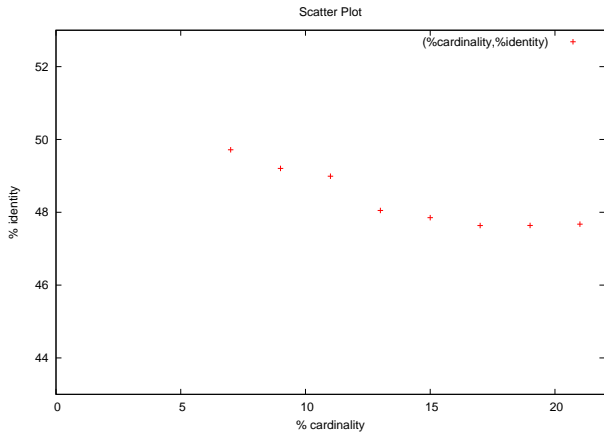


(e) Pali1539 file

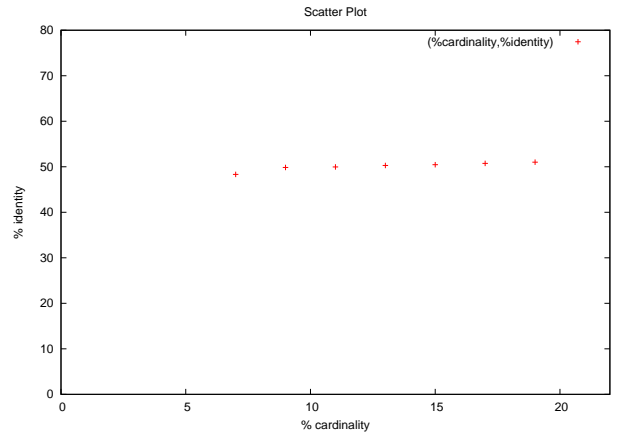


(f) Pali1486 file

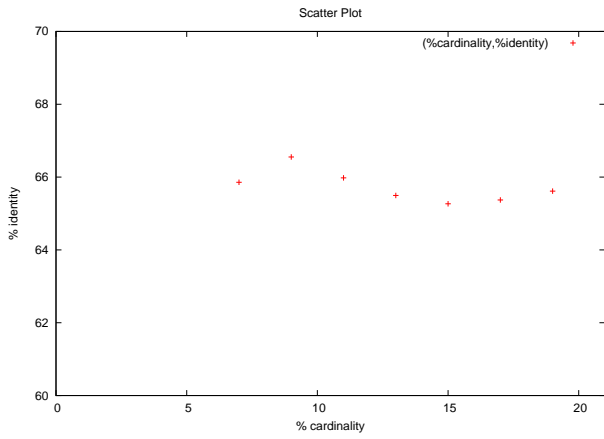
Figure 8:



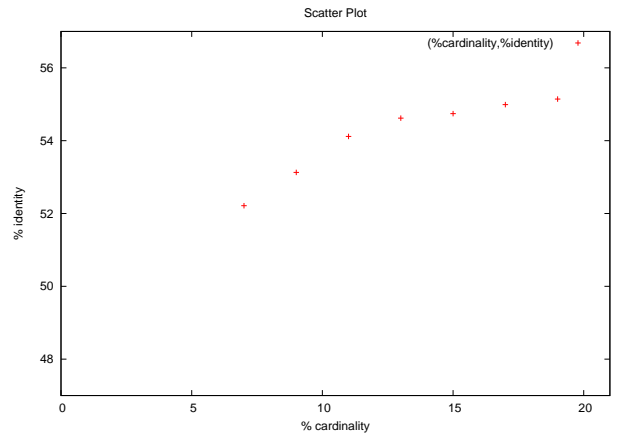
(g) Pali1392 file



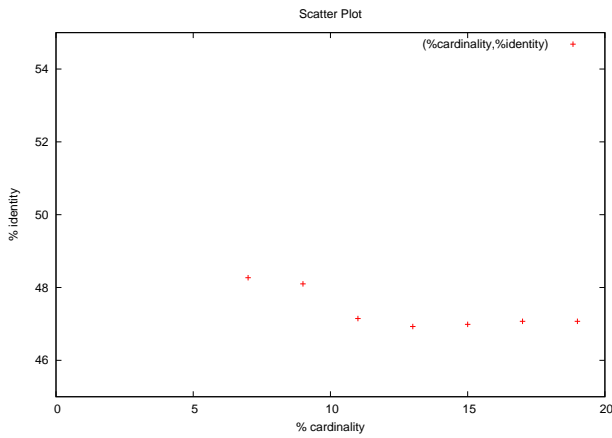
(h) Pali1254 file



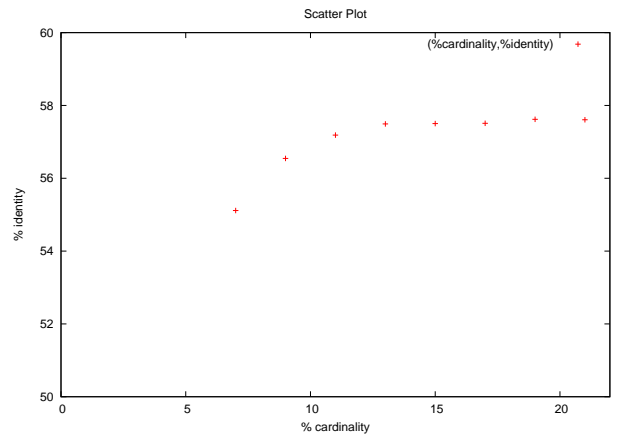
(i) Pali1093 file



(j) Pali1092 file

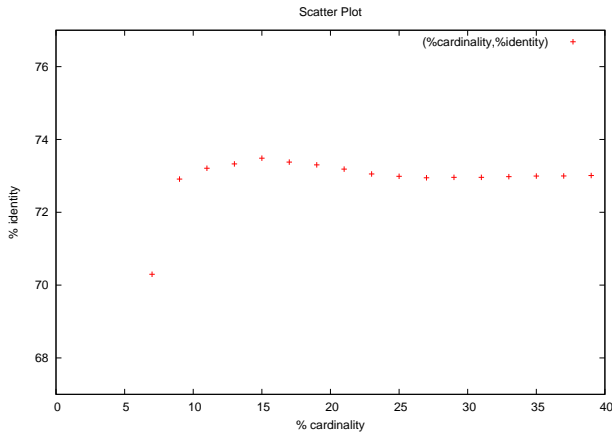


(k) Pali1250 file

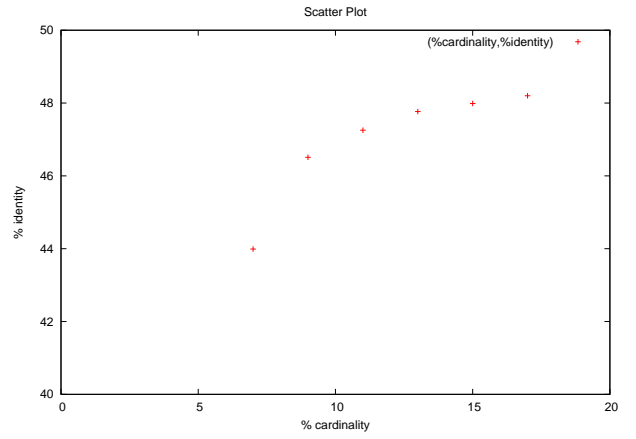


(l) Pali1360 file

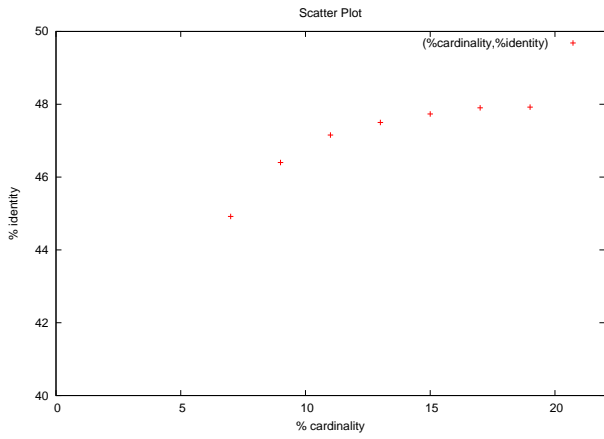
Figure 9:



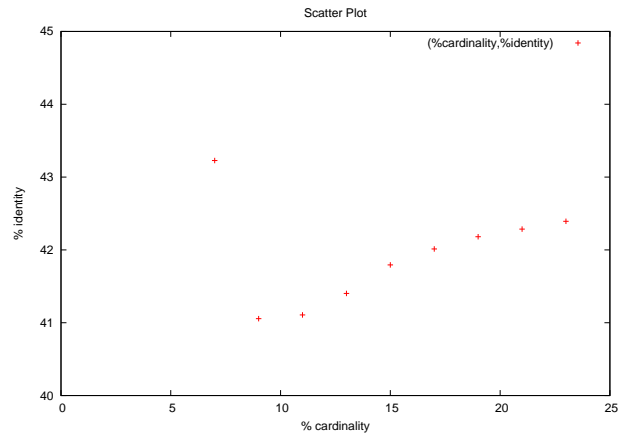
(m) Pali1444 file



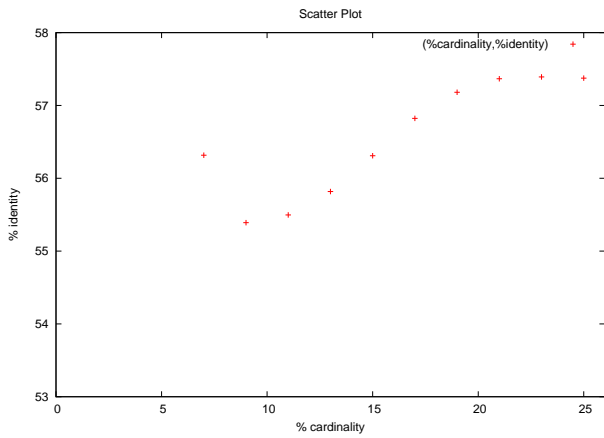
(n) Pali1521 file



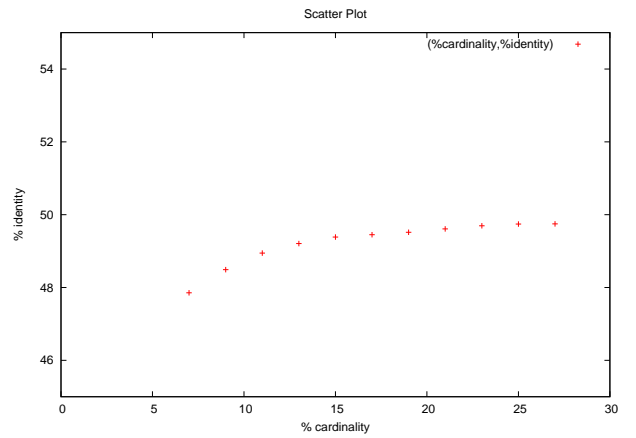
(o) Pali19 file



(p) Pali477 file



(q) Pali578 file

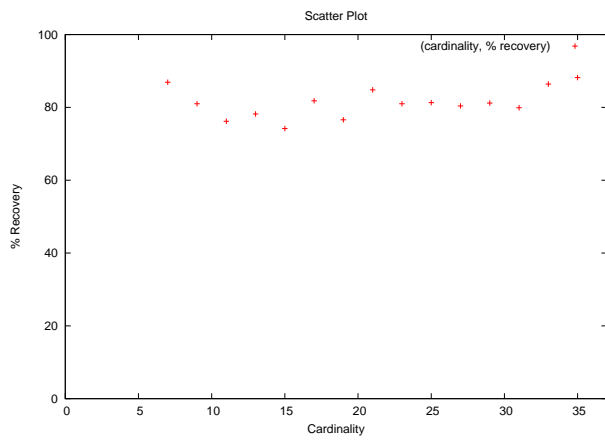


(r) Pali809 file

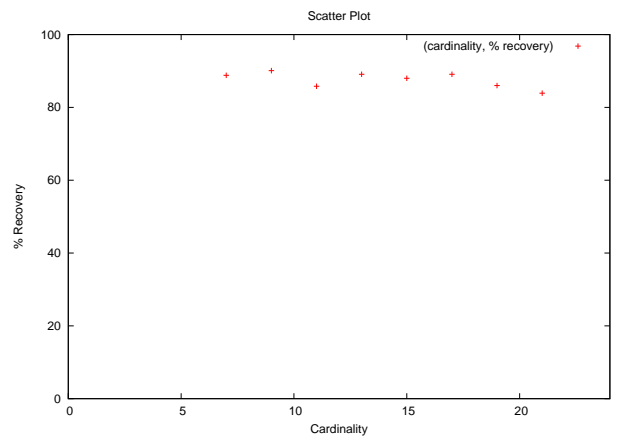
Figure 10:

9 APPENDIX B:

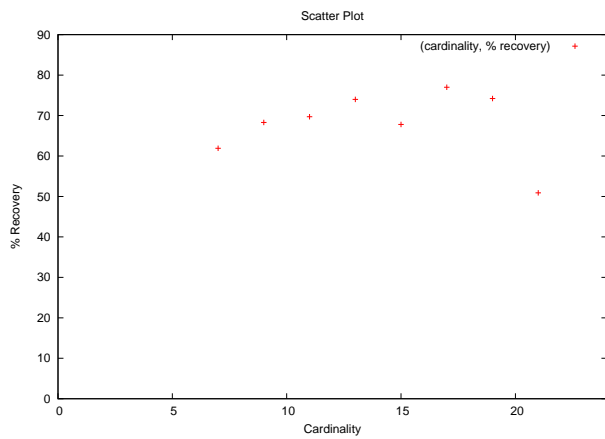
The following graphs are the scatter plots of Percentage Recovery against Cardinality of the input subsets.



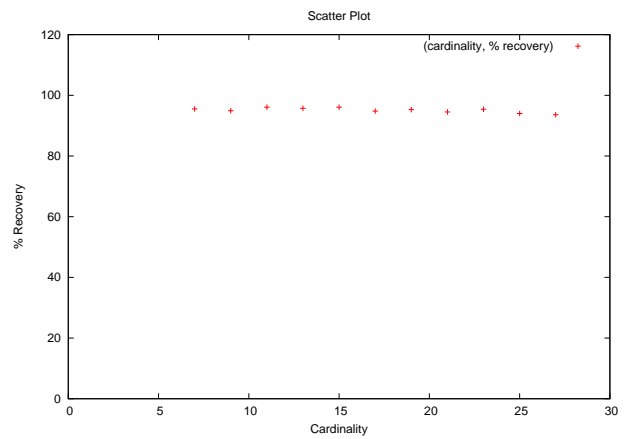
(a) Pali815 file



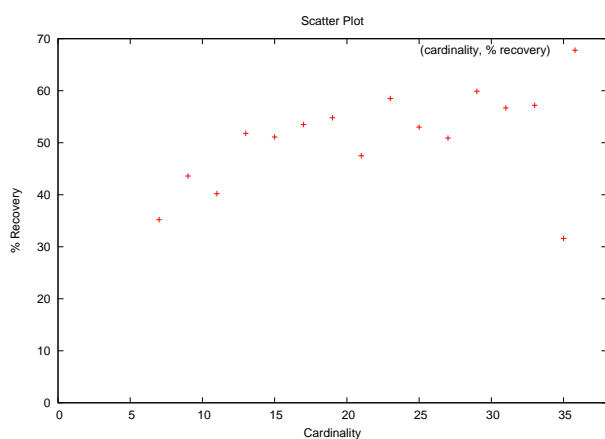
(b) Pali792 file



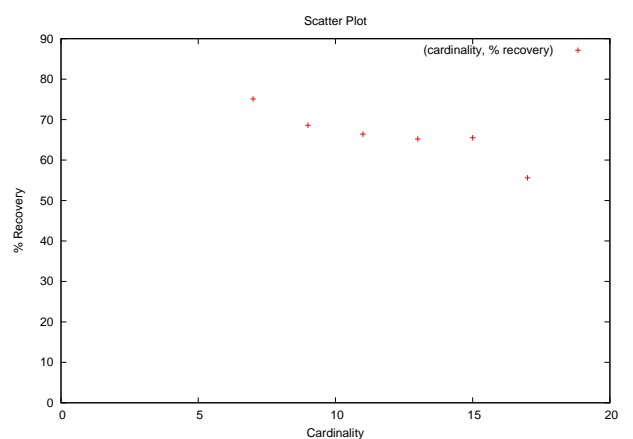
(c) Pali552 file



(d) Pali475 file

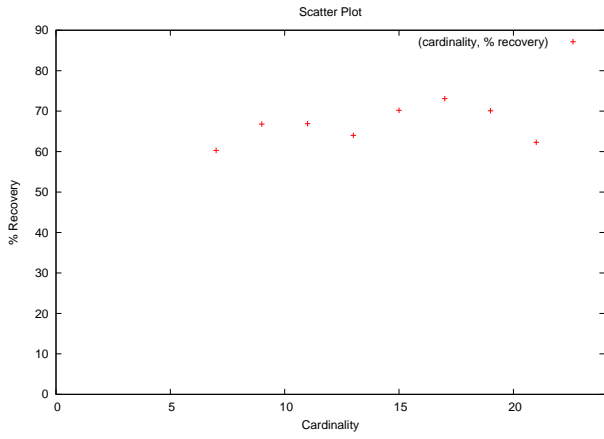


(e) Pali1539 file

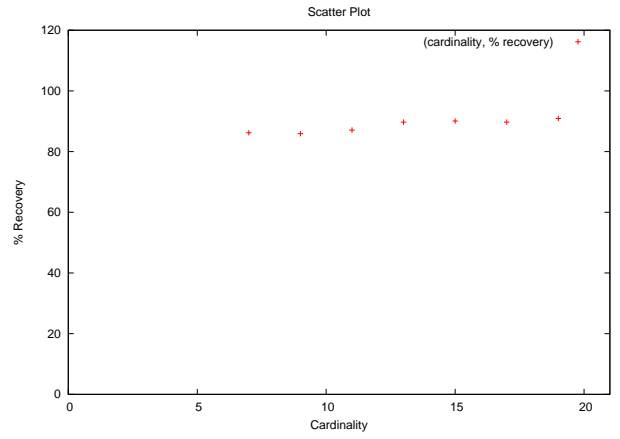


(f) Pali1486 file

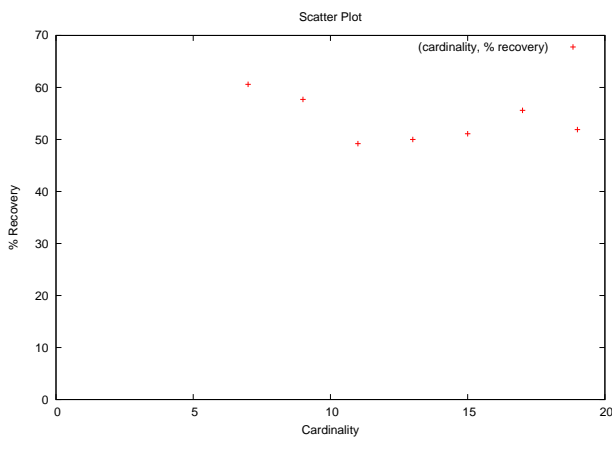
Figure 11:



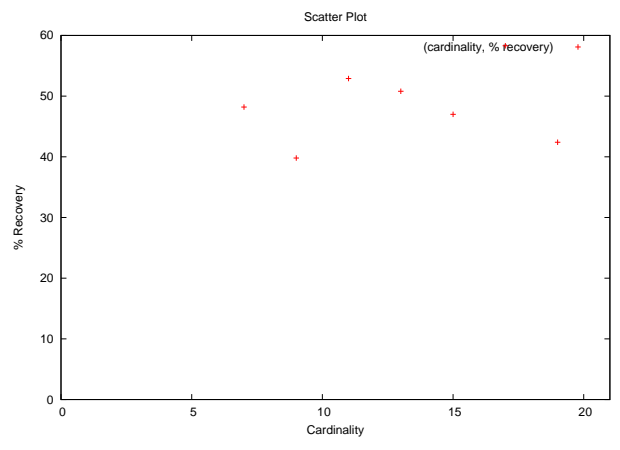
(g) Pali1392 file



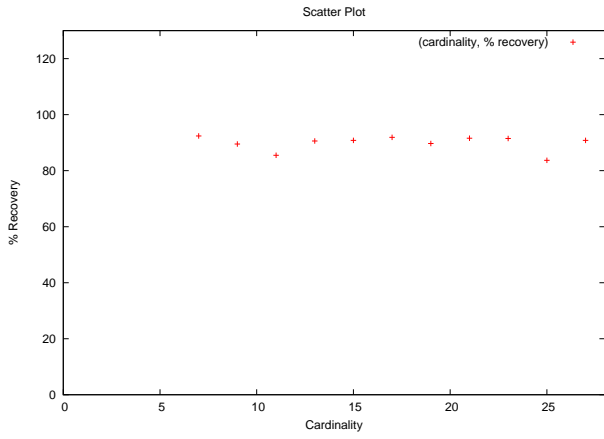
(h) Pali1254 file



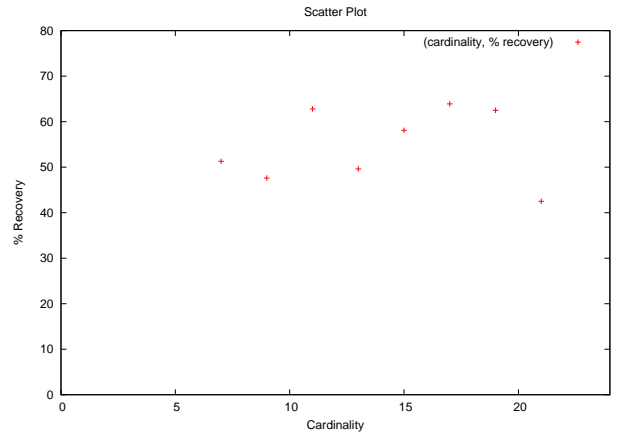
(i) Pali1093 file



(j) Pali1092 file

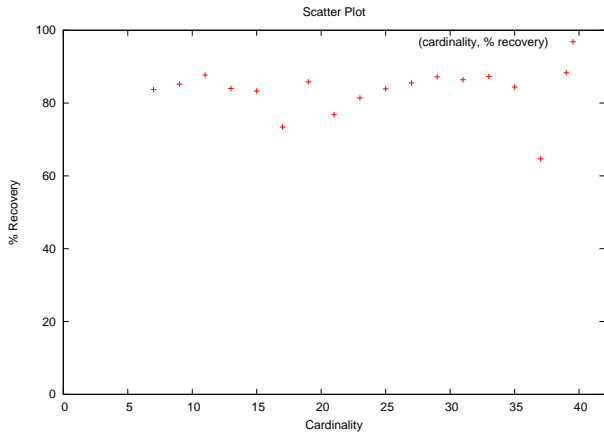


(k) Pali1250 file

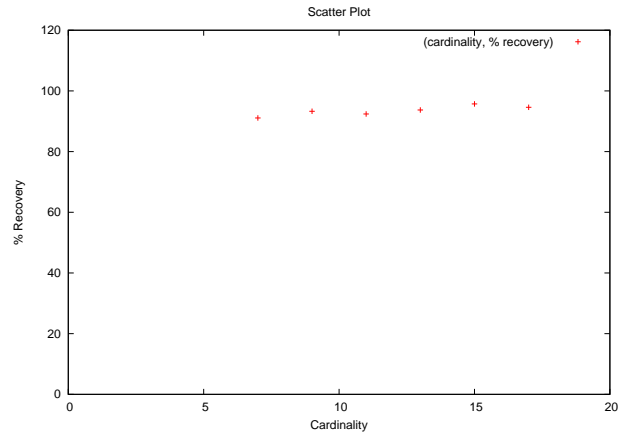


(l) Pali1360 file

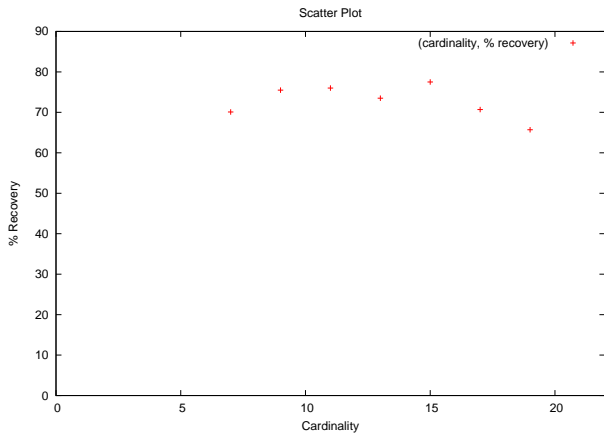
Figure 12:



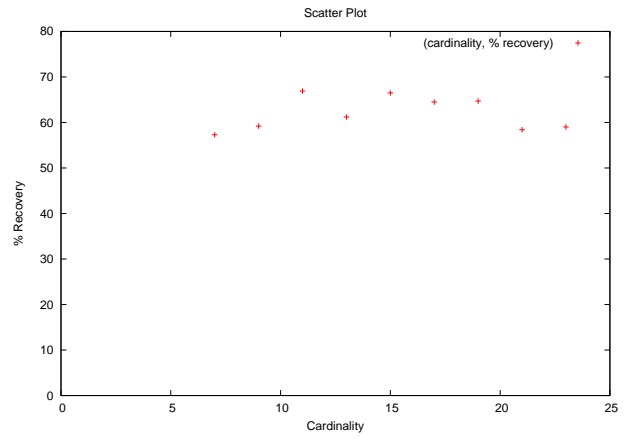
(m) Pali1444 file



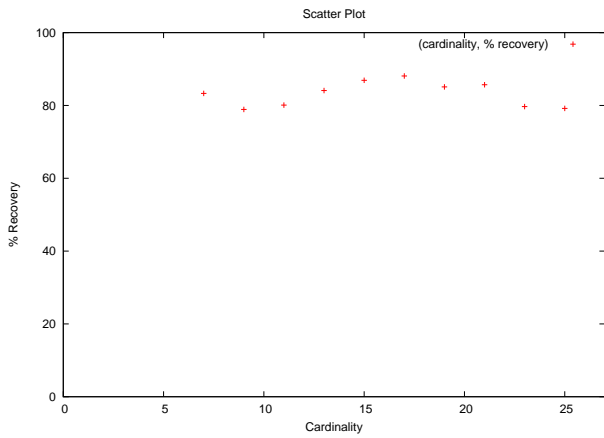
(n) Pali1521 file



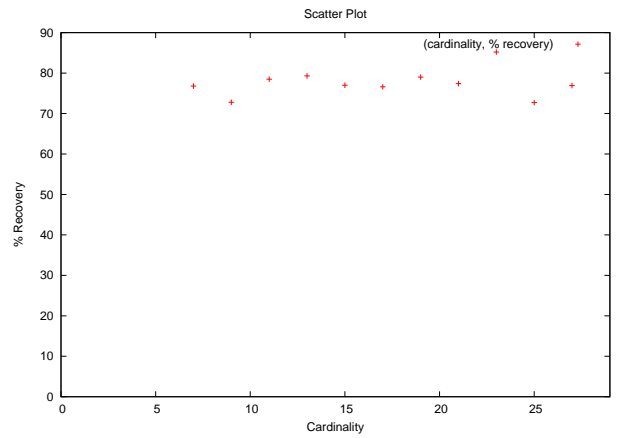
(o) Pali19 file



(p) Pali477 file



(q) Pali578 file



(r) Pali809 file

Figure 13: