# EVALUATING INDELS AS CHARACTERS OF BIOLOGICAL INFORMATIVENESS

by

Cory L. Strope

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Computer Science

Under the Supervision of Stephen D. Scott

Lincoln, Nebraska

December, 2009

# EVALUATING INDELS AS CHARACTERS OF BIOLOGICAL INFORMATIVENESS

Cory L. Strope, Ph.D.

University of Nebraska, 2009

Advisor: Stephen D. Scott

Reconstructing the evolutionary history of biological sequences will provide a better understanding of mechanisms of sequence divergence and functional evolution. Long-term sequence evolution includes not only substitutions of residues but also more dynamic changes such as insertion, deletion, and long-range rearrangements. Such dynamic changes make reconstructing sequence evolution history difficult and affect the accuracy of molecular evolutionary methods, such as multiple sequence alignments (MSAs) and phylogenetic methods. In order to test the accuracy of these methods, benchmark datasets are required. However, currently available benchmark datasets have limitations in their sizes and evolutionary histories of the included sequences are unknown. These are the serious drawbacks as benchmarks. Such problems can be solved by simulating sequences to create benchmark datasets with known evolutionary history. However, currently available simulation methods do not allow biologically realistic dynamic sequence evolution.

We introduced indel-Seq-Gen version 1.0 (iSGv1.0), a program that simulates realistic evolutionary processes of protein sequences with insertions and deletions (indels). iSGv1.0 allows the user to simulate multiple subsequences according to different evolutionary parameters, tracks all evolutionary events including indels and outputs the "true" MSA of the simulated sequences. With indel-Seq-Gen version 2.0 (iSGv2.0), we aimed at simulating evolution of highly divergent DNA sequences and protein superfamilies. iSGv2.0 adds lineage-specific evolution, motif conservation,

indel tracking, subsequence length constraints, and incorporates coding and non-coding DNA evolution. We uncovered a flaw in the modeling of indels used in current state of the art methods, and fixed it by using a novel discrete stepping procedure.

Finally, we developed a new MSA scoring metric called the *gap profile score* that utilizes insertion and deletion placements to evaluate MSA accuracy. Using a series of benchmark alignments created with iSGv2.0, we examined the performance of our scoring method against currently used character-based scoring metrics, including the sum of pairs score. We examined how well the scoring metric output correlates with accuracy of phylogenetic reconstruction. We show that the gap profile score opens a novel way to gauge the efficacy of MSA reconstructions, potentially opening the door to the research of better models of indel placement into MSA reconstruction methods.

# DEDICATION

To my mom, Connie Strope, whose support during a time of self-doubt early in my graduate career helped me persevere to the completion of this journey.

# ACKNOWLEDGMENTS

I cannot overstate the gratitude I have for my advisors, Dr. Stephen Scott and Dr. Etsuko Moriyama, who provided invaluable assistance in all phases of my graduate career. Without their valuable insights into my research and their help in clarifying my writing and presentation techniques, I would have been unable to complete the graduate rollercoaster.

I would also like to thank the rest of my committee, Dr. Khalid Sayood and Dr. Jitender Deogun, for their helpful suggestions to improve my dissertation.

I am forever indebted to Dr. Russell Mosemann, whose belief in my abilities as a student both encouraged me to excel, and whose endorsement of me also lead to my acceptance into the graduate program.

I owe a debt of gratitude to Dr. Don Costello for helping me to obtain the Graduate Assistance in Areas of National Need fellowship, which allowed me to concentrate on my research by reducing my teaching responsibilities.

I would like to thank the members of Moriyama lab, who provided a fun and stimulating environment, and who were particularly helpful (and patient) in improving oral presentations.

I am indebted to the staff of the Computer Science Department, Deb Heckens, Shelley Everett, Marilyn Augustyn, and Sally Hawkins, who made the path from acceptance into the graduate program to the completion of my dissertation smooth, along the way providing assistance in obtaining financial support, simplifying the process of conference travel, and always being around to lend assistance.

I also want to express my deepest gratitude to Dr. Charles Riedesel, who not only gave me a large amount of his time to help me study for my qualifying exams, but also allowed me the opportunity to be a full instructor for several classes in the

Computer Science department.

This dissertation would not have been possible without the love and support of my family. I am particularly thankful to my wife, Pooja, and daughter, Adishree, who provided emotional support and taught me what is truly important during stressful times.

# Contents

**Contents** **7**

**List of Figures** **12**

**List of Tables** **23**

**1 Introduction** **25**

1.1 Background . . . . . 27

1.1.1 Multiple Sequence Alignment . . . . . 27

1.1.1.1 Progressive Alignment: . . . . . 27

1.1.2 Benchmarking Multiple Sequence Alignments . . . . . 33

1.1.2.1 Scoring Multiple Sequence Alignment Accuracy . . . 33

1.1.2.2 Benchmark Datasets . . . . . 35

1.1.2.3 Benchmark Dataset Issues . . . . . 40

1.2 Protein Family . . . . . 40

1.2.1 Protein Structures . . . . . 40

1.2.2 Protein Families Used in this Study . . . . . 41

1.2.2.1 Class A G Protein-Coupled Receptors . . . . . 42

1.2.2.2 Calycin/Lipocalin Superfamily . . . . . 42

1.3 Sequence Simulation . . . . . 43

1.3.1 Sequence Simulation Input . . . . . 43
1.3.2 Simulation Method Background . . . . . 47
1.4 Performance Analysis Concepts Used . . . . . 52
1.4.1 Sensitivity and Specificity . . . . . 53
1.4.2 Receiver Operating Characteristic Curve . . . . . 53
1.5 Incorporating Indel Information in MSA . . . . . 55

**2 indel-Seq-Gen version 1.0: Protein Family Simulator Incorporating Domains, Motifs, and Indels 58**

2.1 Method . . . . . 59
2.1.1 Protein Sequence Evolution . . . . . 59
2.1.2 Protein Family Creation . . . . . 59
2.1.2.1 Motif Conservation . . . . . 60
2.1.2.2 Heterogeneous Evolution among Domains . . . . . 61
2.1.3 Indel Event Handling . . . . . 61
2.1.3.1 Probability of Indels . . . . . 61
2.1.3.2 Placement of Indels . . . . . 62
2.1.3.3 Length Distribution of Indels . . . . . 62
2.1.3.4 Insertion Sequence Options . . . . . 63
2.1.4 Root Sequence Options . . . . . 63
2.1.5 Implementation . . . . . 65
2.1.6 Comparison of Simulation Methods . . . . . 65
2.1.6.1 Transmembrane Region Prediction . . . . . 65
2.1.6.2 Beta-Strand Prediction . . . . . 65
2.1.6.3 Blast Similarity Search . . . . . 66
2.1.6.4 Pfam Search . . . . . 66

2.1.6.5 Parametric Bootstrap and Phylogenetic Analysis . . 66
2.2 Results and Discussion . . . . . 67
2.2.1 Simulation Setup . . . . . 67
2.2.1.1 Template MSA . . . . . 67
2.2.1.2 Setting Parameters . . . . . 70
2.2.1.3 Comparison of Simulated Sequences between iSGv1.0 and Other Methods . . . . . 74
2.2.1.4 Simulated Sequences . . . . . 75
2.2.1.5 Conservation of Transmembrane Regions . . . . . 77
2.2.1.6 Beta-Region Prediction . . . . . 77
2.2.1.7 Blast and PFAM Search Results . . . . . 79
2.2.1.8 Phylogenetic Analysis with Parametric Bootstrap . . 81
2.3 Conclusion . . . . . 85

**3 Biological Sequence Simulation for Testing Complex Evolutionary Hypotheses: indel-Seq-Gen version 2.0 86**

3.1 Methods . . . . . 87
3.1.1 Discrete Evolution . . . . . 88
3.1.1.1 Substitution and indel models . . . . . 89
3.1.1.2 Formalization of substitution and indel processes . . 90
3.1.1.3 Novel indel characterization . . . . . 91
3.1.1.4 Simulating indel occurrence . . . . . 91
3.1.2 Lineage-specific evolution . . . . . 93
3.1.3 Functional constraint modeling . . . . . 93
3.1.3.1 Site-specific constraints . . . . . 93
3.1.3.2 Subsequence length constraints . . . . . 94

3.1.4 Nucleotide sequence simulation . . . . . 98
3.1.5 Event tracking . . . . . 98
3.1.6 Implementation . . . . . 102
3.1.7 Indel simulation comparison . . . . . 102
3.1.7.1 Experimental setup . . . . . 103
3.1.8 Protein superfamily comparison . . . . . 104
3.1.8.1 Experimental setup . . . . . 107
3.1.8.2 Indel statistics . . . . . 108
3.2 Results and Discussion . . . . . 108
3.2.1 Test 1: Insertions alone or deletions alone . . . . . 108
3.2.2 Test2: Including both insertions and deletions with various $P_{ins}$ and $P_{del}$ . . . . . 113
3.2.3 Example application for a protein superfamily simulation . . . 117
3.3 Conclusion . . . . . 120

**4 Gap Profiling** **121**

4.1 Methods . . . . . 123
4.1.1 Construction of Benchmark Datasets . . . . . 123
4.1.2 Gap Profile Score . . . . . 126
4.1.2.1 Profiling Gaps . . . . . 126
4.1.2.2 Sensitivity and Specificity . . . . . 126
4.1.2.3 Calculating Gap Profile Scores . . . . . 128
4.1.3 MSA Scoring Technique Comparison . . . . . 130
4.1.3.1 Scoring Method Implementation . . . . . 133
4.1.3.2 Scoring Metric Comparison . . . . . 133
4.1.4 Multiple Sequence Alignment Programs . . . . . 134

4.1.4.1 Statistics Calculation . . . . . 134
4.1.5 MSA Scoring Methods versus Phylogenetic Accuracies . . . . 136
4.2 Results and Discussion . . . . . 136
4.2.1 Scoring Method Comparison . . . . . 137
4.2.1.1 Scoring Method Inconsistencies . . . . . 137
4.2.1.2 Independence of Measured Values . . . . . 141
4.2.2 Gap Profile and $Q$ scores versus Phylogenetic Reconstruction . 152
4.3 Conclusion . . . . . 156

**5 Conclusion** **158**

**Bibliography** **162**

**A indel-Seq-Gen version 1.0 Supplementary Material** **175**
A.1 Template Alignments . . . . . 175
A.2 Tree file inputs to indel-Seq-Gen . . . . . 187

**B Gap Profiling Supplementary Material** **192**
B.1 Scoring Methods Colored by Benchmark Parameter Values . . . . . 192
B.2 Scoring Methods Side-by-side Comparison . . . . . 303
B.3 Metric Scores against Nucleotide Phylogeny Reconstruction . . . . . 442

# List of Figures

1.1 An example guide tree for the progressive alignment of a set of sequences T1–T5. (A, B) Nodes containing profiles built from pairwise sequence alignments. (C) Node containing a profile built from a profile–profile alignment. (D) Node containing a profile built from a sequence–profile alignment. Progressive alignment begins by aligning A (T1/T2) and B (T2/T3), followed by C (A/B, or T1/T2/T3/T4), and D (root, C/T5). . 28

1.2 Examples of the true MSA compared to the MSA reconstructed by ClustalW2.0, MAFFT, Muscle3.7, and ProbCons. Each black pixel represents a character in the sequences and each white pixel is a gap in the alignment (the result of an insertion or deletion). The true MSA was generated using indel-Seq-Gen version 2.0 (iSGv2.0) [86]. The parameters used to simulate sequences are as follows: 16 taxa, root-to-tip tree length of 0.75 substitutions per site, $P_{ins} = P_{del} = 0.1$, indel length of 5 characters. The guide trees used for the simulation are shown at the top (A: balanced, B: pectinate). iSGv2.0 is described in Chapter 3. . . . . 34

1.3 The canonical structure of a GPCR sequence. Image from [56]. . . . . 42

1.4 The calycin superfamily tree. Calycins are a superfamily of soluble, antiparallel $\beta$-barrel protein subfamilies. The $\beta$-barrel conformation is such that each $\beta$-strand interacts with its immediate neighboring strands with respect to primary sequences, with the exception of the interacting first and last $\beta$-strands. The members of the Calycin superfamily are the fatty-acid binding proteins (FABPs), avidins, metalloproteinase inhibitors (MPIs), triabins, and the lipocalins. The lipocalins are also a superfamily of proteins that consist of two subfamilies, the kernel lipocalins and the outlier lipocalins. The structure of each family is represented by a cartoon, where the triangles represent $\beta$-strands (triangle direction represents the direction of the sequence, from N- to C-terminus), circles representing $\alpha$-helices, and lines representing random coil regions. Image from [29]. . . 44

1.5 The canonical structure of a lipocalin protein. Eight $\beta$-strands (A–H) make up the barrel region of the lipocalin, with three structurally conserved regions (SCR1–SCR3) imparting specific functions to the lipocalin members. Kernel lipocalins contain all three SCRs, while outlier lipocalins contain either 1 or 2 of the SCRs. Image from [29]. . . . . 45

1.6 Two examples of bifurcating trees: (A) the balanced tree, and (B) the pectinate tree. . . . . 46

1.7 A comparison of the basic simulation paradigm and more realistic biological sequence evolution. (A) Substitution ($\Theta$) and indel ($\Lambda$) parameters used for simulation methods. (B) The basic simulation paradigm. Given a root sequence and a global set of substitution and indel parameters ($\Theta_G$ and $\Lambda_G$, respectively), simulation proceeds by applying changes in a Monte Carlo manner over all sequence positions, following the guide tree and ending with a set of sequences, which are called as operational taxonomic units (OTUs). (C) More realistic biological sequence evolution. Starting with a root sequence and an initial level of substitution and indel parameters ($\Theta_G$ and $\Lambda_G$, respectively), evolution at sites may become constrained by gaining a functional motif (the gray box shown in the gray lineage), and the substitution and indel parameters may be changed ($\Theta_L$ and $\Lambda_L$), or the initial parameters are maintained without gaining any functional motif as shown with $\text{OTU}_X$. . . . . 47

1.8 An example of the Receiver Operating Characteristic (ROC) curve for five binary classifiers. Classifier D is a discrete classifier, producing only a single point in the ROC space. Conversely, classifiers A, B, C, and E rank test examples by assigning a probability, a numeric value that represents the confidence that a particular instance belongs to the predicted class. . 54

2.1 A flow chart showing the process of iSGv1.0 protein sequence simulation. In this example, parameterization of evolutionary information is done based on the vertebrate olfactory receptor family. Steps 1–5 illustrate the process of obtaining the template MSA, phylogeny, and various parameters. Step 6 shows the sample input for iSGv1.0 with each line showing parameters used for a different subsequence (domain). See Figure A.2 for the example input file. Two evaluation methods, phylogenetic analysis and alignment comparison using profile HMMs, are shown in dashed boxes. 68

2.2 MSAs of the template and simulated GPCR sequences. Each pixel represents one position in the MSA. The color of the pixel represents: a gap (white), an amino acid from a transmembrane region (gray), and an amino acid not from a transmembrane region (black). The template alignment (a) includes 29 GPCR sequences, and the 15 subsequence regions are indicated above. For 5 simulated data sets produced by iSGv1.0 (b) and ROSE (c), the true MSAs and MSAs reconstructed by T-Coffee are shown. 69

2.3 Distribution of predicted beta-strands along the lipocalin sequences. (a) Proportions of predicted beta-strands are plotted for each amino acid position of the alignments. For the reference alignment, the proportions are based on 23 lipocalin sequences. For simulated sequences, average proportions were plotted based on 5 simulated data sets using their true alignments obtained from each simulation method. Seven regions are mainly gaps with amino acids inserted into a few sequences (less than half the sequences have an amino acid). The proportions of predicted beta-strands in these 7 regions are represented by a single dot. These regions correspond to those in which indels are allowed (3, 7, 9, B and F as well as the start and end regions in the reference alignment, see the region labels in b). For each method, the average numbers of positions predicted as beta-strands are calculated from subsequence regions simulated as beta-strands (regions 2, 4, 6, 8, A, C, E, G in b) and coils (regions 1, 3, 5, 7, 9, B, D, F, H in b). These values are shown as mean-beta and mean-coil, respectively, in each plot. (b) The reference lipocalin alignment obtained from Sánchez et al. [80] is schematically shown with a single pixel representing 1 amino acid. The gray pixels represent beta-strands. The alignment is 213 positions long and consists of 23 lipocalins. Seventeen regions are labeled 1 through H. (c) Twelve subsequence regions used for simulation. The region $B_{5678}$ concatenated short consecutive regions from 9 to G. . . . . 71

2.4 Comparisons of the template alignment and the true alignments generated by indel-Seq-Gen (a) and ROSE (b). Comparisons were done by using COACH [24]. The profile hidden Markov model was generated from the template alignment, and compared against each of 1000 simulated datasets. A Viterbi score was calculated from each comparison. The distributions are a): $\mathcal{N}(75.03, 17.20)$ and b): $\mathcal{N}(72.30, 16.72)$, respectively. . 75

2.5 The distribution of the number of predicted transmembrane regions for the olfactory receptor datasets simulated by Seq-Gen, indel-Seq-Gen, and ROSE using HMMTOP 2.0. For each method, five datasets (including 29 sequences) are used in this analysis. . . . . 78

3.1 The continuous and discrete-steps models of indel events. The continuous model calculates the expected number of indel events based on sequence length at node $i$, and uses this same value throughout the branch length, $\mathrm{BL}_{i \to i+1}$. This causes either over- or underestimating the number of indels along the branch until recalculating the expected number of indel events at node $i + 1$. The discrete-steps model reduces the impact of this by recalculating the expected number of indel events based on the sequence length after each such event. . . . . 89

3.2 A sequence model that includes substitutions (Sub), insertions (Ins), and deletions (Del) for a length-constrained subsequence $S$ ($X_0 \cdots X_7$, where $X_i$ is the $i$th residue of the sequence). The description of the symbols used and their effects is listed in the table below. For example, positions $X_3 \cdots X_7$ are conserved in a way akin to the $CXXC$ motif of the Thioredoxin sequence motif [16]. The maximum lengths of insertions are shown above the sequence ranging from 0 (no insertion), 1, 2, ... to $i_S$ (upper bound of the subsequence length). The maximum lengths of deletions are determined by either (1) the number of sequence positions to the first deletion-constrained position (2 for $X_i$ in this figure) or $d_S$, defined as the number of positions that can be deleted before reaching the minimum subsequence length. . . . . 92

3.3 The MSA input used for the calycin superfamily simulation with iSGv1.0 and iSGv2.0. Yellow highlighted regions in the MSA are beta strands, gray highlighted regions are alpha helices, and unhighlighted regions are coils. Conserved regions (PS00213, SCR1, SCR2, PS00577, and SCR3) are marked above the alignment, and the characters of the sequences corresponding to the motifs are shown in the same colors as the headings. The quaternary invariable array of iSGv1.0 (*iSGv1.0-inv*) and the iSGv2.0 template (*iSGv2.0-templ*) are listed below the alignment. Below the template line, x(min,max) shows the maximum and minimum sizes for each region. The conserved secondary structure regions (beta1 - beta8 and alpha1) corresponding to these regions are also marked below the template line. . . . . 97

3.5 The four simple guide trees and their corresponding Newick formats used to test indel simulation schemes. These have 0, 1, 3, and 7 branching points for Trees 1, 2, 3, and 4, respectively. Note that at each branching point (or node), one branch is given a zero length branch, as shown in the Newick format. The total length of the guide tree is set to 8 substitutions per site. Branching points are named from "node0" (at the root) to "node8." During the simulation, sequences are saved at each internal node as well as terminal nodes, and used for indel analysis. . . . . 104

3.7 Indel simulation performance among the seven methods (Test 1). Correct simulations are expected to produce a plot with a horizontal line. (A) Size-1 deletions, (B) size-2 deletions, (C) size-4 deletions, (D) size-8 deletions, (E) size-1 insertions, (F) size-2 insertions, (G) size-4 insertions, (H) size-8 insertions. The test statistics (characters left for deletion-only tests and true alignment lengths for insertion-only tests) and the coefficients of variation are shown in Table 3.1. The standard deviations for each data point are shown in Table 3.2. The average values obtained from 100 simulations are plotted. . . . . 110

3.8 Test 2 results with different insertion/deletion probability ratios. For each method, the total numbers of insertions (dark bars) and deletions (light bars) generated are shown for simulation experiments using the guide trees with different numbers of segments (see Figure 3.5. Note that for MySSP we used the average expected value of the Zipfian distribution to obtain the results. For all methods, the average values obtained from 100 simulations are used. . . . . 115

4.1 Counting the two gap profiles for the pairwise alignment of sequences $S_i$ and $S_j$ from the MSA $M$. . . . . 127

4.2 The $N \times N$ gap profiles that explain gapping pattern for the MSA $M$. . 127

4.3 Calculating specificity and sensitivity. (A) The gap profiles for $M^{True}$ $M^{Recon}$ for sequence $S_i$ when compared to $S_j$ ($GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$). (B) Counting the distance of gaps from $GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$ for sensitivity and specificity calculation. Positions of the search space correspond with (A). A distance $d = 0$ starts where the gap is found. If the corresponding position in the comparison gap profile is not one, $d$ is incremented, and the gap profile values at positions $x \pm d$ are checked, and so on until a corresponding position either contains a value of one or $d = w$. Numbers in italics denote that a corresponding gap has been found, ending the search. (C) The sensitivity and specificity arrays after finishing the comparison between $GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$. . . . . 129

4.4 Comparison of the gap profile scores with respect to the maximum gap distance ($w$) for (A) ClustalW2.0, (B) MAFFT-linsi, (C) Muscle3.7, and (D) ProbCons. *Left*: Sensitivity versus specificity curves for different window sizes. Note that maximum gap distances of 500 and 1000 may not be visible, as they return near perfect scores, and as such are superimposed with the axes. *Right*: gap profile score for maximum gap distances 1...50, 100, 250, and 500. The represented benchmark condition of this example is a balanced tree of 16 taxa and a length of 0.75 substitutions per site, $P_{ins} = P_{del} = 0.1$, with indel lengths equal to 5. . . . . 132

4.5 Categories of MSA-scoring metrics. . . . . 132

4.6 Counting topological accuracy. The true tree is the guide tree used for simulating the dataset. The inferred tree is the maximum likelihood tree inferred from reconstructed multiple sequence alignments. Two internal branches, "branch 1" and "branch 2," are non-trivial to infer: branch 1 creates the bipartition {{`Taxon1, Taxon2`}, {`Taxon3, Taxon4, Taxon5`}} in both the true and inferred tree, while branch 2 creates an incorrect bipartition of {{`Taxon1, Taxon2, Taxon4`}, {`Taxon3, Taxon5`}} in the inferred tree compared to the true bipartition {{`Taxon1, Taxon2, Taxon3`}, {`Taxon4, Taxon5`}}. Thus, the topological accuracy for this example is 0.5 (one correct branch divided by two internal branches). . . . . 137

4.7 Comparison of the SPS scores from the `baliscore` ($x$-axis) and `qscore` ($y$-axis) packages. Alignment dataset points are plotted based on the four different indel sizes used to create the benchmark alignments: indel sizes 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡'). . . . . 138

4.8 Relationships between the length of contiguous indels and the SPS values in scoring alignments by different MSA reconstruction methods. *Left:* The $Q$ score metric. *Right:* The `baliscore` SPS metric. Alignment dataset points are plotted based on the four different indel sizes used to create the benchmark alignments: indel sizes 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡'). . . . . 140

4.10 Comparison of the Total Column scores from the `baliscore` and `qscore` packages, with respect to (A) indel size: 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡') and (B) tree length: 0.25 (red '+'), 0.5 (green 'x'), 0.75 (blue '*'), and 1 (purple '⊡'). . . . . 143

4.11 Comparison of scoring method performances on the benchmark dataset for ClustalW2.0 (top left), MAFFT-linsi (top right), Muscle3.7 (lower left), and ProbCons (lower right). (A) Gap profile score vs. $Q$ score. (B) gap profile score vs. $f_M$ score. (C) gap profile score vs. total column score. (D) $Q$ score vs. $f_M$ score. (E) gap profile score vs. shift score. . . . . 147

4.12 Comparison of the total column score ($\text{bali}_{TC}$) and the $Q$ score ($\text{q}_Q$). . . 151

4.13 Comparison of the gap profile scores using different $w$ values. $w = 10$ vs. $w = 25$. . . . . 154

4.14 Comparison between the $\text{GP}_5$ score (red '+') and the $\text{q}_Q$ score (green 'x') versus the phylogenetic reconstruction accuracy for the MSAs reconstructed by different methods. . . . . 155

A.1 The template multiple alignments of the 28 vertebrate olfactory receptors (plus one outgroup sequence OPSD_BOVIN; a.) and the 23 lipocalin sequences (b). On top of each alignment the invariable arrays used with indel-Seq-Gen are shown. The GPCR invariable array contains the motif patterns (shown with '1', '2', and '3') identified by Fuchs et al. [30]. . . . 186

A.2 The input file that gives the specification of each subsequence of the (a) vertebrate olfactory-receptor and (b) lipocalin template multiple alignments. For a detailed description of each of the options. . . . . 191

B.1 Scoring metric scores versus benchmark statistic. Color of data points correspond to the benchmark parameter. . . . . 302

B.2 Side by side comparison of scoring methods, colored by benchmark statistics. 441

B.3 Comparison of scoring metric values against phylogenetic topological accuracy using Nucleotide reconstructed MSAs. . . . . 446

# List of Tables

1.1 A comparison of sequence simulation methods. . . . . 51

2.1 Parameter settings for indel-Seq-Gen, ROSE, and SIMPROT used in this study. . . . . 73

2.2 The BLAST and PFAM statistics of vertebrate olfactory receptors when using simulated sequences from iSGv1.0, Seq-Gen, SIMPROT and ROSE as the query sequence. . . . . 80

2.3 The BLAST and PFAM statistics of lipocalins when using simulated sequences from iSGv1.0, Seq-Gen, SIMPROT and ROSE as the query sequence. . . . . 82

2.4 Branch support range for the phylogenetic trees obtained by parametric bootstrap using iSGv1.0 simulated datasets of GPCRs and lipocalins. . . 84

3.1 The data points and coefficient of variation for the plots in Figure 3.7. . 111

3.2 Standard deviations for each data point in Figure 3.7. . . . . 112

3.3 The effects of internal-node numbers with varying insertion and deletion rates among methods. . . . . 116

3.4 Performance comparison among iSGv1.0, ROSE, and iSGv2.0 for the calycin superfamily simulation. . . . . 118

4.1 List of parameters and reconstruction methods used for dataset creation. 124

4.2 Comparison of balanced and pectinate trees, as shown in Figure 1.6. . . . 125

4.3 Statistics gathered from the alignments generated by different methods for each dataset. . . . . 135

4.4 The average root mean square deviation values for scoring metrics on reconstructed MSAs. . . . . 148

# Chapter 1

# Introduction

Generating multiple sequence alignments (MSAs) and reconstructing phylogenetic relationships from biological sequences are frequently the most important first steps for various bioinformatics and molecular evolutionary analyses. The general belief is that reconstructing reliable phylogenetic relationships, for example, depends on the quality of MSAs [40]. However, generating reliable MSAs becomes extremely difficult when one has to deal with distantly related sequences displaying low similarity. Over long evolutionary times, sequences often undergo dynamic events such as duplication, recombination, insertion, and deletion. Such dynamic changes are important when we consider the evolution of protein families and their functions. Rigorous evaluation of MSA and phylogenetic methods, especially for highly diverged sequences with large numbers of sequence insertions or deletions, collectively called as indels, is possible if we have a tool that can simulate realistic sequence evolution incorporating indel events based on a valid biological model derived from empirical data.

In this Chapter, we describe most often used MSA methods, benchmark alignment databases currently available for testing MSA methods, and of sequence simulation methods. We discuss our contributions towards improving the realism of protein

sequence simulation by improved primary and secondary structure representation. We present two example protein superfamilies, which display the characteristics described above for primary and secondary structures. They are used as examples in Chapters 2 and 3 in order to test the weaknesses in the current sequence simulation methodologies. Finally, we discuss MSA reconstruction scoring methods and their limitations.

In Chapters 2 and 3, we introduce indel-Seq-Gen (versions 1.0 and 2.0), the first sequence simulation method for simulating realistic sequence evolution incorporating insertions and deletions. indel-Seq-Gen version 1.0 (iSGv1.0) begins this process by incorporating the ability to simulate heterogeneous evolution of multidomain protein families and length-dependent sequence conservation. indel-Seq-Gen version 2.0 (iSGv2.0) further improves realism by incorporating DNA simulation, lineage-specific evolution, motif conservation (using PROSITE-like regular expressions), and a richer representation of subsequence length constraints that allows for sequences to change length, but constricts the sequence to a minimum and maximum length. We also formalize and implement sequence constraints imposed by insertions, deletions, and substitutions. We fix a flaw in the indel models of most sequence simulators that biases simulation results for hypotheses involving indels.

In Chapter 4, we introduce a new method to score MSA reconstruction accuracies informed only by the location of gaps in each sequence, the gap profile score. We compare the gap profile score against curently used scoring metrics based on MSAs from four popular MSA reconstruction methods. We create a benchmark MSA dataset using iSGv2.0, in which all insertion and deletion events are correctly placed, and which covers a large range of parameterizations including: (1) tree shape, (2) number of taxa, (3) tree root-to-tip length, (4) $P_{ins}$:$P_{del}$ ratio, and (5) indel length.

In Chapter 5, we summarize our work for evaluating indels as characters of bio-

logical informativeness and present future directions for our research.

## 1.1 Background

### 1.1.1 Multiple Sequence Alignment

We first describe four commonly used MSA methods: ClustalW2.0 [94, 50], MAFFT-L-INS-i [47], Muscle (version 3.7) [24], and ProbCons [20]. All of these alignment methods are based on progressive alignment methods. These methods are used in the comparative analysis in the following chapters.

#### 1.1.1.1 Progressive Alignment:

Computing the optimal MSA for a set of sequences using a common heuristic, seeking to maximize the summed alignment score for each pair of characters (the sum of pairs score, further discussed in Section 1.1.2.1), is an NP-complete problem [100]. As such, it is only practical for very small numbers of sequences. In order to make MSA reconstruction practical for much larger sequence sets, MSA method designers have adopted a method that progressively adds sequences to a MSA, called progressive alignment [28].

In progressive alignment, the relationships between sequences are represented in a tree, called the guide tree. Sequence alignment progresses along this tree in prefix order (from the leaves of the tree upward toward the root). Alignment begins by aligning the closest pair of sequences (e.g., the alignment of `T1` and `T2` in Figure 1.1) by either using full dynamic programming [58, 81] or using faster $k$-tuple methods (similar to the method in FASTA [52]). The resulting alignment, called a *profile*, is represented as a matix of real values that represent the probability of

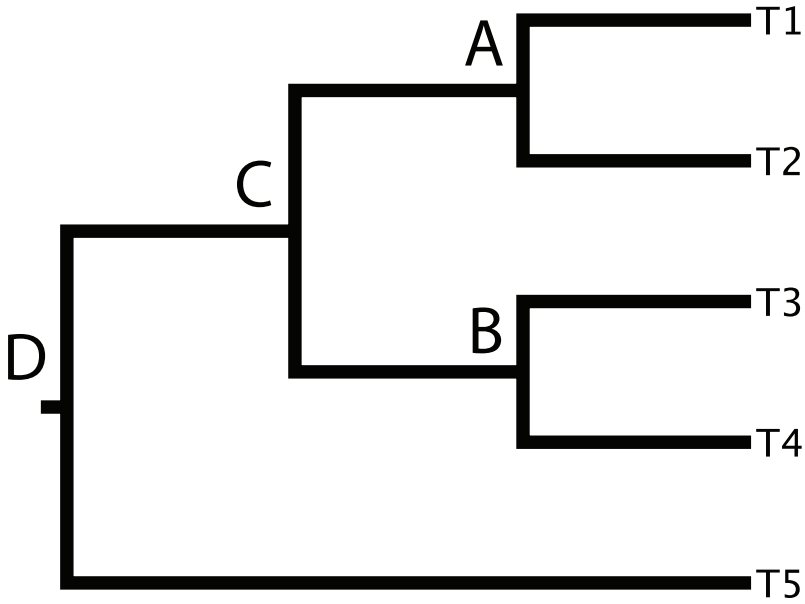



Figure 1.1: An example guide tree for the progressive alignment of a set of sequences T1–T5. (A, B) Nodes containing profiles built from pairwise sequence alignments. (C) Node containing a profile built from a profile–profile alignment. (D) Node containing a profile built from a sequence–profile alignment. Progressive alignment begins by aligning A (T1/T2) and B (T2/T3), followed by C (A/B, or T1/T2/T3/T4), and D (root, C/T5).

characters at each position from the alignment. Alignments at the internal nodes of the tree (e.g., nodes marked 'A', 'B', 'C', and 'D' in Figure 1.1) are derived from the descendant sequences of the inner branch. Each step of the alignment from leaves to root consists of aligning two sequences (Figure 1.1A, B), aligning a sequence to a profile (Figure 1.1D), or aligning two profiles (Figure 1.1C). Sequence–profile and profile–profile alignment is analogous to pairwise alignment, except positions in the alignment reflect the frequency distribution of characters at the sites (profiles) [25]. After the last alignment is completed at the root, the reconstructed MSA is returned. The progressive alignment strategy is considered to be justified since alignments of closer sequences will have less ambiguity.

Progressive alignment, however, suffers from two major problems stemming from

the greedy algorithm described above: (1) Corrections cannot be made for mistakes made early in the alignment process, and (2) it is difficult to choose correct substitution weight matrices and gap penalty values. Substitution weight matrix series (such as the Point Accepted Mutation [19] series or BLOSUM [38] series matrices) affect the alignment by encouraging the alignment of amino acids with similar properties (assigning positive values) and discouraging the alignment of amino acids that have differing properties (assigning negative values). The specific substitution matrix used when aligning two sequences should reflect the estimated evolutionary distance between the sequences, since larger changes occur between more distant sequences. Alignment methods often employ affine gap penalties. This means that there are two penalty values: the *gap-open* penalty $o$ and the *gap-extension* penalty $e$. The affine gap penalty for a gap of length $l$ is then $o + (l - 1)e$. This model is chosen to favor fewer gaps with longer lengths. Small changes in the values of $o$ and $e$ can have a high impact on the success of alignment methods. Additionally, the affine gap penalty model does not account for the localization of gap placement, i.e., gaps should be much more heavily penalized for occurring in important structural elements than in regions of little structural constraint. Each alignment step in the progressive alignment depends on the substitution and gap penalty parameters chosen to avoid committing mistakes that will propagate up the guide tree.

All of the chosen methods solve problem (1) above by using the strategy of *iteration*. In these methods, the first progressive alignment is created, and the MSA is returned. Errors made in the earlier progressive alignment procedure are fixed by breaking the returned MSA into smaller subalignments based on a given criterion. Each subalignment is re-aligned, and the full MSA is reconstructed by aligning the subalignments. The number of times this iteration occurs is often left to the user as a parameter. We present each alignment method, along with their progressive

alignment and iteration strategies below.

**ClustalW2.0:** ClustalW2.0 [50] is among the original progressive alignment methods. ClustalW2.0 attempts to reduce the number of mistakes made early in the alignment by dynamically changing the alignment parameters in a position- and region-specific manner. For example, gap-opening penalties are reduced early in the alignment within short stretches of hydrophilic residues, which are hypothesized to be unstructured regions. Previous research observed that gaps tend not to be closer than eight residues of existing gaps [65]. ClustalW2.0 utilized this observation by increasing the gap-open penalties within eight residues of existing gaps. Gap-open and -extension penalties are also changed based on a range of other considerations, such as the similarity of sequences (upweighting gaps in very similar sequences, downweighting for highly diverged sequences) and downweighting gap penalties in positions with existing gaps. Dynamic changes are also employed in choosing substitution matrices. The guide tree used with progressive alignment is a Neighbor-Joining (NJ) tree [74] calculated based on the pairwise-distance matrix. ClustalW2.0 chooses substitution matrices that most closely match the estimated divergence between sequences and profiles for each step in the alignment guide tree. ClustalW2.0 also uses sequence-weighting based on the distances in the guide tree, where the largest weight has a value of 1.0. Groups of closely related sequences are given small weights since much of the information contained in the sequences are duplicated, while highly divergent sequences with few relatives are given larger weights. Progressive alignment progresses as above, but the score for each position in the alignment is multiplied by the weights. Iteration is an option in version 2.0 and later, but is not done in default settings. Iteration is performed by removing each sequence from the MSA and realigning it to the MSA. This step is performed until either the alignment score does not improve

or the maximum number of iterations has been reached.

**MAFFT-L-INS-i:** MAFFT-L-INS-i [47] calculates an initial pairwise distance using the Smith-Waterman pairwise alignment scores [81], and creates an unweighted pair group method with arithmetic mean (UPGMA) guide tree. It then collects all gap-free portions of each pairwise alignment, and assigns an importance value to each segment as a function of the length of the sequence, the frequency with which the aligned position participates in pairwise alignments, and the total alignment score of the pairwise alignment. Internal nodes are represented as *groups* of sequences, whose scores are calculated by scoring the pairwise positions of all sequences from one group against all sequences from the other group, and tempering each contribution by the weighted sum of pairs (calculated as done in ClustalW2.0) assigned to each pair of sequences. The iterative refinement phase recalculates the importance values, and optimizes the alignment with the weighted sum of pairs score for group-to-group alignment as proposed by [34] (a method similar to ClustalW2.0, which weights sequence pairs based on their phylogenetic distances to compensate for biased contributions of highly similar sequence pairs) and the recalculated importance values as the objective function.

**Muscle3.7:** Muscle3.7 [24] builds alignments in multiple steps. In the first step, an UPGMA tree is reconstructed from the distance matrix created by calculating the fractional number of shared contiguous subsequences of length $k$ ($k$mers) between, based on a reduced alphabet, i.e., a smaller alphabet created by grouping amino acids with similar properties. Progressive alignment of sequences is performed using this tree. Using the returned MSA, a second UPGMA tree is reconstructed based on the distances between the aligned pairs of sequences, using Kimura's distance correction

for multiple substitutions at a single site distance measure [49]. A second progressive alignment is done using this tree. Iterative refinement is done by partitioning the guide tree into two subtrees by removing an edge from the guide tree, and reconstructing profile alignments for each subtree. These two profiles are realigned, at which time the sum of pairs score (SPS, described in Section 1.1.2.1) is calculated. If the SPS is improved over the current alignment, the new alignment is saved, otherwise it is deleted. Further iterations are performed by partitioning the guide tree by deleting a different edge from the second UPGMA guide tree, and repeating the process.

**ProbCons:** ProbCons [20] is a consistency-based alignment program that consists of five alignment steps. (1) It computes the match quality scores in a matrix of posterior probabilities that particular positions in each pair of sequences will align in the unknown "true" biological alignment. (2) Expected accuracies are computed by performing a variant of the Needleman-Wunsch [58] pairwise alignment where gap penalties are zero and match/mismatch scores are given by the posterior probabilities calculated in step 1. (3) The match quality scores are re-estimated by incorporating the similarities of a pair of sequences to a third sequence in the sequence set (*Probabilistic Consistency*). (4) The guide tree is reconstructed using UPGMA where the distance matrix is defined by the expected accuracies of alignments between sequences (as calculated in step 2), rather than the evolutionary distance between sequences. This distinction places importance on generating reliably aligned sequences early, rather than aligning evolutionary closer sequences, in the multiple alignment phase. (5) Progressive alignment is done based on the guide tree. The focus of the alignment step is maximizing the alignment score, where the score assigned to each match is from the transformed match quality scores calculated in step 3. As such, gap penalties are again set to zero. The iterative refinement procedure incorporated in ProbCons

is to randomly partition the sequences in the existing multiple alignment into two groups, and performing the progressive alignment procedure (step 5) to realign the two projected alignments.

MAFFT-L-INS-i, ProbCons, and Muscle3.7, are state-of-the-art MSA methods, and despite their differences, perform competitively in studies comparing their accuracy. ClustalW2.0, however, is often outperformed by these methods, yet it remains the most popular MSA method by far in phylogenetic studies [57]. Figure 1.2 is an example of how the MSA reconstruction methods affect the resulting MSA. The alignments, which reflect the true insertion and deletion histories, are displayed on top. Below this alignment are the MSAs reconstructed by each of the four MSA methods discussed above. Note that each of the MSA methods overfit the alignments by shrinking multiple gaps into long, non-gapped segments.

### 1.1.2 Benchmarking Multiple Sequence Alignments

Evaluation of MSA methods requires two components: a metric that measures how well a reconstructed MSA matches a given benchmark, and a benchmark dataset that is the "gold standard," with known correct alignments.

#### 1.1.2.1 Scoring Multiple Sequence Alignment Accuracy

Accuracies of MSAs are scored based on the ability to correctly identify the positional accuracy of (1) pairs, (2) columns, or (3) structurally-related characters between sequences as represented in the benchmark datasets.

The pair-based methods of MSA accuracy assessment come in three flavors: the sum of pairs score (SPS), the shift score [18], and the Modelers score ($f_M$) [75]. The

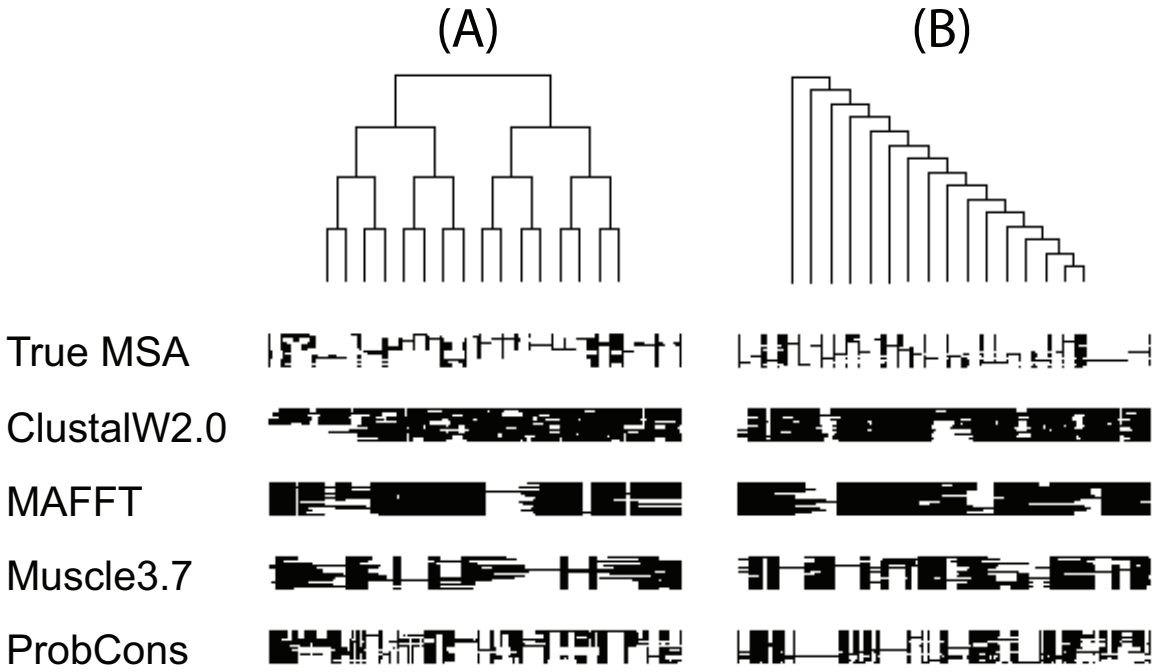



Figure 1.2: Examples of the true MSA compared to the MSA reconstructed by ClustalW2.0, MAFFT, Muscle3.7, and ProbCons. Each black pixel represents a character in the sequences and each white pixel is a gap in the alignment (the result of an insertion or deletion). The true MSA was generated using indel-Seq-Gen version 2.0 (iSGv2.0) [86]. The parameters used to simulate sequences are as follows: 16 taxa, root-to-tip tree length of 0.75 substitutions per site, $P_{ins} = P_{del} = 0.1$, indel length of 5 characters. The guide trees used for the simulation are shown at the top (A: balanced, B: pectinate). iSGv2.0 is described in Chapter 3.

SPS is defined as the percentage of correctly aligned character pairs. The shift score is similar to the SPS, except it assigns positive scores to close misses, e.g., one-off errors, and assigns a negative score to far misses, e.g., pairs that misaligned (shifted) by a certain number of characters. The $f_M$ score is defined as the percentage of correctly aligned pairs divided by the total number of aligned pairs in the reconstructed MSA. There is only one column-based method, the *total column* score (TC), defined as the percentage of correctly aligned columns of the MSA.

Structural-base scoring methods, such as the Structural Alignment of Multiple

Proteins (STAMP) [73], iRMSD [3], and APDB [62] scores, require the structural data for the aligned sequences. Such information is available only for well-characterized sequences, and as such, is a very fine-grained measure of alignment accuracy. As our research is predominantly associated with very distantly related sequences, we are unable to use such metrics.

The above MSA scoring methods also tend to ignore a potentially valuable source of information for very distantly related sequence sets, the insertions and deletions (indels) in the MSA.

#### 1.1.2.2 Benchmark Datasets

In order to introduce a new MSA method, a comparative analysis of the new method must be performed to show its superior performance versus existing methods. Prior to 1999, a common practice was to choose a small number of datasets that exhibited the features that showcased a method's strengths versus competing methods, while concealing the method's weaknesses [10]. These subjective test cases allowed developers to carefully choose the dataset and parameter combinations that would give their method the superior performance.

Since then, multiple objective benchmarks have been introduced: BAliBASE [93, 92], OXBench [68], PREFAB [24], IRMbase [88], and SABmark [98]. Although some structural databases have been used as MSA benchmarks, such as the Structural Classification of Proteins (SCOP) [41], the Homologous Structure Alignment Database (HOMSTRAD) [55], the Simple Modular Architecture Research Tool (SMART) [76], they are not organized as a benchmark suite, and therefore cannot be considered objective, as MSA method developers can choose the set of structures that best fit their methodologies.

**BAliBASE:** Benchmark Alignment dataBASE (BAliBASE) version 3.0 [92] is the most popular MSA benchmarking suite. It provides datasets that are based on both full-length sequences and on "core blocks." Core blocks are discovered by extracting secondary structures, such as $\alpha$-helices and $\beta$-strands, from the structural superpositions of sequences in the Protein Data Bank (PDB) [9]. Full-length sequences (sequences including unalignable regions) are provided to allow for more difficult alignment problems. BAliBASE contains 218 benchmark datasets subdivided into five reference sets, as follows: Reference 1 contains equi-distant sequences, Reference 2 are family alignments with one highly divergent "orphan" sequence added, Reference 3 are alignments of diverged subfamilies, where between-subfamily residue identity is $< 25\%$, Reference 4 are sequences with long N- and C-terminal extensions, and Reference 5 are sequences with long internal insertions. BAliBASE version 2.0 and later added three more reference alignments, References 6–8. Reference 6 are alignments of 12 families with repeated regions, Reference 7 are alignments of nine transmembrane protein families, where transmembrane regions are defined as the core blocks, and Reference 8 are alignments of five families for which the domain order has been permutated. Accuracy assessment is done using SPS and TC scores. Assessment of alignment accuracy on full-length sequences reports only the accuracy of the aligned core blocks.

**PREFAB:** The Protein REFerence Alignment Benchmark (PREFAB) [24] benchmarks reduces the time and work necessary to create benchmark alignments by using a pair of sequences as the benchmark. The pair of sequences are aligned using the structural alignment methods CE [77] and SOFI [11], annotating only those sequence pairs upon which they agreed. In order to make the multiple alignment, each sequence was used as a query in PSI-BLAST [2], from which up to 25 of the returned

sequences were extracted and added to the pool of sequences to be used for the benchmark dataset. Finally, assessment of MSA reconstruction accuracy is performed by comparing pairwise accuracies of alignment, using the $Q$ score, against the annotated regions of the original two sequences. The final PREFAB version 4.0 benchmark dataset contains 1681 alignments.

**SABmark:** The Sequence Alignment Benchmark (SABmark) [98] version 1.65 is separated into two categories based on sequence identity of the benchmark sequences: TWILIGHT, which contains 209 sequence groups displaying very low (0–25%) between-sequence identities, and SUPERFAMILY, which contains 425 sequence groups with low to intermediate (0–50%) between-sequence identities. SABmark also covers the known fold-space, i.e., the range of possible three-dimensional structure folds for proteins, by choosing high quality structures from the Structural Classification of Proteins (SCOP) [41] database. As in PREFAB, a pair of sequences is used for assessment, based on their structural annotations from CE and SOFI. SABmark selects up to 25 additional sequences for each pair from the SCOP database to make the final benchmark dataset. Assessment of MSA accuracy is performed using the $Q$ and $f_M$ scores. Unlike PREFAB, SABmark does not discard regions of the structural alignments that do not agree between CE and SOFI. As a result, MSA methods cannot obtain a perfect score for many of the benchmark datasets.

**OXBench:** OXBench [68] draws reference sequences from the 3Dee database of structural domains [78], creating reference alignments using the multiple structure alignment program STAMP [73]. STAMP returns a measure of reliability for each structurally aligned position. Similar structural domains are heirarchically organized based on the inferred global structural similarity between pairs or groups of proteins

($S_c$ score) returned by STAMP. OXBench consists of a total of 218 sequence families, organized into three datasets: the Master dataset, Extended dataset, and Full-length dataset.

**The Master dataset:** The Master dataset breaks down the 218 sequence families into subfamilies by clustering each pair of sequences based on their pairwise-identity between structural domains, using pairwise identity cutoffs of 60, 40, 30, 20, 10, and 5%, where the pairwise-identity cutoffs are defined upon the most dissimilar sequences in the cluster. This created an additional 391 sequence subfamilies, whose structural alignments were again optimized by STAMP. Finally, additional sequence subfamilies were created using a similar technique, where the clustering was performed using six cutoffs of the $S_c$ score. Removing identical subfamilies left a total of 672 benchmark datasets, containing anywhere from two to 122 sequences in each dataset. The Master dataset is further broken into sub-datasets: (1) set of pairwise families, containing families with only two members, (2) set of multiple families, containing more than two members, (3) set of small families, which consists of families containing eight or fewer structural domains, and (4) test and training sets, a two-fold separation of the Master dataset.

**The Extended dataset:** The Master dataset contains only sequences with known structure. The Extended dataset adds more sequences to each of the 672 alignment problems by extracting sequences with unknown structure that are clearly similar to each family from the SWALL [39] sequence database.

**The Full-length dataset:** In contrast to the Master and Extended datasets, which contain only the structural domains, the Full-length dataset contains the full-length proteins sequences obtained from SWISS-PROT database [6]. Only 605 se-

quence families are in the Full-length dataset, as it was not possible to identify full-length sequences for the domain sequences of 67 families of the Master dataset.

Assessment of MSA method performance is done in three ways: comparing alignment reconstruction to the reference alignment for (1) the entire length of the alignment and (2) regions with high $S_c$ values, called Structurally Conserved Regions (SCRs, no relation to the SCRs in the following Section), and (3) calculating the root mean square deviation and $S_c$ scores of the structural superposition implied by the alignment.

**IRMBASE:** The Implanted Rose Motifs Base (IRMBASE) [88] benchmark is the only commonly used benchmark based on simulated sequences. IRMBASE uses the Random Model of Sequence Evolution (ROSE) [85] to simulate substitution, insertion, and deletion processes acting on a single ancestral sequence to produce related descendant sequences, called *motifs*, along with the "true" MSA of the motifs. These sequences are then embedded into randomly generated, unrelated sequences, to create sets of artificial proteins. IRMBASE contains three reference datasets named *ref1, ref2,* and *ref3*. Reference sets contain 60 test sets. Test sets differ in the number of motifs and length of the random sequence added, i.e., embedding one, two, and three motifs into random sequences of length 400, 500, and 500 for ref1, ref2, and ref3, respectively. The structure of the 60 reference test sets remains the same, however: 30 test sets contain ROSE motifs of length 30, and 30 contain ROSE motifs of length 60, and 20 contain MSAs with four sequences, 20 contain MSAs with eight sequences, and 20 contain MSAs with 16 sequences). IRMBASE uses the SPS and TC scores in order to test MSA accuracy. Unlike the other benchmark datasets, which test the accuracy of aligning sequences globally, IRMBASE is designed to test alignment ac-

curacies locally, i.e., aligning only the embedded ROSE motifs rather than the entire sequence.

#### 1.1.2.3 Benchmark Dataset Issues

Multiple issues in benchmarking remain unaddressed. Benchmark datasets that are based on sequence simulations are biologically unrealistic [51, 88], since they do not represent the complex protein sequence heterogeneity and interrelationships between different sequence regions and the structure or function of such regions. Hand-curated and structure-based benchmarks based on real sequences [68, 24, 92, 98] have multiple limitations, such as small dataset size, ambiguous positional homology, unknown indel histories, and redundant test cases, among others [60]. The most commonly used benchmark database, BAliBASE v.3.0 [92], is additionally hampered by the non-independence between datasets [22], i.e., the same sequence is used in multiple datasets. Finally, the benchmark alignments cannot be used for testing phylogenetic methods, since the evolutionary history among the sequences is unknown.

## 1.2 Protein Family

### 1.2.1 Protein Structures

Most biological processes necessary for life are carried out by proteins, through catalyzing reactions, molecular transport, cellular signalling, and so on. Protein structure often relates to the protein function. The primary structure of a protein is the sequence of amino acids. Sites that confer function to the protein can often be pinpointed by comparing the primary structures since they are well-conserved between

members of the protein family. These regions of related proteins are tightly constrained because the function of the protein depends on its specific positions having amino acids with specific properties. The subsequences that confer the function are also often length-dependent, and have low tolerance of subsequence length changes, e.g., the CXXC motif in Thoiredoxin-fold proteins [16]. Because of their conservation, many of the functional subsequences have been characterized in the form of a regular expression, as in the Prosite database [79] (an example regular expressions can be seen in Figure 3.6, SCR1 and SCR2). Another important category of structures are secondary structures, which are the folding units of subsequences such as $\alpha$-helices and $\beta$-strands, whose folding structures distinguish them from flexible regions, commonly called random coils. Secondary structures are important in shaping the larger three-dimensional structure of the protein, and evolve under different evolutionary pressures in order to preserve their structures: On conserved regions, fewer and shorter insertions and deletions are found and amino acid substitution rates are lower. Additionally, secondary structures have different amino acid frequencies [17], which are additionally affected by where the sequence will reside in the cell, for example, transmembrane regions contain a higher proportion of hydrophobic amino acids [59].

### 1.2.2 Protein Families Used in this Study

A protein family is a set of proteins derived from a common ancestral sequence, that share similar structures and functions, and often share high sequence similarity. For developing the simulation methods for Chapters 2 and 3, we will use two example protein families, the G protein-coupled receptor (GPCR) class A family and the lipocalin superfamily, which is a subordinate of the calycin superfamily.

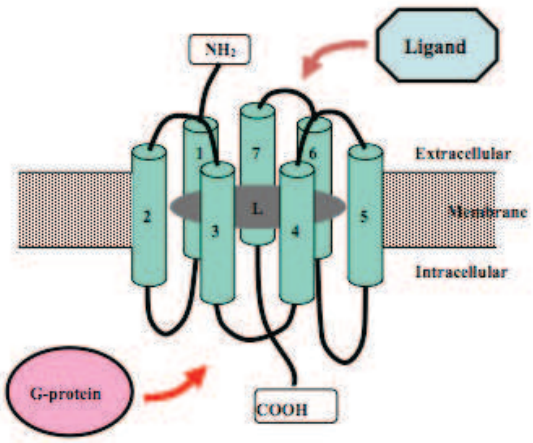



Figure 1.3: The canonical structure of a GPCR sequence. Image from [56].

#### 1.2.2.1 Class A G Protein-Coupled Receptors

The GPCR superfamily includes many proteins that are medically important. As many as 50% of pharmaceuticals act on GPCRs. The GPCR superfamily is characterized by seven transmembrane $\alpha$-helical regions (green cylinders in Figure 1.3), four each of cytosolic and extracellular regions, with the N-terminal starting in the extracellular region. A canonical GPCR is illustrated in Figure 1.3. Each region (extracellular, transmembrane, and cytosolic) is constrained by different evolutionary pressures, such as amino acid frequencies and ability to accept indels. These constraints make the GPCR family an ideal candidate for simulating heterogeneous sequence evolution. Class A, or rhodopsin-like, are the most diverged class of GPCR, divided into 19 subfamilies [45].

#### 1.2.2.2 Calycin/Lipocalin Superfamily

In the Structural Classification of Proteins (SCOP) database [13], calycins belong to the "all-beta proteins" class. The calycin superfamily consists of a number of $\beta$-barrel protein families, such as the lipocalins, avidins, and metalloproteinase in-

hibitors (MPIs), as shown in Figure 1.4. As illustrated in Figure 1.5, the lipocalin family proteins have three structurally conserved regions (SCRs 1, 2 and 3) [29]. Lipocalins are divided into two groups: kernel lipocalins, which contain all three SCRs, and outlier lipocalins, which contain at least one SCR. The avidin family contains a motif different from the lipocalins, while the MPIs have no motif recorded. The lineage-specific motifs make the calycin superfamily an ideal candidate to model motif conservation and lineage-specific simulation, while the eight $\beta$-barrels of the lipocalins are a good candidate to simulate heterogeneous sequence evolution.

## 1.3 Sequence Simulation

The goal of simulating sequence evolution is to realistically portray the evolutionary tension between (1) processes that change a biological sequence through point mutations, insertion/deletion (indel), as well as more dynamic chromosomal rearrangement, and (2) functional constraints that restrict such changes. The tension between these processes form the patterns that we see in extant homologous biological sequences.

### 1.3.1 Sequence Simulation Input

Biological sequences, as mentioned above, are not static entities. As time passes, these sequences evolve through changes in nucleotide bases or insertions and deletions of sequences, which can be inferred from extant sequences. Such changes can be mapped onto a bifurcating tree, a phylogenetic tree, that represents the evolutionary relationships of the extant sequences. From these evolutionary patterns, hypotheses of historical events can be tested, e.g., the timing of events that cause genetic isolation such as geographical separation [42]. Each bifurcation of the phylogenetic

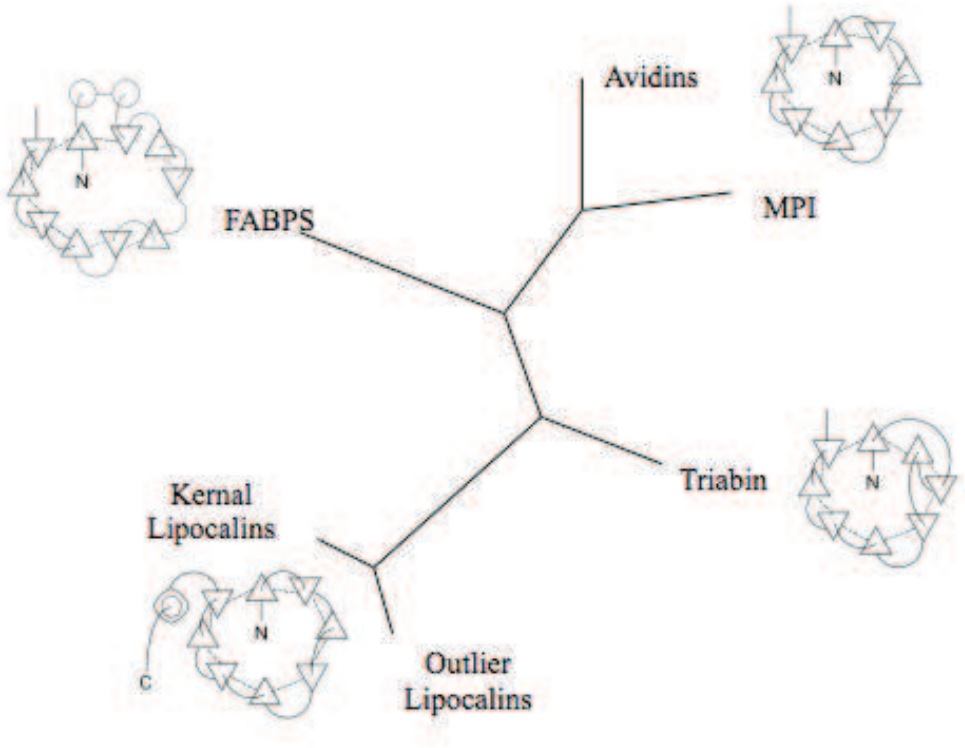



Figure 1.4: The calycin superfamily tree. Calycins are a superfamily of soluble, antiparallel $\beta$-barrel protein subfamilies. The $\beta$-barrel conformation is such that each $\beta$-strand interacts with its immediate neighboring strands with respect to primary sequences, with the exception of the interacting first and last $\beta$-strands. The members of the Calycin superfamily are the fatty-acid binding proteins (FABPs), avidins, metalloproteinase inhibitors (MPIs), triabins, and the lipocalins. The lipocalins are also a superfamily of proteins that consist of two subfamilies, the kernel lipocalins and the outlier lipocalins. The structure of each family is represented by a cartoon, where the triangles represent $\beta$-strands (triangle direction represents the direction of the sequence, from N- to C-terminus), circles representing $\alpha$-helices, and lines representing random coil regions. Image from [29].

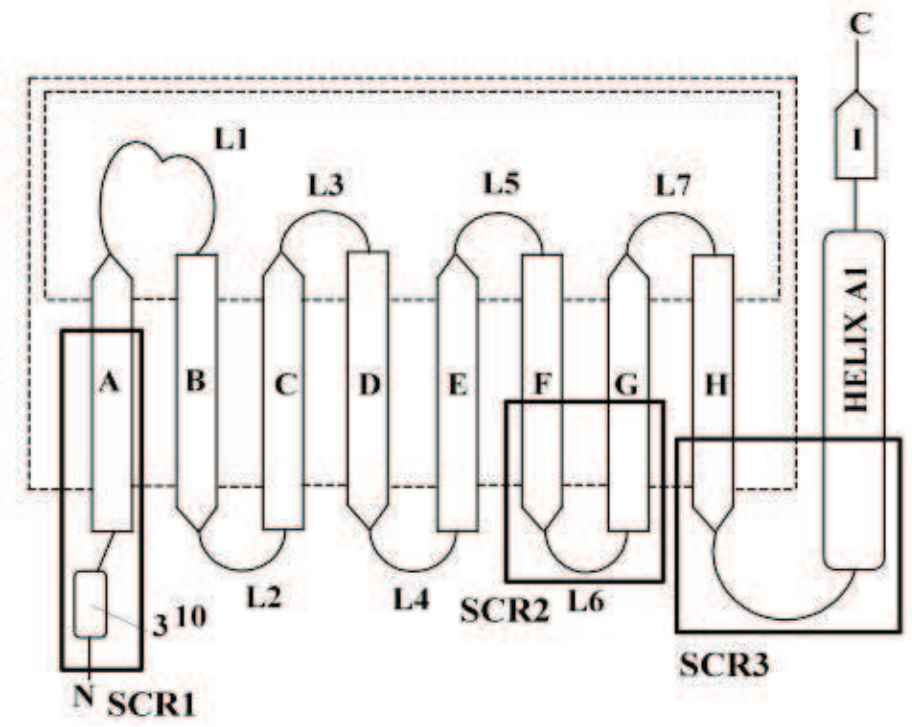



Figure 1.5: The canonical structure of a lipocalin protein. Eight $\beta$-strands (A–H) make up the barrel region of the lipocalin, with three structurally conserved regions (SCR1–SCR3) imparting specific functions to the lipocalin members. Kernel lipocalins contain all three SCRs, while outlier lipocalins contain either 1 or 2 of the SCRs. Image from [29].

tree is the inference of such genetic isolation between sibling populations. In order to mimic such evolutionary events, sequence simulation methods require: (1) the phylogenetic tree, and (2) the models of sequence evolution, including substitution and insertion/deletion (indel) parameters.

Phylogenetic trees are made up of two components: (1) the topology of the tree, i.e., the branching pattern, and (2) branch lengths, with the length corresponding to some evolutionary metric, in the case of simulation methodology, the number of substitutions per site of the sequence that is evolving along the branch. Two extreme examples of topologies are balanced trees and pectinate trees shown in Figure 1.6. A tree is balanced if the descendant subtrees are equivalent for any branching point,

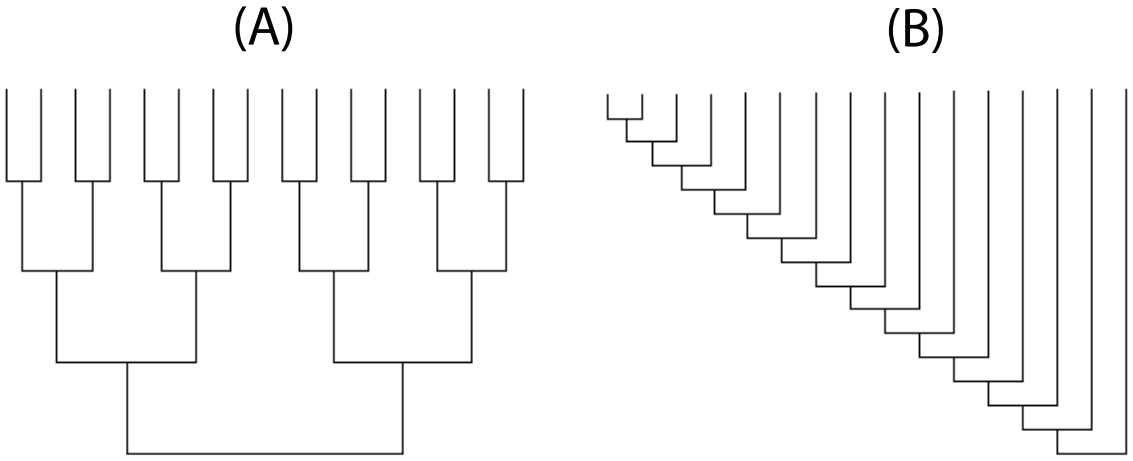



Figure 1.6: Two examples of bifurcating trees: (A) the balanced tree, and (B) the pectinate tree.

whereas a tree is pectinate if one descendant subtree consists of the a single descendant sequence for any branching point. The branching patterns of any random phylogeny will fall in between these two extremes. Phylogenetic trees are necessary to simulate sequence evolution. In the simulation framework, such trees are called *simulation guide trees*.

The simulation of evolutionary processes are additionally guided by components that describe how sequences evolve. One component is the model of substitution processes, shown as $\Theta_T$ in Figure 1.7. Substitutions are based on models of continuous substitution evolution processes: *e.g.*, JTT [44], PAM [19], or BLOSUM [38] for proteins, and HKY [37] or GTR [104] for nucleotides, in addition to the equilibrium character frequencies. Modeling site-specific substitution parameters (site rates, e.g., rates distributed among sites according to the Gamma distribution [103], and percent invariable sites) can also be specified, although such specification is optional. The second component to sequence evolution specification is indel specifications, shown as $\Lambda_T$ in Figure 1.7. The probability of creating an insertion or deletion is usually specified as a percentage of the number of substitutions that are expected to occur,

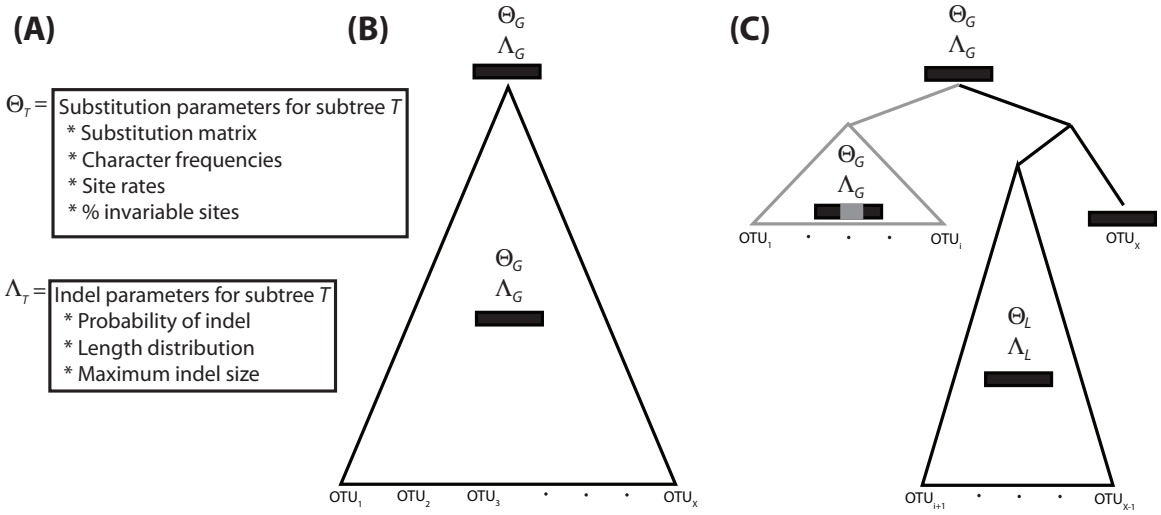



Figure 1.7: A comparison of the basic simulation paradigm and more realistic biological sequence evolution. (A) Substitution ($\Theta$) and indel ($\Lambda$) parameters used for simulation methods. (B) The basic simulation paradigm. Given a root sequence and a global set of substitution and indel parameters ($\Theta_G$ and $\Lambda_G$, respectively), simulation proceeds by applying changes in a Monte Carlo manner over all sequence positions, following the guide tree and ending with a set of sequences, which are called as operational taxonomic units (OTUs). (C) More realistic biological sequence evolution. Starting with a root sequence and an initial level of substitution and indel parameters ($\Theta_G$ and $\Lambda_G$, respectively), evolution at sites may become constrained by gaining a functional motif (the gray box shown in the gray lineage), and the substitution and indel parameters may be changed ($\Theta_L$ and $\Lambda_L$), or the initial parameters are maintained without gaining any functional motif as shown with $\text{OTU}_X$.

e.g., an insertion probability of 0.02 specifies that one insertion is expected to occur for every 50 substitutions. Besides the probability of an insertion or deletion occuring, simulation methods require the input of the maximum allowable size of an indel, and the expected distribution of indel lengths.

### 1.3.2 Simulation Method Background

The first generation of simulation methods, for example, `evolver` [105] and the simulator in Molphy [1], has mostly focused on reconstructing substitution events (for both nucleotides and amino acids) over entire sequences. These methods cannot ac-

commodate the numerous evolutionary forces that act upon, e.g., functional domains found in families of proteins. These domains frequently, but not always, coincide with structural domains and are connected together by sequence regions that have fewer constraints. Conservation of these domain regions often imposes different evolutionary constraints, including different evolutionary parameters (e.g., amino acid frequencies and substitution rates) and indel parameterizations (e.g., maximum indel size, length distribution, and probability of opening an indel). Examples of these regions are the GPCR transmembrane region, in which hydrophobic amino acids are overrepresented, along with lengths constrained to be between 17–24 amino acids. Smaller functional units, or motifs, may require other specific models, such as invariability of amino acid positions and prevention of indel events within the motif, *e.g.*, the structurally conserved regions of the lipocalin family. Seq-Gen [69] introduced substructures that are allowed to have different lengths and evolve along different evolutionary trees. However, indels are still unaccounted for. One cannot change evolutionary pressures acting upon the substructures, such as differing substitution models and motif conservation, either.

Indels are a necessary component when portraying sequence evolution. However, their addition in sequence simulation poses a number of problems. One set of problems relates to the indel creation itself, that is, where to and where not to place indels, the distribution of indel lengths, and how to generate inserted sequences. The first program incorporating indel events simulation is the Random Model of Sequence Evolution (ROSE) [85]. ROSE implements indel events using a simple, strict model; indels occur linearly as a function of the substitution rate and with a user-defined length distribution. The sequence inserted is a random sequence based on the given amino acid frequencies. Empirical indel models whose indel probabilities are nonlinear with respect to evolutionary distance, such as those described by Benner et al. [8] and

Chang and Benner [15], cannot be utilized in ROSE. User input parameters are also global across the entire sequence, not allowing for domain-specific parameterizations. ROSE also extended the Gamma distribution, a distribution that is used to explain the rates in which sites accept substitutions [106], to encompass indel constraints as well. If the Gamma rate is below a certain threshold, it forbids an indel to occur in the region. At the end of a simulation run, ROSE returns the "true" MSA that reflects the true evolutionary path of the sequences. The utility of true MSAs in indel-producing methods is that it can be used to test the accuracy of multiple sequence alignment methods and various hypotheses [84, 51, 88]. Despite these advances, ROSE does not simulate known evolutionary pressures that act upon biological sequences, such as domain- and motif-specific constraints.

To improve the realism of simulated sequences over ROSE, two areas must be addressed: lineage- and site-specific sequence conservation and indel processes. The DNA With Gaps (DAWG) [14] program, which simulates non-coding DNA evolution, introduced indels based on an exponential time distribution that determines the waiting time until the next indel event. The waiting time is calculated based on the indel probability as a function of sequence size. DAWG adjusts the sequence length after each indel event. DAWG also incorporated an indel length distribution that follows the empirically derived power law distribution of Chang and Benner [15]. The MySSP program simulates non-coding DNA, and chooses indel lengths in a normally distributed fashion centered around a user-input mean length [71]. A parameterized model was introduced in the program SIMPROT, which employs another empirically determined indel length model, the Qian-Goldstein distribution [67], and simulates a continuous indel model by correcting for multiple indels in the same position based on the starting branch length [64]. The EvolveAGene3 program simulates coding sequences with indel frequencies as empirically observed in *Escherichia coli*

evolution [36]. The recently introduced GSIMULATOR package directly estimates parameters by training the simulator on a set of pairwise alignments, and using the estimated parameters to perform the simulations [99, 12]. Consequently, users cannot directly set indel parameters in GSIMULATOR.

Lineage- and site-specific conservation as well as heterogeneous evolution are needed to improve the realism of sequence evolution simulators. Homogeneous sequence evolution, as illustrated in Figure 1.7B, is prevalent in many simulation methods, including EvolveAGene3 [36] and DAWG [14]. Richer representations allow heterogeneous evolution among sequence partitions, where each partition can be defined by a different set of substitution and indel parameters (e.g., Figure 1.7C, gray lineage). SIMPROT [64] includes such partition-wise simulation. For site-specific conservation, the current state of the art is found in ROSE, implemented by disallowing sites and subsequences from accepting indels. Site-specific substitution processes, however, are constrained to be either completely invariable or mutable to any other character. Thus, functional constraints on substitution patterns within the conserved region cannot be simulated, although functional regions often depend on the properties of the residues within the regions to maintain their functions. This inability to conserve residue sets in the sequences affects the ability to simulate highly diverged superfamily-level evolution. Lineage-specific evolution is represented only by MySSP [71], which allows users to set substitution and indel parameters on each branch of the input guide tree. The functionalities of these simulation methods are summarized in Table 1.1.

Table 1.1: A comparison of sequence simulation methods.

| | ROSE | DAWG | MySSP | SIMPROT v1.03 | iSGv1.0 | EvolveAGene3 | iSGv2.0 |
|---|---|---|---|---|---|---|---|
| Data simulated: | | | | | | | |
| Non-coding DNA | Yes | Yes | Yes | No | No | Yes | Yes |
| Coding DNA | No | No | No | No | No | Yes [a] | Yes |
| Protein | Yes | No | No | Yes | Yes | No | Yes |
| Indel treatment: | | | | | | | |
| Continuous | Yes | Yes | Yes | Yes | Yes | Yes | No |
| Dynamic length adjustment | No | Yes | No | No | No | No | Yes |
| Event tracking | No | No | No | No | No | No | Yes |
| $P_{ins}, P_{del}$ independent | Yes | Yes | Yes | No | Yes | Yes | Yes |
| Insertion/deletion placement treated differently | No | No | No | No | No | No | Yes |
| Empirical length distribution[b] | No | CB04 | No | CB04,QG01 | CB04 | *E. coli* | CB04 |
| Overlapping indels | Yes | Yes | Yes | Yes | Yes | No | Yes |
| Heterogeneous evolution (partitions) | | | | | | | |
| Gamma distribution | No | Yes | No | No | No | No | Yes |
| Invariable site proportion | No | No | No | Yes | Yes | No | Yes |
| Indel probabilities | No | No | No | Yes | Yes | No | Yes |
| Lineage treatment & Functional constraints | | | | | | | |
| *Lineage options:* | | | | | | | |
| Parameter changing | No | No | Yes | No | No | No | Yes |
| Pseudogene simulation | No | No | No | No | No | No | Yes |
| *Indel modeling:* | | | | | | | |
| Indel depends on $N_{ins}, N_{del}$ | No | No | No | No | No | No | Yes |
| *Motif conservation:* | | | | | | | |
| Lineage-specific | No | No | No | No | No | No | Yes |
| Length-specific | Yes | No | No | No | Yes | No | Yes |
| Site-specific | No | No | No | No | No | No | Yes |

[a] EvolveAGene3 can output amino acid sequences. However, simulation is done only at the DNA level. The amino acid sequence outputs are translations of the resulting DNA sequences.

[b] CB04: Zipfian distribution [15], QG01: Qian-Goldstein distribution [67].

Limitations of biological realism with simulated sequences is the primary reason MSA methods continue to be evaluated on the hand-curated and structural benchmarks, despite the issues involved with them. Therein lies the primary reason we develop sequence simulation methods. Minor improvements to the limitations of benchmark datasets can be made with more information, such as the addition of more datasets and larger sequence alignments, whereas major improvements can be made to simulation methods by incorporating more realistic models. In curated benchmarks, the true evolutionary history can never be known. In simulated datasets the evolutionary history is always known. The disadvantage of simulated datasets, the lack of realism *can* be improved. In Chapter 2, we introduce indel-Seq-Gen version 1.0 (iSGv1.0), which introduces heterogeneous protein sequence evolution, insertion and deletion events, and subsequence length conservation. In Chapter 3, we further improve sequence simulation by correcting a flaw present in the indel models of many simulation methods. We improve sequence simulation realism further by incorporating lineage- and site-specific conservation models, improving motif conservation through the introduction of regular expression conservation, and incorporating nucleotide simulation.

## 1.4 Performance Analysis Concepts Used

Before introducing our method for incorporating indels into MSA informativeness measures, we first introduce some fundamental concepts of performance analysis, often used in machine learning, whose metrics we modify to incorporate into our method.

### 1.4.1 Sensitivity and Specificity

Sensitivity and specificity are statistical variables that measure the performance of binary classifiers, i.e., classifiers that make predictions for only two classes. Sensitivity ($Sn$) measures the proportion of predicted positives in a classification example that are truly positive, i.e.,

$$Sn = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

where TP is the number of true positives and FN is the number of false negatives. Using sensitivity alone, a classification method will be perfectly sensitive simply by predicting all examples as true. Specificity ($Sp$) is used to rule out such predictors, as it measures the number of positive predictions that are falsely predicted as positive:

$$Sp = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}},$$

where TN is the number of true negatives, and FP is the number of false positives.

### 1.4.2 Receiver Operating Characteristic Curve

The Receiver Operating Characteristic (ROC) curve is a graphical plot of the false positive rate (or 1 − specificity) and true positive rate (or sensitivity) for a binary classifier [82]. Discrete classifiers, such as decision trees, produce only a point in the ROC space (Figure 1.8, classifier D). Other classifiers, such as neural networks and Support Vector Machines (SVMs), assign a numeric value associated with the confidence that an instance belongs to the predicted class. Such classifiers can be converted into a discrete binary classifier when a threshold is applied. In such cases, any instance in which the confidence is above the threshold will be assiged to one class (positives), any instance below the threshold will be assigned to the other class

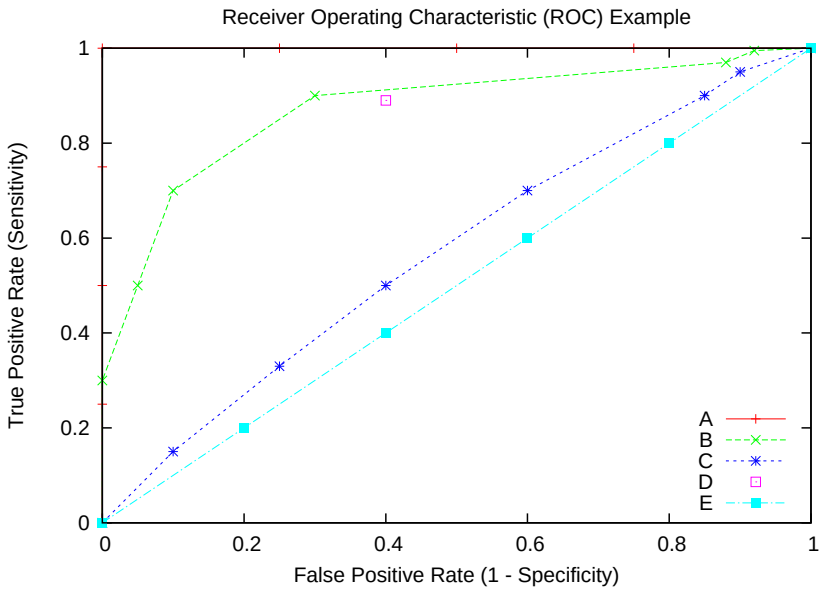



Figure 1.8: An example of the Receiver Operating Characteristic (ROC) curve for five binary classifiers. Classifier D is a discrete classifier, producing only a single point in the ROC space. Conversely, classifiers A, B, C, and E rank test examples by assigning a probability, a numeric value that represents the confidence that a particular instance belongs to the predicted class.

(negatives). Since each threshold will produce a different value in ROC space, varying the threshold from the highest value to the lowest value will induce a curve in ROC space (Figure 1.8, classifiers A, B, C, and E). In order to judge the goodness of the classifier, the area under the ROC curve (AUC) is calculated. The worst possible performance of a classifer is achieved by an AUC value of 0.5 (Figure 1.8, classifier E). Such a case is the equivalent of random guessing, i.e., the expected outcome of a classifier with no information about the data being classified. The best case for a classifier is an AUC value of 1 (Figure 1.8, classifier A), i.e., correctly predicting all true positive values, while also correctly identifying the true negatives.

## 1.5 Incorporating Indel Information in MSA

Building a MSA is done by aligning the nucleotides or amino acids, hereafter referred to as characters, in columns based on their similarity. Gaps, which represent the insertion or deletion (indels) of one or more characters of the sequence, are inserted to align the characters in columns. The assumption of many MSA methods is that substitutions provide adequate information for inferring the evolutionary relationships (*homology*) in any or all biological sequence sets. As such, MSA objective functions tend to minimize the number of indel events in an alignment under the assumption that the most parsimonious explanation of indel histories is achieved by minimizing the number of unique indel events. For highly diverged datasets that contain a large number of indels, however, this strategy combines small, spread out gaps into fewer but larger gaps during MSA inference, as shown in a study by Golubchik *et al.* [32]. This process of minimizing the number of gaps in the alignment creates "gap magnets" [54], i.e., large gaps in the alignment created by the magnet-like attraction of small gaps towards gap-rich regions. Since secondary analyses, such as phylogenetic reconstruction, often ignore gaps in their inference, one may be inclined to favor MSA methods that create gap magnets – they are visually appealing and look right. Gap magnets, however, overrepresent character similarity by overfitting the sequence data. Such overfitting introduces noise into later analyses (e.g., phylogenetic reconstruction). The effects of removing noisy data has lead to studies such as Talavera and Castresana [90], which remove ambiguously aligned blocks (blocks with many gaps; noisy regions) to improve the inference of phylogenies from MSAs. Such techniques are not yet well-understood. For example, Liu *et al.* [53] performed a phylogenetic study using non-coding DNA for large-scale phylogenetic reconstruction. In their study, removing ambiguous regions not only did not improve the reconstructions, but

often degraded the phylogenetic inferences.

Ideally, homologies of both characters and indels should be used in secondary analyses. Particularly, the rarity of indels can provide unique information for large-scale evolutionary relationships for highly divergent taxa, where character substitutions are dense, resulting in many homoplasies (convergent evolution, i.e., acquiring the same character state through differing evolutionary paths). Using indel information is a very difficult issue, however, as it requires MSAs to provide accurate positional inferrence of indels. The ability of MSA methods to accurately place indels has not been studied. To do so, we require a benchmark dataset to test MSA method performance that is not limited by unknown character homologies or unknown phylogenetic relationships. In practice, a computer-generated alignment is rarely used as it is, but rather it is adjusted by shifting characters to get them aligned to represent their assumed homologies. This manual adjustment process causes gaps to be moved from one site to another. It is not applicable to the regions where homologous relationships among characters are not clear, and often such adjustments are done arbitrarily. As Morrison [57] pointed out, judging the efficacy of MSA methods has not been a strong motivation to switch MSA methods when phylogenetic analysis is done. Morrison comprehensively surveyed phylogenetics literature from 2007–2009 to understand how the MSAs were generated. Over 50% of the initial alignments were generated with ClustalW2.0 [50], an alignment method that has been shown to compare unfavorably to more recent alignment methods [24, 88, 20, 61]. The Sum of Pairs and Total Column character scoring methods were used in these comparative studies of MSAs. In Chapter 4, we compare the gap profile scoring method against commonly used scoring schemes based on character homologies: the sum of pairs, total column, shift, and $f_M$ [75] scores. We compare four MSA methods: ClustalW2.0 [94], MAFFT [48], Muscle3.7 [24], and ProbCons [20]. Finally, we assess how well "good" alignments

(with respect to MSA quality scores) affect secondary analyses by comparing the accuracy of the phylogenetic trees inferred from the alignments.

# Chapter 2

# indel-Seq-Gen version 1.0: Protein Family Simulator Incorporating Domains, Motifs, and Indels

indel-Seq-Gen version 1.0 (iSGv1.0) [87] is our new method of generating realistic protein families, accomplished through the introduction of multiple models of indel evolution and the ability to parameterize and simulate heterogeneous domains. It provides a simple and biologically realistic method of modeling a protein family. It allows the use of multiple related root sequences, instead of a single static root sequence or multiple random sequences, to generate a realistic large sequence space. This is useful for evaluating, for example, protein classification methods. We show that when compared with other methods, our method could generate divergent protein sequences without destroying important functional properties, whereas providing a clean and concise method of model creation. iSGv1.0 also addresses functional subsequence conservation for indels using a novel quaternary invariable array.

## 2.1 Method

### 2.1.1 Protein Sequence Evolution

As the name implies, we use Seq-Gen (Version 1.3.2) from Rambaut and Grassly [69] as the substitution evolution engine. The current Seq-Gen incorporates both nucleotide and amino acid sequence simulation and succeeded the original protein sequence generator PSeq-Gen [35]. The reason for using Seq-Gen as the core engine is threefold: 1) We found no reason to reinvent a method that evolves sequences with substitutions as many methods already exist [1, 69, 105, 85, 64], 2) Seq-Gen has been widely used in simulation studies, and 3) the setup used in Seq-Gen is a good fit to iSGv1.0s family-modeling system. Our approach adds new capabilities to Seq-Gen as described below.

### 2.1.2 Protein Family Creation

Protein families are often characterized by their structural and functional components, domains, and motifs. The heterogeneity in evolutionary rates caused by domains and motifs is often modeled with continuous or discrete gamma rate distributions (e.g., [103, 105, 1]). However, if the functional regions are well known and the user wishes to simulate a sequence family that reflects such characteristics, it is better to provide region- or site-specific constraints and evolutionary models, rather than to allow the simulator to randomly select conserved sites. For example, a transmembrane region should not be guided by the same substitution model as a hydrophilic coil region. Neither should a sequence motif occurring in a loop region be changed at the same rate as the surrounding region. Highly diverged families such as the G protein-coupled receptors (GPCRs) also accumulate indel events throughout their evolution without

destroying their functional units. Our strategy for simulating protein family evolution is through 1) the introduction of domain units and 2) the introduction of invariable sites, motifs.

#### 2.1.2.1 Motif Conservation

The conservation of essential motifs in sequence generation can be accomplished by disallowing substitution events within the motifs. Invariable sites are introduced for this purpose. However, when indels are incorporated, the conservation of length-sensitive motifs becomes a problem. For example, ROSE allows for the preservation of motifs by joining the gamma-distribution rates and invariable sites into a mutation probability vector, which we call the I + $\gamma$ array. Each site in the I + $\gamma$ array specifies the substitution rate for the corresponding amino acid site. If the rate is 0.0, then the site is invariable. If the rate is less than 1.0, the site is not allowed to have any indel. Tying indels to the relative substitution frequencies is a drawback in this representation because it cannot represent low frequency indels in regions that have high substitution rates, such as transmembrane regions. Neither can it represent highly variable sites in length-based motifs that depend on the distance between particular amino acids, such as the $CXXC$ motif. When shorter or longer, this motif loses its propensity for creating the disulfide bridges that are essential for thioredoxin-fold proteins [16].

To rectify these issues, we created a new representation of invariable arrays with 4 classes of invariable sites: 0, no constraint; 1, invariable, 2, no indel; and 3, invariable and no indel. “Invariable” sites can have no substitution, but insertions are allowed between consecutive invariable sites. “No-indel” sites refer to the position in which indels are not allowed but substitutions are. “Invariable and no-indel” sites allow neither substitution nor indel to occur. Neither an insertion nor a deletion can exist

between 2 consecutive "no-indel" positions (positions represented as "22," "23," "32," and "33" in the array). For example, the thioredoxin-fold $CXXC$ motif (where $X$ is any amino acid) can be represented as "3223" in the invariable array. This will hold the two $C$s invariable, allow the two $X$s to be substituted independently, and disallow indels from occurring within the motif, preserving the length dependence.

#### 2.1.2.2 Heterogeneous Evolution among Domains

The idea of partitions was introduced in Seq-Gen [69] to allow variations in evolutionary rates and patterns among domains. Each partition represents a subsequence of a protein, and each partition evolves independently as specified by the evolutionary tree. However, more variations in evolutionary patterns need to be incorporated to represent, for example, different indel rates in secondary-structure regions compared with nonstructured coil regions [95]. In iSGv1.0, the percentage of invariable sites, branch lengths (representing substitution rates), amino acid frequencies, substitution models, and indel rates can all be varied between partitions. Such flexible options allow us to generate realistically complex protein families.

### 2.1.3 Indel Event Handling

Indel events are governed by four parameters: the probability of an indel occurring, the placement of indels, the length distribution, and the maximum indel length. For insertion events, one more parameter is required for generating the amino acids for an inserted sequence.

#### 2.1.3.1 Probability of Indels

In iSGv1.0, the following two indel models are used:

- Linear model, which assumes that the number of indel events is linearly related to the substitution rate: $P(\text{indel}) = kd$, where $k$ is a user-defined constant and $d$ is the substitution rate specified by the branch length between the ancestral and descendent sequences.

- The Chang and Benner [15] model, which specifies the probability of an indel based on the following exponential equation:

$$P(\text{indel}) = 0.224k - 0.0219e^{-0.01168d} \tag{2.1}$$

#### 2.1.3.2 Placement of Indels

The placement of indels is chosen randomly. For insertions, invariable and no-indel sites are excluded from consideration. Deletion can happen only at "no constraint" positions ("0" in the invariable array). For a deletion of size $n$, any position that is within $n-1$ positions from a non-zero position in the invariable array is excluded.

#### 2.1.3.3 Length Distribution of Indels

The distribution of gap lengths in protein sequences has been studied through the use of pairwise alignments [15] as well as by examining structural databases [67, 33]. By default, iSGv1.0 will create the normalized Zipf distribution for indel lengths [15]:

$$N = 2628(X^{-1.821}), \tag{2.2}$$

where $N$ is the number of indels of length $X$. Other models can also be input by the users as precalculated indel distributions (e.g., following a structural indel distribution developed by Qian and Goldstein [67] or Goonesekere and Lee [33]).

#### 2.1.3.4 Insertion Sequence Options

iSGv1.0 incorporates 2 methods of amino acid sequence generation. The random model randomly chooses amino acids based on the given frequencies. The second method is unique because it chooses amino acids in the insertion sequence by incorporating the "neighbor preference" [102] along with the amino acid frequencies. Neighbor preference was derived by studying neighboring amino acid positions in functional proteins in order to empirically determine the effect of the side chains in the acceptance of functionally beneficial amino acid changes.

We use their $20 \times 20$ matrix to generate insertion sequences in a Bayesian fashion:

$$P(i|j) = \frac{P(j|i)P(j)}{P(i)}, \tag{2.3}$$

where $P(j|i)$ is the probability that amino acid $j$ follows amino acid $i$ and $P(j)$ and $P(i)$ are the frequencies of amino acids $j$ and $i$ in the sequence, respectively. Given an amino acid, we choose the next amino acid based on the probability $P(j|i)$. When the first amino acid is inserted, we use the amino acid preceding the insertion point as amino acid $i$. If the insertions are longer than 1 amino acid, we use the newly generated amino acid as the predecessor to find the next amino acid.

Originally, neighbor preferences were built on the protein sequences derived from the *Escherichia coli* K-12 genome. We also provide neighbor preferences calculated over all of the protein sequences contained in the Swiss-Prot database [5]. The neighbor preferences are global over the entire simulation run of a family of sequences.

### 2.1.4 Root Sequence Options

iSGv1.0 provides two options to incorporate a root sequence provided by the user: a single root sequence or a set of sequences. For the single root sequence option, a

sequence and its invariable array are read in and used verbatim for all simulation runs. However, for protein families that may contain highly variable regions, varying the root sequence in order to explore a larger sequence space may be advantageous. To facilitate this, iSGv1.0 incorporates the option of inputting a set $S$ of sequences in the form of a multiple sequence alignment (MSA). For each simulation run, a single root sequence, which may vary between simulation runs, is constructed from $S$ in the following manner. For each position (column) in the MSA, where half or more of the sequences are non-gapped, we choose a representative amino acid using either of the following 2 methods: 1) the consensus method, which chooses the representative using majority rule or 2) the random method, which probabilistically chooses the representative based on the amino acids that exist in the column of the MSA. If any alignment position has a gap in more than half of the sequences, the position is considered to be an insertion position and is ignored in the construction of the root sequence. The following 3 parameters can be set for building the root sequence from a set of sequences:

- The range of columns in the MSAs to use.
- The number of sequences used. All (the default) or a given number of sequences can be selected from the MSA using the bootstrap-sampling method (selecting randomly with replacement). The bootstrap sampling option is useful, for example, when using highly divergent sequences with a large number of gap sites. The simple consensus from such divergent sequences may include mainly gaps and very few amino acids. On the other hand, when there are many equally probable root sequences in $S$, the user may opt to randomly select 1 sequence as the root for each simulation run.
- The root sequence generation method. The default consensus method uses the

majority rule to choose the amino acid for the column, using a coin toss to break ties. The random method randomly chooses the amino acid representing the column based on the position-specific amino acid frequencies, with the exception of invariable columns. For invariable columns, the representative amino acid is chosen using the consensus method.

### 2.1.5 Implementation

iSGv1.0 is freely available at http://bioinfolab.unl.edu/∼cstrope/iSGv1.0/. iSGv1.0 has been tested on the RedHat Linux, SuSE Linux, IRIX, and Macintosh OS X operating systems with the PERL and gcc compilers.

### 2.1.6 Comparison of Simulation Methods

#### 2.1.6.1 Transmembrane Region Prediction

For the GPCR family simulation, we used HMMTOP 2.0 [96] to predict the number of transmembrane regions as well as the position of the N-terminal region (intra- or extracellular). Realistically simulated sequences should have seven transmembrane regions and extracellular N-terminal regions.

#### 2.1.6.2 Beta-Strand Prediction

For the lipocalin family simulation, we used PSIPRED [43] to predict the secondary structures. Realistically simulated sequences should have 8 beta-strands that correspond to the beta-barrel structure found in the lipocalin family proteins.

#### 2.1.6.3 Blast Similarity Search

In order to see how the simulated sequences preserved the similarities against the template proteins, we performed BlastP [2] protein similarity searches against the protein database UniProt [5], using the simulated sequences as the queries. We did not use the option to filter out the low-complexity sequences. The default $E$-value threshold (10) was used to cut off the search results.

#### 2.1.6.4 Pfam Search

Simulated sequences were used to search the profile hidden Markov model database, Pfam [7], using the program `hmmpfam` of the HMMER package [21] with the `Pfam_ls` database to find the glocal alignments. The scores against the models "7tm_1" (PF00001; for GPCRs) and "Lipocalin" (PF00061; for lipocalins) were recorded. The default $E$-value threshold (10) was used to cut off the search results.

#### 2.1.6.5 Parametric Bootstrap and Phylogenetic Analysis

Following the phylogenetic tree reconstructed from the template alignment (see Results and Discussion), 1,000 simulated data sets were generated, each including the same number of protein sequences as in the template set. Phylogenetic trees were reconstructed from these data sets. The maximum parsimony and Neighbor-Joining [74] with JTT distance [44] estimation methods, consensus trees, and bootstrap values were calculated using PHYLIP programs (version 3.65) [26].

## 2.2 Results and Discussion

### 2.2.1 Simulation Setup

#### 2.2.1.1 Template MSA

We chose to model protein sequences based on two protein family data sets: 1) the vertebrate olfactory receptor family and 2) the lipocalin protein family.

**G protein-coupled Receptors.** The vertebrate olfactory receptor family belongs to the GPCR superfamily, Class A (rhodopsin-like). GPCR proteins have seven transmembrane regions, and their sequences are known to be highly diverged including many indels. Among them, vertebrate olfactory receptors are relatively conserved and generating a template MSA is not too difficult. We chose two sequences from each of the 14 subfamilies, yielding 28 protein sequences. We also included one outgroup sequence (OPSD_BOVIN) from the GPCR rhodopsin class (Class A), as was done previously [31]. Thus, the total number of sequences in our template alignment is 29.

The process of building the template MSA is illustrated in Figure 2.1, steps 1–3. Based on the sequence characterization in UniProt, we first split each sequence into 15 parts: the seven transmembrane regions (TM1-TM7), four extracellular regions (EC1-EC4), and foru cytosolic regions (CY1-CY4). We manually adjusted region boundaries where the UniProt characterizations were conflicting.

MSAs of each region were done using T-Coffee [61] and adjusted manually. These regional alignments were concatenated together obtaining the template MSA for the entire protein sequences (see Figure 2.2a and the actual alignment is shown in Figure A.1). In Step 4, we reconstructed the maximum parsimony phylogeny using PAUP* version 4.0b10 [89]. Using the topology obtained from the template MSA, the branch lengths were calculated for each region. The average number of changes

per site for each segment was obtained by dividing each tree length by the number of sites in the segment. Subsequence parameters are set as shown in step 6 of Figure 2.1 (full specifications are found in Figure A.2a). They include amino acid frequencies, relative evolutionary rates, root sequence templates, indel parameters, and phylogenies. We obtained the amino acid frequencies specific to the 3 regions (EC, CY, and TM) using 1594 GPCR Class A sequences found in UniProt.

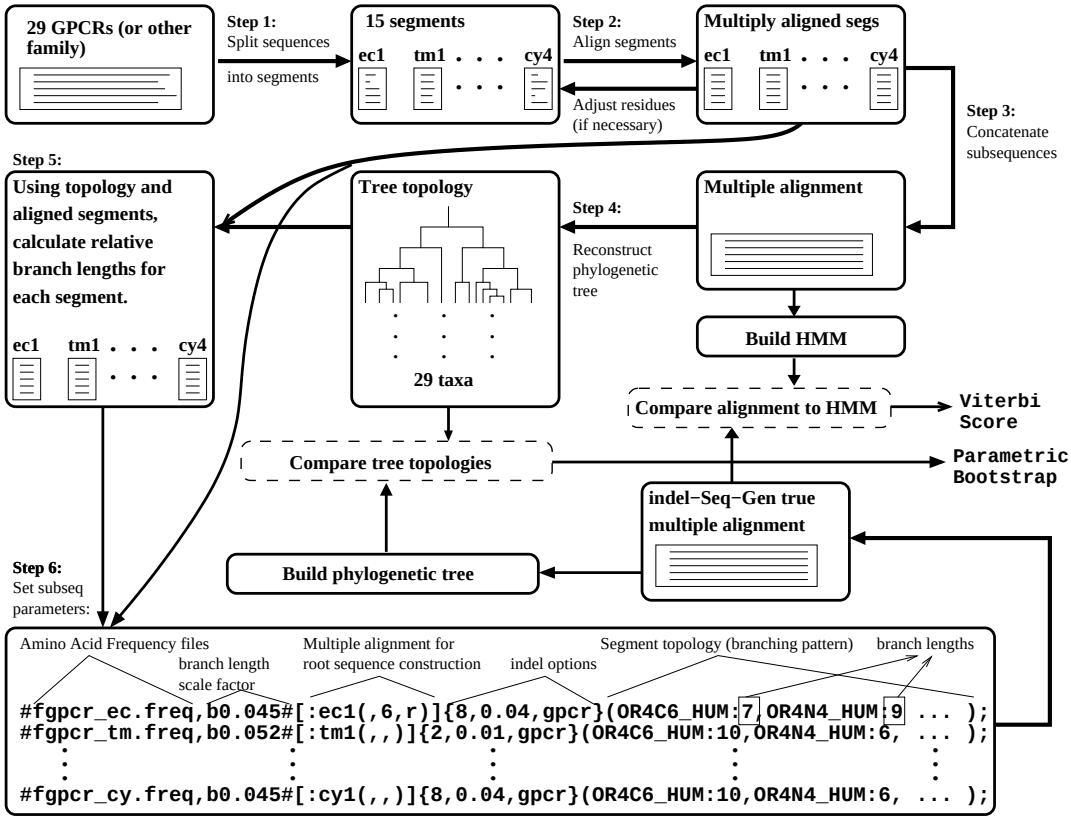



Figure 2.1: A flow chart showing the process of iSGv1.0 protein sequence simulation. In this example, parameterization of evolutionary information is done based on the vertebrate olfactory receptor family. Steps 1–5 illustrate the process of obtaining the template MSA, phylogeny, and various parameters. Step 6 shows the sample input for iSGv1.0 with each line showing parameters used for a different subsequence (domain). See Figure A.2 for the example input file. Two evaluation methods, phylogenetic analysis and alignment comparison using profile HMMs, are shown in dashed boxes.

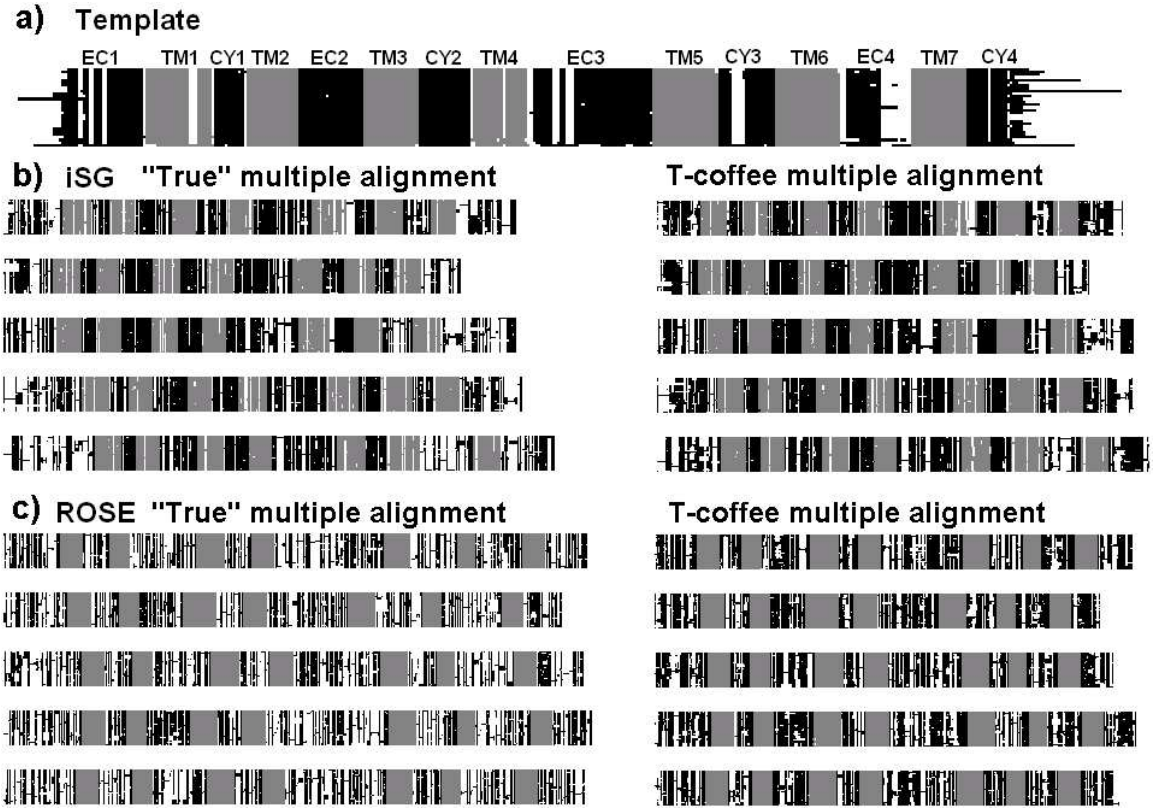



Figure 2.2: MSAs of the template and simulated GPCR sequences. Each pixel represents one position in the MSA. The color of the pixel represents: a gap (white), an amino acid from a transmembrane region (gray), and an amino acid not from a transmembrane region (black). The template alignment (a) includes 29 GPCR sequences, and the 15 subsequence regions are indicated above. For 5 simulated data sets produced by iSGv1.0 (b) and ROSE (c), the true MSAs and MSAs reconstructed by T-Coffee are shown.

**Lipocalins.** The template MSA and phylogeny for the lipocalin family including 23 sequences were obtained from an evolutionary study done by Sánchez et al. [80]. The sequences were split into 12 regions including 5 beta-strand regions as shown in Figure 2.3c. Note that the region including 4 short beta-strand regions ($B_{5678}$ in Figure 2.3c) is treated as a single evolution unit. Branch lengths were estimated from each segment as described before. Four coil regions (indicated by "C" in Figure 2.3c) were very short (each with 4–6 amino acids in length). Therefore, their branch lengths

were obtained based on their concatenated sequences. The amino acid frequencies specific to the beta-strand, coil, or alpha-helix regions were calculated using those obtained from the template MSA (75% weighting) combined with pseudocounts (25% weighting). The pseudocounts were obtained either from the conformational parameters by Chou and Fasman [17] for beta-strands and alpha-helices or from Jones et al. [44] for the N-terminus, C-terminus, and coil regions.

#### 2.2.1.2 Setting Parameters

The setting of the parameters was done so that the simulated sequences were close to the template alignment, but still general enough to allow for variations. For example, in the GPCR template alignment (Figure 2.2a and Figure A.1a:

- TM1, 2, 4, and 6 contain more gaps than TM3, 5, and 7 and
- The loop regions between TM4 and TM7 (EC4, CY3, and EC5) are more diverged than other regions.

Parameters such as indel rates for each region are set to closely follow these features. Figure A.1 shows the invariable array used with iSGv1.0 that holds the conserved motifs found in the vertebrate olfactory receptors by Fuchs et al. [30]. For the lipocalin template alignment, we allowed no indel in the beta-strand or alpha-helix regions. For the other lipocalin regions, we set the deletion rate to be 1 per 100 substitutions and the insertion rate to be twice higher than the deletion rate. Table 2.1 shows a list of the parameters used in iSGv1.0, ROSE, and SIMPROT. In this study, a single global substitution matrix (JTT for iSGv1.0, SIMPROT, and Seq-Gen and PAM for ROSE) was used. However, with iSGv1.0 we can specify different substitution matrices for different subsequences (*e.g.*, a TM-specific substitution matrix).

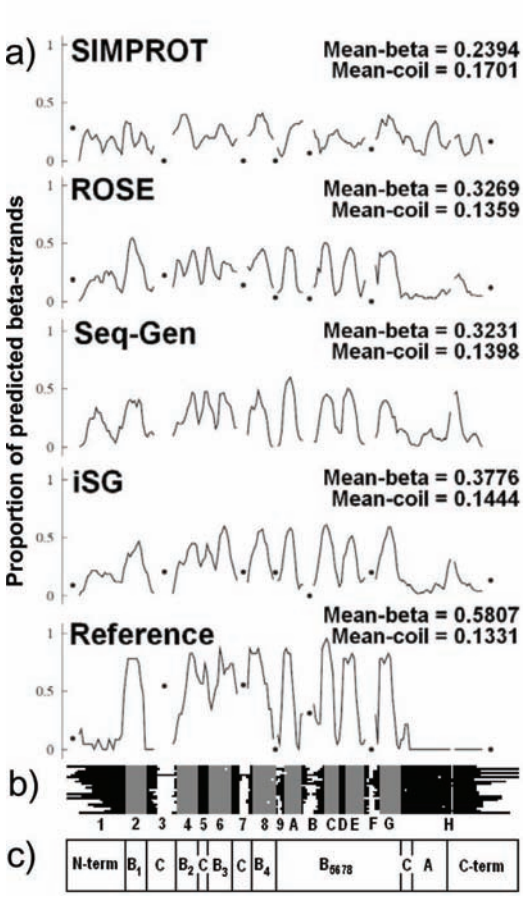



Figure 2.3: Distribution of predicted beta-strands along the lipocalin sequences. (a) Proportions of predicted beta-strands are plotted for each amino acid position of the alignments. For the reference alignment, the proportions are based on 23 lipocalin sequences. For simulated sequences, average proportions were plotted based on 5 simulated data sets using their true alignments obtained from each simulation method. Seven regions are mainly gaps with amino acids inserted into a few sequences (less than half the sequences have an amino acid). The proportions of predicted beta-strands in these 7 regions are represented by a single dot. These regions correspond to those in which indels are allowed (3, 7, 9, B and F as well as the start and end regions in the reference alignment, see the region labels in b). For each method, the average numbers of positions predicted as beta-strands are calculated from subsequence regions simulated as beta-strands (regions 2, 4, 6, 8, A, C, E, G in b) and coils (regions 1, 3, 5, 7, 9, B, D, F, H in b). These values are shown as mean-beta and mean-coil, respectively, in each plot. (b) The reference lipocalin alignment obtained from Sánchez et al. [80] is schematically shown with a single pixel representing 1 amino acid. The gray pixels represent beta-strands. The alignment is 213 positions long and consists of 23 lipocalins. Seventeen regions are labeled 1 through H. (c) Twelve subsequence regions used for simulation. The region $B_{5678}$ concatenated short consecutive regions from 9 to G.

Using these parameters a data set of 29 GPCR-like sequences or 23 lipocalin-like sequences were simulated following their template phylogenies.

Table 2.1: Parameter settings for indel-Seq-Gen, ROSE, and SIMPROT used in this study.

| Set | Parameter | indel-Seq-Gen | ROSE | SIMPROT |
|---|---|---|---|---|
| **GPCR** | Partitions[a] | $\texttt{EC}_1 \ldots \texttt{EC}_4$, four extracellular partitions<br>$\texttt{TM}_1 \ldots \texttt{TM}_7$, seven transmembrane partitions<br>$\texttt{CY}_1 \ldots \texttt{CY}_4$, four cytosolic partitions | Single Sequence, I+$\gamma = 1.0$,<br>*except for:* TM region I+$\gamma = 0.7$<br>Conserved region I+$\gamma = 0.0$ | 15 partitions:<br>corresponds with iSGv1.0 partitions |
| | Amino acid frequencies | `gpcr-cy, gpcr-tm,` and `gpcr-ec` | `gpcr-all` | JTT frequencies |
| | Substitution matrix | JTT | PAM | JTT |
| | Guide tree[b] | | | |
| | –*Topology* | Same as template, for all partitions | Same as template | Same as template, for all partitions |
| | –*Branch lengths* | Calculated for each partition | Same as template | Calculated for each partition |
| | Indels: | | | |
| | –*Maximum size* | $\texttt{EC}_1 : 8, \texttt{EC}_2 : 2, \texttt{EC}_3 : 6, \texttt{EC}_4 : 6$<br>$\texttt{TM}_1 \ldots \texttt{TM}_7 : 2$<br>$\texttt{CY}_1 : 2, \texttt{CY}_2 : 2, \texttt{CY}_3 : 6, \texttt{CY}_4 : 8$ | 4<br>corresponding partition | Same as iSGv1.0 for each |
| | –*Frequency (per substitution)* | $\texttt{TM}_1 \ldots \texttt{TM}_7 = 0.01$<br>$\texttt{CY}_4, \texttt{EC}_1 = 0.04$, $\texttt{EC}_4 = 0.03$<br>$\texttt{CY}_1 \ldots \texttt{CY}_3, \texttt{EC}_2, \texttt{EC}_3 = 0.02$ | 0.00065[c] | Same as iSGv1.0 for each<br>corresponding partition |
| | –*Length distribution* | Zipfian | Zipfian | Zipfian |
| **Lipocalin** | Partitions | $\texttt{B}_1 \ldots \texttt{B}_4$, four beta partitions<br>$\texttt{B}_{5678}$, a region of four betas<br>`N-term, C-term`, terminal regions<br>`C`, concatenated coil regions. | Single sequence, I+$\gamma = 1$,<br>*except for:* Beta-strands I+$\gamma = 0.9$<br>Terminal regions I+$\gamma = 1.2$ | Same as iSGv1.0 for each<br>corresponding partition |
| | Amino acid frequencies | `beta, alpha,` and JTT frequencies | `lipo-all` | JTT frequencies |
| | Substitution matrix | JTT | PAM | JTT |
| | Guide tree[b] | | | |
| | –*Topology* | Same as template, for all partitions | Same as template | Same as template, for all partitions |
| | –*Branch lengths* | Calculated for each partition, *except for:*<br>Regions $\texttt{B}_{5678}$ and `C` | Same as template | Calculated for each partition |
| | Indels: | | | |
| | –*Maximum size* | $\texttt{B}_1 \ldots \texttt{B}_4 : 0, \texttt{B}_{5678} : 1$<br>`C: 2, N-term, C-term: 4` | 4 | Same as iSGv1.0 for each<br>corresponding partition |
| | –*Frequency (per substitution, insertion/deletion)* | $\texttt{B}_1 \ldots \texttt{B}_7 = 0.0$, $\texttt{B}_{5678} = 0.01/0.0$<br>$\texttt{Nterm}, \texttt{Cterm} = 0.04$<br>$\texttt{C} = 0.02/0.01$ | 0.00090/0.00045 | $\texttt{B}_1 \ldots \texttt{B}_7 = 0.0$, $\texttt{B}_{5678} = 0.01/0.01$<br>$\texttt{Nterm}, \texttt{Cterm} = 0.04$<br>$\texttt{C} = 0.02$ |
| | –*Length distribution* | Zipfian | Zipfian | Zipfian |

[a] For ROSE, each partition was specified by assigning different values in the I+$\gamma$ array as shown. Information regarding the conserved regions were taken from Fuchs et. al. [30].

[b] The guide tree was obtained from the template alignment as shown in Figure A.1.

[c] The indel frequency was set so that the hidden Markov model generated by ROSE was reasonably diverged from the template sequence, i.e. the Viterbi scores shown in Figure 2.4 were similar to those generated by indel-Seq-Gen.

#### 2.2.1.3 Comparison of Simulated Sequences between iSGv1.0 and Other Methods

We compared iSGv1.0 with ROSE, Seq-Gen, and SIMPROT for their protein family simulation performance. As described before, ROSE uses the I + $\gamma$ array for simulating protein family sequences, and SIMPROT and Seq-Gen both allow subsequence generation. The major disadvantage of using ROSE was that we could not allow low-frequency indels although allowing amino acid substitutions as observed in TM regions in GPCRs. Because indel rates are linearly correlated with substitution rates in ROSE, in low-frequency indel regions, we were required to set low substitution rates. Because parameters (*e.g.*, indel rates and amino acid frequencies) could not be varied among subsequences in ROSE, we were forced to set parameters with the "average case" values across the entire sequence. Note also that setting these parameters in ROSE was very tedious and time consuming. For example, over 300 real-valued numbers had to be assigned to the I + $\gamma$ array in the case of GPCRs, and the array does not visually align with the sequence because the former is a real-valued array and the latter is a character array. iSGv1.0, on the other hand, uses integer values in the invariable array, which can be easily aligned with a character array (see Figure A.1a.

Setting the parameters for SIMPROT and Seq-Gen were more straightforward. The Seq-Gen root sequence was constructed using iSGv1.0s root sequence construction method, and all parameters available to Seq-Gen were set the same as iSGv1.0. For SIMPROT, we partitioned the sequence as we did in iSGv1.0, and many parameters could be set as done with iSGv1.0. The root sequence is randomly generated in SIMPROT. Additionally, SIMPROT uses an equal rate for insertions and deletions.

One thousand data sets were simulated using each method. The "true" MSAs showing the actual indel positions were collected. We compared the true alignments

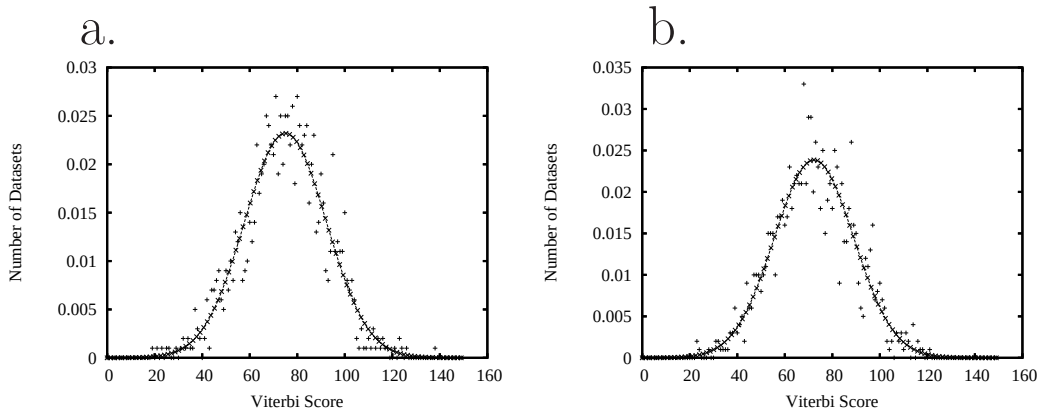



Figure 2.4: Comparisons of the template alignment and the true alignments generated by indel-Seq-Gen (a) and ROSE (b). Comparisons were done by using COACH [24]. The profile hidden Markov model was generated from the template alignment, and compared against each of 1000 simulated datasets. A Viterbi score was calculated from each comparison. The distributions are a): $\mathcal{N}(75.03, 17.20)$ and b): $\mathcal{N}(72.30, 16.72)$, respectively.

generated by iSGv1.0 and ROSE against the profile HMM built from the template GPCR alignment, shown in Figure 2.4 [24]. No significant differences were shown between the normal score distributions obtained from iSGv1.0 (mean = 75.03 and variance = 17.20) and ROSE (mean = 72.30 and variance = 16.72), indicating that the sequences generated by the 2 methods using the set of the parameters we chose were approximately equivalent with respect to the template MSA.

#### 2.2.1.4 Simulated Sequences

In general, functional domains are under strong selective constraints and very few indels are tolerated. TM regions are one such example. On the other hand, changes (both substitutions and indels) within loop regions that simply connect functional or structural domains often have much smaller negative consequences and thus are under very weak constraints. In Figure 2.2, TM regions are illustrated with a gray color and indels are shown as white gaps. As Figure 2.2a shows, vertebrate olfactory receptor

proteins are more conserved in the regions between TM2 and TM4 than other regions, as found in [30]. Note that some TM regions (TM1 and TM4) can have a few small indels. In the simulated sequences, we attempted to recreate these heterogeneous sequence features: fewer indels between TM2 and TM4 and, conversely, more indels in N- and C-terminal regions and those between TM4 and TM7.

As shown in Figure 2.2b, the simulated sequences by iSGv1.0 reproduced the intended sequence features well. Because the equivalent parameters could be set, SIMPROT produced the true alignments similar to those with iSGv1.0 (data not shown). ROSE, however, spread the indels throughout the entire length of the proteins (Figure 2.2c). When the simulated sequences were aligned using T-Coffee MSA method, reconstructed alignments based on iSGv1.0s data sets placed indels in a manner that is much closer to the template MSA than those based on ROSEs data sets.

iSGv1.0 allowed low-frequency indels within TM regions. From Figure 2.2, it appears as if iSGv1.0 allows far too many indels in the TM regions compared with the template alignment. However, in this experiment we intentionally chose parameters to introduce a large number of indels to our simulated data sets. In ROSE, on the other hand, it was not possible to allow indels within TM regions. In order to introduce any indels within TM regions, the mutation array values used in ROSE need to be set at 1.0 or higher. Because indel rates are linearly correlated to substitution rates, using such high mutation array values would perforate each region with so many indels that the resulting loss of protein function would be ineviTable. To avoid such unreasonable simulation conditions, we used conservative substitution rates with ROSE (0.7 in the $I + \gamma$ array). Consequently, in the true alignments produced by ROSE (Figure 2.2c) there are gray "islands" (TM regions) where no indel was found. The side effect of this was an easier MSA reconstruction for T-Coffee, which was nearly perfect in the

TM region reconstructions.

#### 2.2.1.5 Conservation of Transmembrane Regions

To check if the TM regions were retained in the reconstructed data sets, we predicted TM regions from simulated GPCR sequences using HMMTOP 2.0 [96]. The numbers of predicted TM regions were $7.03 \pm 0.30$ by iSGv1.0, $5.94 \pm 1.25$ by ROSE, $0.20 \pm 0.37$ by SIMPROT, and $6.84 \pm 0.91$ by Seq-Gen (histogram of results in Figure 2.5). Clearly, iSGv1.0 has the greatest accuracy in reproducing the seven TM regions. The N-terminus (EC1) was correctly predicted to be extracellular 94% of the time in simulated data sets of iSGv1.0. Conversely, with ROSE and Seq-Gen, only 44% and 55% of simulated sequences, respectively, were predicted to have the extracellular N-terminus correctly. These results clearly show that iSGv1.0s ability to allow different amino acid frequencies between domains is effective in preserving important features of transmembrane proteins more accurately than other methods. This is important because results from Panchenko and Madej [63] suggested patterns found in non-domain regions, such as loop or N-/C-terminal regions, were more important in recognizing superfamily members than considering only the conserved core regions.

#### 2.2.1.6 Beta-Region Prediction

To examine the conservation of beta-strand regions, secondary structures were predicted from simulated lipocalin sequences. As shown in Figure 2.3, comparing with the reference sequences, all simulated sequences have lower proportions of beta-strands indicated by lower "mean-beta" values and lower peaks of beta-strand plots corresponding to the eight beta-strand regions (gray regions in Figure 2.3c). However, iSGv1.0 performed better than the other methods with just over 5% better accuracies.

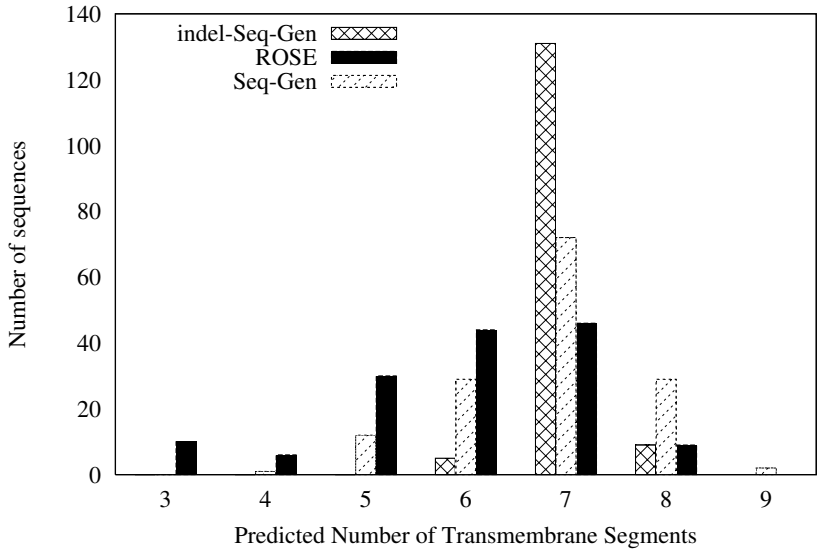



Figure 2.5: The distribution of the number of predicted transmembrane regions for the olfactory receptor datasets simulated by Seq-Gen, indel-Seq-Gen, and ROSE using HMMTOP 2.0. For each method, five datasets (including 29 sequences) are used in this analysis.

Among the simulation methods compared, SIMPROT performed most poorly. Eight beta-strand regions were indistinguishable in its simulated sequences, indicated also by the lowest mean-beta and highest "mean-coil" values. The poor performance by SIMPROT is explained by its inability to use a specific root sequence.

In this study, beta-strand regions were characterized only by the amino acid composition derived from small samples of lipocalin sequences. Therefore, the performance by iSGv1.0 to reconstruct beta-strand structures was surprisingly good considering the use of such simple parameters. Better performance can be expected by optimizing pseudocount methods and using secondary structure-specific amino acid composition and substitution matrices derived from larger samples.

#### 2.2.1.7 Blast and PFAM Search Results

In order to further examine how simulated sequences preserved functionally important sequence properties, we used the simulated sequences as queries for BlastP protein similarity search. Table 2.2 summarizes the search results. The top 250 hits obtained by all methods except SIMPROT were correctly from various members of vertebrate olfactory receptors. Seq-Gen performed the best of the three methods indicated by the highest average scores. The absence of indels in Seq-Gen generated sequences must have caused a low percentage of gaps in the alignments. iSGv1.0 performed nearly as well as Seq-Gen and outperformed ROSE with higher scores and lower percentages of gaps. The lower percentage of gaps found in Blast alignments with iSGv1.0 sequences than those with ROSE sequences can be explained by the use of amino acid frequencies specific to subsequences (e.g., TM, extracellular, and cytosolic regions) in iSGv1.0. It must have produced amino acids that are most likely to appear in each region of GPCRs. As a result, fewer gaps were required in the iSGv1.0 alignment making better hit scores than in ROSE. Contrary to iSGv1.0, ROSE, and Seq-Gen, simulated sequences by SIMPROT did not find any vertebrate olfactory receptor protein under the $E$-value threshold used with the BlastP search. Within the threshold, SIMPROT sequences averaged between three and five unrelated hits, with the average highest score = 30 (E value = 1.5). PFAM search results showed similar conclusions as BlastP. Except for SIMPROT sequences, the search against PFAM profile HMM database using simulated sequences returned 7tm_1, the profile-HMM entry for GPCR Class A, as the most common hit. Table 2.2 (in the bottom half) summarizes the scores against the 7tm_1 profile-HMM by the four simulation methods. iSGv1.0 clearly outperformed ROSE. The difference between iSGv1.0 and Seq-Gen was again very small, with slight advantage to iSGv1.0-generated sequences. No significant hit by

GPCR-derived profile-HMMs were found using SIMPROT-simulated sequences as queries.

Table 2.2: The BLAST and PFAM statistics of vertebrate olfactory receptors when using simulated sequences from iSGv1.0, Seq-Gen, SIMPROT and ROSE as the query sequence.

| Method | Number of top hits | avg[a] | min[a] | max[a] | %ID[b] | %SIM[b] | %GAP[b] |
|---|---|---|---|---|---|---|---|
| iSGv1.0 | 25 | 174.0 | 115.0 | 303.2 | 34.3 | 55.9 | 3.2 |
| | 100 | 164.7 | 106.0 | 303.2 | 33.1 | 54.6 | 3.3 |
| | 250 | 155.9 | 95.8 | 303.2 | 32.1 | 53.5 | 3.4 |
| ROSE | 25 | 141.1 | 93.6 | 258.8 | 35.3 | 53.1 | 7.5 |
| | 100 | 132.7 | 85.4 | 258.8 | 34.1 | 52.1 | 7.6 |
| | 250 | 124.4 | 74.4 | 258.8 | 33.0 | 51.0 | 7.6 |
| Seq-Gen | 25 | 196.7 | 132.2 | 324.8 | 34.5 | 55.9 | 0.0 |
| | 100 | 187.3 | 122.2 | 324.8 | 33.5 | 54.8 | 0.4 |
| | 250 | 177.8 | 110.2 | 324.8 | 32.5 | 53.7 | 0.5 |
| SIMPROT[c] | 250 | – | – | – | – | – | – |

| | | | | |
|---|---|---|---|---|
| iSGv1.0 | $PFAM_{7tm_1}$ | −5.09 | −99.5 | 158.8 |
| ROSE | $PFAM_{7tm_1}$ | −31.47 | −114.9 | 97.7 |
| Seq-Gen | $PFAM_{7tm_1}$ | −7.18 | −129.2 | 160.1 |
| SIMPROT[c] | $PFAM_{7tm_1}$ | – | – | – |

[a] The average (ave), minimum (min), and maximum (max) bit scores from each set of the top hits.

[b] The average % identity (ID), similarity (SIM), and gaps (GAP) obtained from each set of the top hit alignments.

[c] SIMPROT obtained no hits in BLAST and PFAM searches to vertebrate olfactory receptors.

For simulated lipocalin proteins, BlastP and PFAM search did not produce as many significant hits as we observed with GPCR sequences, which implies the difficulty in simulating beta-strand proteins. As shown in Table 2.3, the BlastP results

were comparable among the simulation methods except for SIMPROT. Only about two lipocalins were found per simulated sequence query. Interestingly, slightly more different lipocalin sequences were found among simulation sets produced by iSGv1.0 than among those produced by ROSE and Seq-Gen. This is probably a result of the MSA root sequence option in iSGv1.0 changing the root sequences between data sets. Against the PFAM Lipocalin profile HMM entry, iSGv1.0 had the lowest scores. However, iSGv1.0 produced the highest number of sequences (33.62%) that showed a hit to the lipocalin profile, compared with ROSE and Seq-Gen (25.68% and 22.41%, respectively). Again, simulated sequences by SIMPROT did not find any lipocalin profile HMM.

Note that, as described earlier, the root sequences used for ROSE and Seq-Gen were constructed by the "consensus method" we incorporated in iSGv1.0. In this method, a consensus sequence was constructed based on the reference MSA and this sequence was used as the root sequence for ROSE and Seq-Gen. This may explain the similar performances observed by BlastP and PFAM among iSGv1.0, ROSE, and Seq-Gen. Neither ROSE nor Seq-Gen in their original methods included this capability.

#### 2.2.1.8 Phylogenetic Analysis with Parametric Bootstrap

Neighbor-Joining phylogenies were reconstructed based on the T-Coffee as well as the true MSAs of simulated sequences. For both GPCR and lipocalin simulations using the parameters described earlier, the topologies of the consensus trees obtained from all four methods were identical to the original phylogenies reconstructed from the template MSAs, and all nodes were supported with 99% or better bootstrap values. To explore the effect of indels on phylogenetic reconstruction, we simulated another 100 data sets of GPCRs and lipocalins using iSGv1.0, with twice higher substitution rates

Table 2.3: The BLAST and PFAM statistics of lipocalins when using simulated sequences from iSGv1.0, Seq-Gen, SIMPROT and ROSE as the query sequence.

| Method | Hits[a] | Unique hits[b] | $\text{PFAM}_{\text{Lipocalin}}$ avg. score | % hits[c] |
|---|---|---|---|---|
| iSGv1.0 | 8.06 (2.22) | 60 | −30.12 | 33.62 |
| ROSE | 7.29 (2.25) | 56 | −18.34 | 25.86 |
| Seq-Gen | 7.90 (2.31) | 56 | −14.14 | 22.41 |
| SIMPROT | 5.32 (0.00) | 0 | – | – |

[a] The average number of BLAST hits for each sequence, with the average number of lipocalin hits in parentheses.
[b] The total number of **unique** lipocalins found from the set of all sequences.
[c] The percentage of sequences from the set of all sequences that are identified as members of the Lipocalin family as parameterized by PFAM.

and with and without indels as shown in Table 2.4. When substitution rates were not changed ("Substitution rate ×1"), indels did not significantly affect the phylogenetic reconstructions. However, when the substitution rates were doubled ("Substitution rate ×2"), for both GPCR and lipocalin simulations, the branch supports became lower when indels were incorporated in the sequence evolution. For example, the number of internal branches supported by 90% or higher was nine for the lipocalin phylogeny based on T-Coffee alignments when indel was not incorporated. However, the same number decreased to only one when indels were incorporated. Similar results were obtained for GPCR simulations although the effect was less drastic. The average branch support value (80.2%) for simulated lipocalin phylogenies using T-Coffee MSAs with indels was 10 percent lower than when no indel was incorporated. Note also that when indels were incorporated, phylogenies based on T-Coffee alignments were less supported than those based on the true MSAs. Phylogenies based on simulated GPCR sequences were consistently more supported than those based on

simulated lipocalin sequences, and incorporating indels showed only small decreases in branch supports. This is because the lipocalin template sequences are more diverged (the average pairwise sequence identity = 13.7%) than GPCR template sequences (the average pairwise sequence identity = 35.4%). Thus, our simulations generated more diverged lipocalin-like sequences than GPCR-like sequences.

Table 2.4: Branch support range for the phylogenetic trees obtained by parametric bootstrap using iSGv1.0 simulated datasets of GPCRs and lipocalins.

| | Substitution rate × 1[a] | | | | Substitution rate × 2[a] | | | |
|---|---|---|---|---|---|---|---|---|
| | indel = 0[a] | | indel ×1[a] | | indel = 0[a] | | indel ×1[a] | |
| % support | *lipo*[b] | *GPCR*[b] | *lipo*[b] | *GPCR*[b] | *lipo*[b] | *GPCR*[b] | *lipo*[b] | *GPCR*[b] |
| $x = 100$ | 14(14) | 25(25) | 11(12) | 25(26) | -(3) | 22(21) | -(-) | 15(18) |
| $90 \leq x < 100$ | 6(6) | 1(1) | 9(8) | 1(-) | 9(12) | 4(5) | 1(15) | 11(8) |
| $80 \leq x < 90$ | -(-) | -(-) | -(-) | -(-) | 5(1) | -(-) | 7(1) | -(-) |
| $70 \leq x < 80$ | -(-) | -(-) | -(-) | -(-) | 1(-) | -(-) | 5(2) | -(-) |
| $x < 70$ | -(-) | -(-) | -(-) | -(-) | 5(4) | -(-) | 7(2) | -(-) |
| Average support | 98.9 (99.0) | 100 (100) | 98.6 (98.8) | 100 (100) | 80.2 (90.4) | 99.5 (99.5) | 70.9 (86.4) | 98.9 (99.7) |

[a] Substitution and indel rates are those given in the template alignments for both lipocalins and GPCRs.

[b] The number of nodes of a given confidence in the phylogenies constructed using the true MSAs (in parentheses) and the T-COFFEE MSAs. A '-' signifies that 0 nodes fell within the corresponding confidence range.

## 2.3 Conclusion

iSGv1.0 is a new method for simulating protein sequence evolution and generating realistic protein families. It has a unique ability to simulate heterogeneous evolution of multidomain protein families in addition to the incorporation of indel evolution. iSGv1.0 is highly flexible and various sequence features can be easily included in the simulation. For example, iSGv1.0 allows highly variable regions in the sequence although still maintaining low indel rates. This is useful in simulating motifs such as *CXXC* in thioredoxin-fold proteins or zinc fingers. This is not possible in other simulation methods. The use of a template MSA is also a unique strength of iSGv1.0. With these new functions, iSGv1.0 can be used effectively for testing the performance of various molecular evolution and bioinformatics methods when they are applied to extremely divergent heterogeneous protein analysis. Because each subsequence can be simulated with a different evolutionary tree, lateral gene transfer and recombination detection methodologies can also be examined. For protein families with well-known features, iSGv1.0 can be used to generate synthetic model sequences. Such model sequences can be used to search their candidate members from databases.

# Chapter 3

# Biological Sequence Simulation for Testing Complex Evolutionary Hypotheses: indel-Seq-Gen version 2.0

In this study, we upgrade iSGv1.0 to indel-Seq-Gen version 2.0 (iSGv2.0) [86] by (i) further improving the realism of biological sequence evolution through the introduction of motif conservation using PROSITE-like regular expressions and lineage-specific evolution, and (ii) incorporating the DNA substitution engine of Seq-Gen to add both coding and non-coding DNA sequence simulation. We introduce novel functional constraint enforcement in sequence simulation and formalize how these constraints change the modeling of substitutions, insertions, and deletions (their probabilities of occurrence and placement). We demonstrate a fundamental flaw in simulation of indel processes in many of the current simulation algorithms, and perform a comparative analysis of the indel schemes among these methods. In iSGv2.0, we introduce

our solution to this problem by incorporating indel simulation in discrete evolutionary steps. The output of iSGv2.0 includes true MSAs and information on each indel event including the relative timing and location on the branch (event tracking). iSGv2.0 allows restrictions on minimum and maximum lengths of subsequences by constraining indel events, as is often the case for protein regions with secondary structures. Conservation of folds, as well as motif conservation/gain along different lineages, will be useful to simulate protein superfamily evolution. In addition to the ability to conserve subsequence lengths in DNA sequences, exon-intron structure can also be incorporated to coding-sequence simulation.

iSGv2.0 is the first tool that is capable of simulating complex substitution and indel processes in constrained evolutionary scenarios. iSGv2.0 incorporates coding DNA, non-coding DNA and protein simulation. It allows for testing problems such as phylogenetic reconstruction, functional-site inference, joint estimation of alignment and phylogeny, and multiple sequence alignments. As an example of a complex evolutionary scenario, we present a simulation of calycin protein superfamily evolution.

## 3.1 Methods

We first describe the discrete evolution paradigm introduced in iSGv2.0, along with the implications for substitution and indel evolution. We formalize the sequence representation for simulating evolutionary events such as substitutions and indels for a functionally constrained sequence. We then describe iSGv2.0's other novel mechanisms: (1) lineage-specific models, (2) site-specific functional constraints, (3) coding DNA-sequence simulation, and (4) indel-event tracking.

### 3.1.1 Discrete Evolution

The most fundamental structure needed for sequence simulation is the guide tree, which specifies the branching order and the expected number of substitutions that will occur from an ancestral sequence to its descendant. Substitution processes are generally modeled over continuous time, allowing multiple substitutions at the same site. No established model exists for insertions and deletions. Current sequence simulation methods introduce indels in a continuous fashion [84, 64, 71, 87, 36], even though insertions and deletions alter the sequence length, as shown in Figure 3.1. In order to keep the indel rate constant along the branch, the number of expected indels needs to be recalculated based on the sequence length after each event. The current continuous model uses the same rate no matter how many positions are inserted or deleted along the branch. Consequently, the number of events can be under- or overestimated, which in turn incorrectly decreases or increases the indel rate after insertion or deletion events, respectively, for the remainder of the branch until the probability of an indel event is recalculated based on the length of the descendant sequence at the next node. The order and time of events within the branch are also unknown in the continuous model of sequence evolution, since the expected numbers of substitutions and indels are estimated based on the entire branch length.

To minimize the effects of sequence length changes and to allow for event tracking (described later in this section), iSGv2.0 simulates sequence evolution with discrete steps using a sufficiently small step size. The step size $\epsilon$ is defined as:

$$\epsilon = \frac{\text{max_path}}{2^{10+c}}, \tag{3.1}$$

where max_path is the maximum length of the root-to-tip paths in the guide tree. The constant $c$ is set such that $\epsilon$ is less than 0.01 substitutions per site or smaller

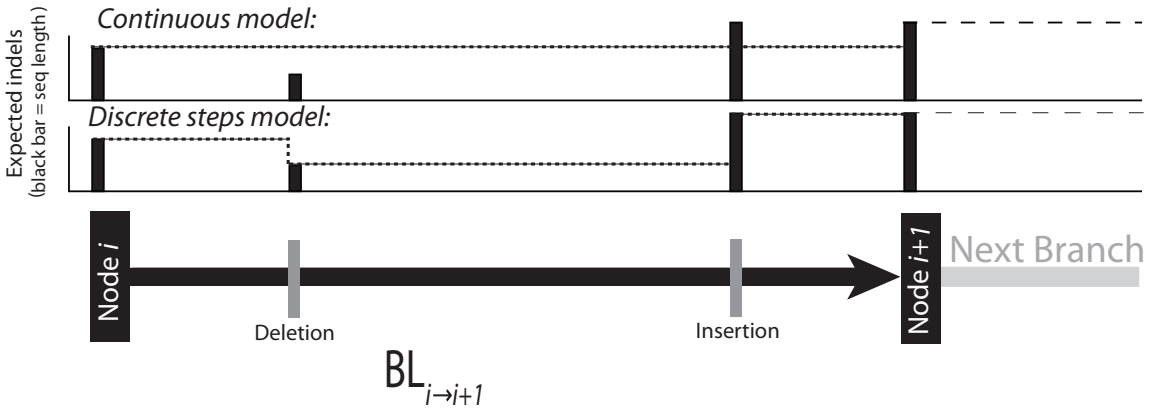



Figure 3.1: The continuous and discrete-steps models of indel events. The continuous model calculates the expected number of indel events based on sequence length at node $i$, and uses this same value throughout the branch length, $\mathrm{BL}_{i \to i+1}$. This causes either over- or underestimating the number of indels along the branch until recalculating the expected number of indel events at node $i + 1$. The discrete-steps model reduces the impact of this by recalculating the expected number of indel events based on the sequence length after each such event.

than the minimum branch length. If the minimum branch length is less than 0.00001, which is the minimum value $\epsilon$ can take, it is considered to be a zero branch length. We call this simulation method the *discrete evolutionary steps* (DES) model.

#### 3.1.1.1 Substitution and indel models

The branch lengths given in the guide tree determine the rates of evolution, which are expressed as the number of substitutions per site for the branch and are introduced based on models of continuous substitution evolution processes: *e.g.*, JTT [44], PAM [19], or BLOSUM [38] for proteins, and HKY [37] or GTR [104] for nucleotides. In the current simulation methods, insertions and deletions are also introduced in a continuous fashion. As mentioned earlier, this continuous method assumes that the

length of the sequence will remain the same during the evolution along the branch (Figure 3.1). This is clearly incompatible with insertion and deletion processes, and is the primary motivation for adopting the DES model. In the following section, we formalize the model of insertions and deletions with respect to the sequence and functional constraints, which will make clear the flaw in the indel representation used in current methods.

#### 3.1.1.2 Formalization of substitution and indel processes

Note that although we refer to our model as "discrete," our substitution processes are simulated using the continuous evolutionary models described above, the difference being that substitutions are simulated in multiple $\epsilon$-sized steps. With respect to insertion and deletion processes, most simulation methods treat them similarly. However, insertions, deletions, and substitutions all work differently with respect to the sequence and functional constraints placed on them. One major difference between insertion and deletion processes is that insertions occur *between* sites whereas deletions occur *on* sites. This fundamental difference not only affects the number of sites that can accept a deletion versus an insertion, but also introduces maximum and minimum length requirements to subsequences in order to enforce possible selective constraints on such subsequence length. This in turn restricts the number of acceptable deletion and insertion lengths in subsequences. Figure 3.2 specifies the model of these constraints for realistic sequence evolution, where the positions $X_3 \cdots X_6$, for example, approximate the $CXXC$ motif of Thioredoxin-fold proteins [16]. While amino acids $X_3$ and $X_6$ can be neither changed nor deleted, amino acids and can be substituted but cannot be deleted. Furthermore, the length from $X_3$ to $X_6$ (4 residues) is constrained and no indel is allowed in this region.

#### 3.1.1.3 Novel indel characterization

Indel modeling requires four parameters: $\Lambda = \{P_{ins}, P_{del}, \lambda_{ins}, \lambda_{del}\}$, where $P_{ins}$ and $P_{del}$ are the probabilities of an insertion and a deletion, respectively, and $\lambda_{ins}$ and $\lambda_{del}$ are the length probability distributions defined as:

$$\lambda = \begin{cases} f(x) & x \in \{1, 2, \cdots, x_{max}\} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $x$ is the number of residues and $x_{max}$ is the maximum insertion and deletion size. $\lambda_{del}$ is defined similarly as $\lambda_{ins}$. For convenience, we assume $\lambda_{ins} = \lambda_{del}$ for the remainder of this section, although iSGv2.0 does allow for $\lambda_{ins}$ and $\lambda_{del}$ to be different. $f(x)$ is the probability density function of indel lengths.

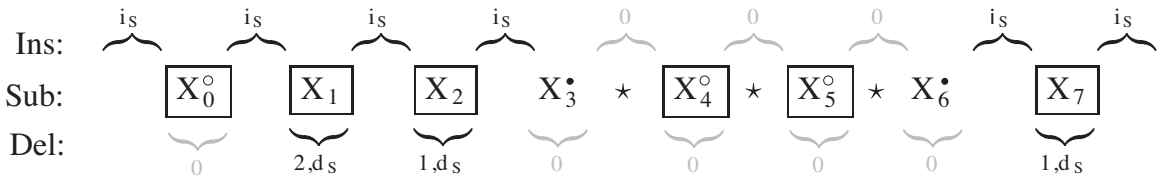
#### 3.1.1.4 Simulating indel occurrence

As shown in Figure 3.2, the maximum number of sites that potentially accept insertions is equal to the number of positions plus 1. Therefore, for an unconstrained sequence at node $i$ with $N(i)$ residues, the number of sites that accept insertions, $N_{ins}(i)$, is $N(i) + 1$, while the maximum number of sites that potentially accept deletions, $N_{del}(i)$, is $N(i)$.

The expected number of residues at node $N(i+1)$ is calculated as:

$$\mathbb{E}[N(i+1)] = N(i) + BL_{i \to i+1} \times (N_{ins}(i) \times P_{ins} - N_{del}(i) \times P_{del}), \tag{3.3}$$

where $BL_{i \to i+1}$ is the branch length from $i$ to $i+1$, and $N_{ins}(i) = N(i)+1$, $N_{del}(i) = N(i)$. $BL_{i \to i+1} \times N_{ins}(i) \times P_{ins}$ and $BL_{i \to i+1} \times N_{del}(i) \times P_{del}$ are the expected numbers of insertions and deletions on the branch $i \to i+1$, respectively. Note that in Equation 3.3, each time an indel event occurs $N_{ins}(i)$ and $N_{del}(i)$ fluctuate, in turn




Symbols and their constraints imposed on sequence sites.

| Symbol | Event Constrained | Description |
|---|---|---|
| $X^\circ$ | Deletion | Position *X* cannot be deleted |
| $X^\bullet$ | Deletion, Substitution | Position *X* can neither be chaged nor removed |
| $X_i \star X_{i+1}$ | Deletion, Insertion | Length-dependent constrained positions; events that change the subsequence length cannot occur between positions linked by $\star$'s |

Figure 3.2: A sequence model that includes substitutions (Sub), insertions (Ins), and deletions (Del) for a length-constrained subsequence $S$ ($X_0 \cdots X_7$, where $X_i$ is the $i$th residue of the sequence). The description of the symbols used and their effects is listed in the table below. For example, positions $X_3 \cdots X_7$ are conserved in a way akin to the $CXXC$ motif of the Thioredoxin sequence motif [16]. The maximum lengths of insertions are shown above the sequence ranging from 0 (no insertion), 1, 2, ... to $i_S$ (upper bound of the subsequence length). The maximum lengths of deletions are determined by either (1) the number of sequence positions to the first deletion-constrained position (2 for $X_i$ in this figure) or $d_S$, defined as the number of positions that can be deleted before reaching the minimum subsequence length.

changing the expectation of future indel events for the remainder of $BL_{i\to i+1}$. For brevity, hereafter we avoid using the node index ($i$) unless it is necessary. Thus, for example, we use $N_{ins}$ instead of $N_{ins}(i)$.

With constraints as shown in Figure 3.2, the numbers of sites available for insertion ($N'_{ins}$) and deletion ($N'_{del}$) are subject to the constraints $N'_{del} \leq N_{del} = N$ and $N'_{ins} \leq N_{ins} = N + 1$. Therefore, under these constraints, Equation 3.3 becomes:

$$\mathbb{E}[N(i+1)] = N(i) + BL_{i\to i+1} \times (N'_{ins} \times P_{ins} - N'_{del} \times P_{del}), \tag{3.4}$$

Most current simulation methods do not correct for sequence length fluctuation

during the evolution with indels, which causes either the underestimation or overestimation of the number of events that will occur for the remainder of the branch as illustrated in Figure 3.1. In Results and Discussion, we examine the consequences of these oversights.

### 3.1.2 Lineage-specific evolution

iSGv2.0 accepts guide trees in Newick format with clade labels. Specifying clades allows lineage-specific parameters to be set. The sequence parameters (character frequency, proportion of invariable sites, site rates, and substitution matrix) and indel parameters (maximum indel size, $P_{ins}$, $P_{del}$, $\lambda_{ins}$, and $\lambda_{del}$) can be changed among subtrees (clades). iSGv2.0 also provides a lineage-specific flag for a lineage evolving as a pseudogene. With this flag, all constraints to the sequence positions, *i.e.*, invariable array, positional $\gamma$ parameters, and codon rates, are removed. It causes the lineage to evolve with a uniform rate and unconstrained for indel events across all sites.

### 3.1.3 Functional constraint modeling

#### 3.1.3.1 Site-specific constraints

iSGv2.0 introduces site-specific conservation using regular expression patterns found in PROSITE [79]. The quaternary invariable array introduced in iSGv1.0 [87] is also retained because of its simplicity of representation. Since motifs are preserved along lineages, sites that correspond with the potential motif in the ancestral sequences still carry the length constraints of the motif, *i.e.*, motifs cannot be gained from insertions and potential motif sites cannot be deleted. While indels are constrained on these sites, substitutions are freely accepted until the sites are accepted as the motif. When the site becomes a part of the motif, which occurs when the site is mutated into a

motif satisfying residue, the site becomes constrained based on the patterns specified in the motif. These constraints on a motif, by definition, cause a slower evolutionary rate within the motif region. Thus, iSGv2.0 compensates for the slower evolutionary rate by increasing the substitution rate in the partition that includes the motifs so that the resultant sequences will evolve at the expected rate, on average, based on the input branch length. For a motif with $k$ characters, $m = m_0 \cdots m_{k-1}$, we calculate the average rate of substitution rejection at each motif site, $\hat{g}_n$, as follows:

$$\hat{g}_n = \frac{\sum_{i \in a_n} \left(1 - \sum_{j \in a_n} s_{ij}\right)}{\mid a_n \mid} \tag{3.5}$$

where $a_n$ is the set of acceptable residues for the motif position $n$, $\mid a_n \mid$ is the number of acceptable residues in the set, and $s_{ij}$ is the probability of substitution of residue $i$ to residue $j$ (for the chosen substitution matrix and character frequencies) over the discrete evolutionary step size $\epsilon$. The term $(1 - \sum_{j \in a_n} s_{ij})$ is the probability of rejected substitutions from the residue $i$. We then define $g_n = \sum_{i=0}^{k-1} \hat{g}_n$, the amount of reduction in evolutionary rate in the motif. Since the expected number of substitutions in an unconstrained sequence with $N$ characters along the branch length $BL$ is $N \times BL$, adjusting for the reduced evolutionary rate in the motif, we calculate $BL'$ as follows:

$$BL' = \frac{g_n + N \times BL}{N} = \frac{g_n}{N} + BL. \tag{3.6}$$

#### 3.1.3.2 Subsequence length constraints

Protein family sequences are often composed of a set of domains that define the folding pattern of member sequences. Functional domains are often under length constraints whose violation could be detrimental to protein function, such as the destabilization

of tertiary structure, improper folding, or removal of functionally important regions. iSGv2.0 represents these constraints through the introduction of a sequence *template*. The template specifies both the minimum and maximum number of residues or nucleotides that can occupy a region. Such constraints limit the number of insertion or deletion events. An example of sequence templates is given in Figure 3.3.

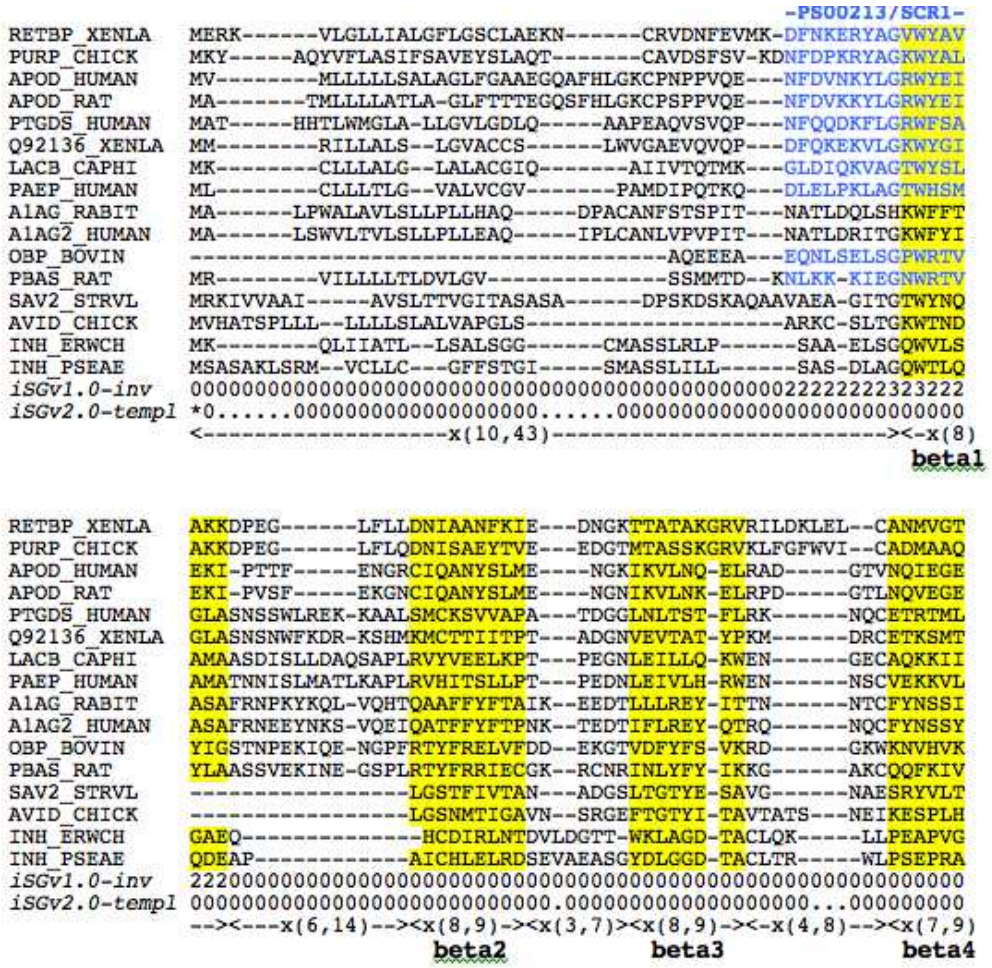



Figure 3.3: *Continued on next page.*

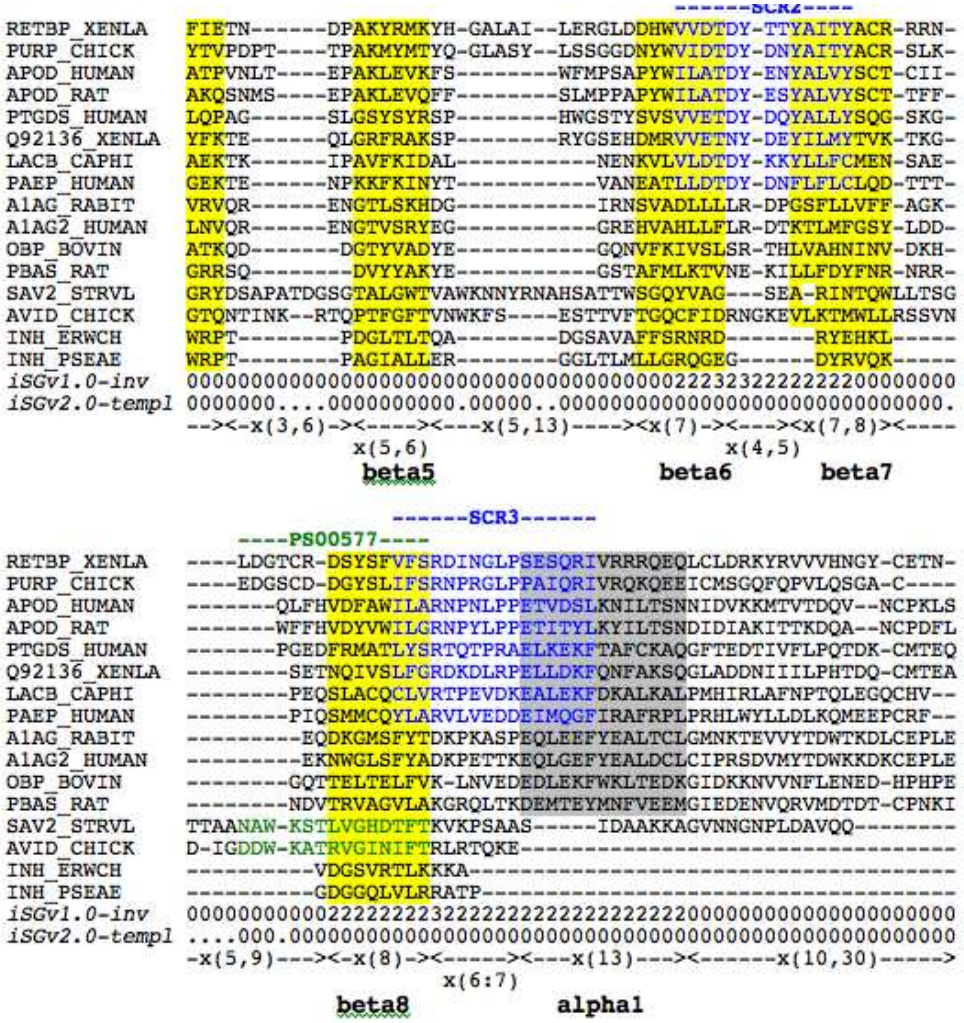



Figure 3.3: *Continued on next page.*

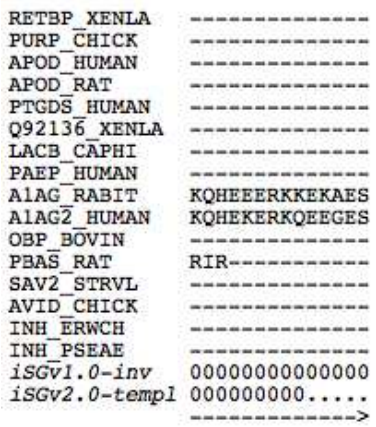



Figure 3.3: The MSA input used for the calycin superfamily simulation with iSGv1.0 and iSGv2.0. Yellow highlighted regions in the MSA are beta strands, gray highlighted regions are alpha helices, and unhighlighted regions are coils. Conserved regions (PS00213, SCR1, SCR2, PS00577, and SCR3) are marked above the alignment, and the characters of the sequences corresponding to the motifs are shown in the same colors as the headings. The quaternary invariable array of iSGv1.0 (*iSGv1.0-inv*) and the iSGv2.0 template (*iSGv2.0-templ*) are listed below the alignment. Below the template line, x(min,max) shows the maximum and minimum sizes for each region. The conserved secondary structure regions (beta1 - beta8 and alpha1) corresponding to these regions are also marked below the template line.

### 3.1.4 Nucleotide sequence simulation

Coding and non-coding nucleotide sequence simulation has been added to iSGv2.0 using the substitution engine Seq-Gen [69]. In the coding sequence simulation, exons and introns can be specified as partitions. Exons are influenced by different rates for codon positions and restriction of stop codon formation. Introns can split codons (i.e., phase 1 and phase 2 introns). In introns, elements (such as the lariat formation sites) can be controlled by the quaternary invariable array and through motifs. Insertion and deletion length distributions for exons can be set to zero for all lengths not divisible by 3 to avoid introducing frame-shifting indel mutations. While insertions and deletions in exons can be restricted to be between codons, such restrictions are not mandatory.

### 3.1.5 Event tracking

One benefit introduced with the DES model is the ability to track indel events by the time of the events, affected taxa, event type (insertion or deletion), and indel length. The positions of indels can be reported in the true alignment. Along with the DES model, iSGv2.0 has added a new presentation method, *time-relative steps* (TRS). With the TRS presentation, the tree is rescaled relative to the time. The resultant tree is "ultrametric-like," having equal height (time) from the root to tips, which allows the mapping of all indel events with respect to the relative time of occurrence. Figure 3.4 illustrates this TRS presentation. With the TRS presentation, events are reported as an ordered list based on the relative time of occurrence as shown in Figure 3.4C. When the TRS presentation is not used, events are listed by partition.

(A)

Discrete Evolutionary Steps (DES)

```
(
 (
  (INH_PSEAE:1.08875,INH_ERWCH:0.31828
  )MPI:2.00196,
  (AVID_CHICK:0.64976,SAV2_STRVL:1.00582
  )avidin:0.64347
 )calycin:0.8423,
 (
  (
   (
    (
     (
      (PURP_CHICK:0.41463,RETBP_XENLA:0.4985
      )RETB:1.0926,
      (APOD_RAT:0.33545,APOD_HUMAN:0.03583
      )ApoD:1.09025
     )kernel_2:0.7268,
     (Q92136_XENLA:0.16702,PTGDS_HUMAN:0.84441
     )PTDG:0.59918
    )kernel_1:0.2931,
    (PAEP_HUMAN:0.53679,LACB_CAPHI:0.38169
    )beta:0.73169
   )kernel:0.2646,
   (A1AG2_HUMAN:0.19957,A1AG_RABIT:0.40769
   )alpha:2.15913
  )alphakernel:0.2401,
  (PBAS_RAT:0.87889,OBP_BOVIN:0.98333
  )OBP:0.6229
 )lipocalin:0.8424
);
```

Time Relative Steps (TRS)

```
(
 (
  (INH_PSEAE:1.09389,INH_ERWCH:1.09389
  )MPI:2.01384,
  (AVID_CHICK:1.89401,SAV2_STRVL:1.89401
  )avidin:1.21371
 )calycin:0.850373,
 (
  (
   (
    (
     (
      (PURP_CHICK:0.498628,RETBP_XENLA:0.498628
      )RETB:1.09389,
      (APOD_RAT:0.371072,APOD_HUMAN:0.371072
      )ApoD:1.21758
     )kernel_2:0.726682,
     (Q92136_XENLA:1.35287,PTGDS_HUMAN:1.35287
     )PTDG:0.962468
    )kernel_1:0.293765,
    (PAEP_HUMAN:1.10162,LACB_CAPHI:1.10162
    )beta:1.50748
   )kernel:0.262843,
   (A1AG2_HUMAN:0.456109,A1AG_RABIT:0.456109
   )alpha:2.41583
  )alphakernel:0.243516,
  (PBAS_RAT:1.90561,OBP_BOVIN:1.90561
  )OBP:1.20985
 )lipocalin:0.84262
);
```

Figure 3.4: *Continued on next page.*

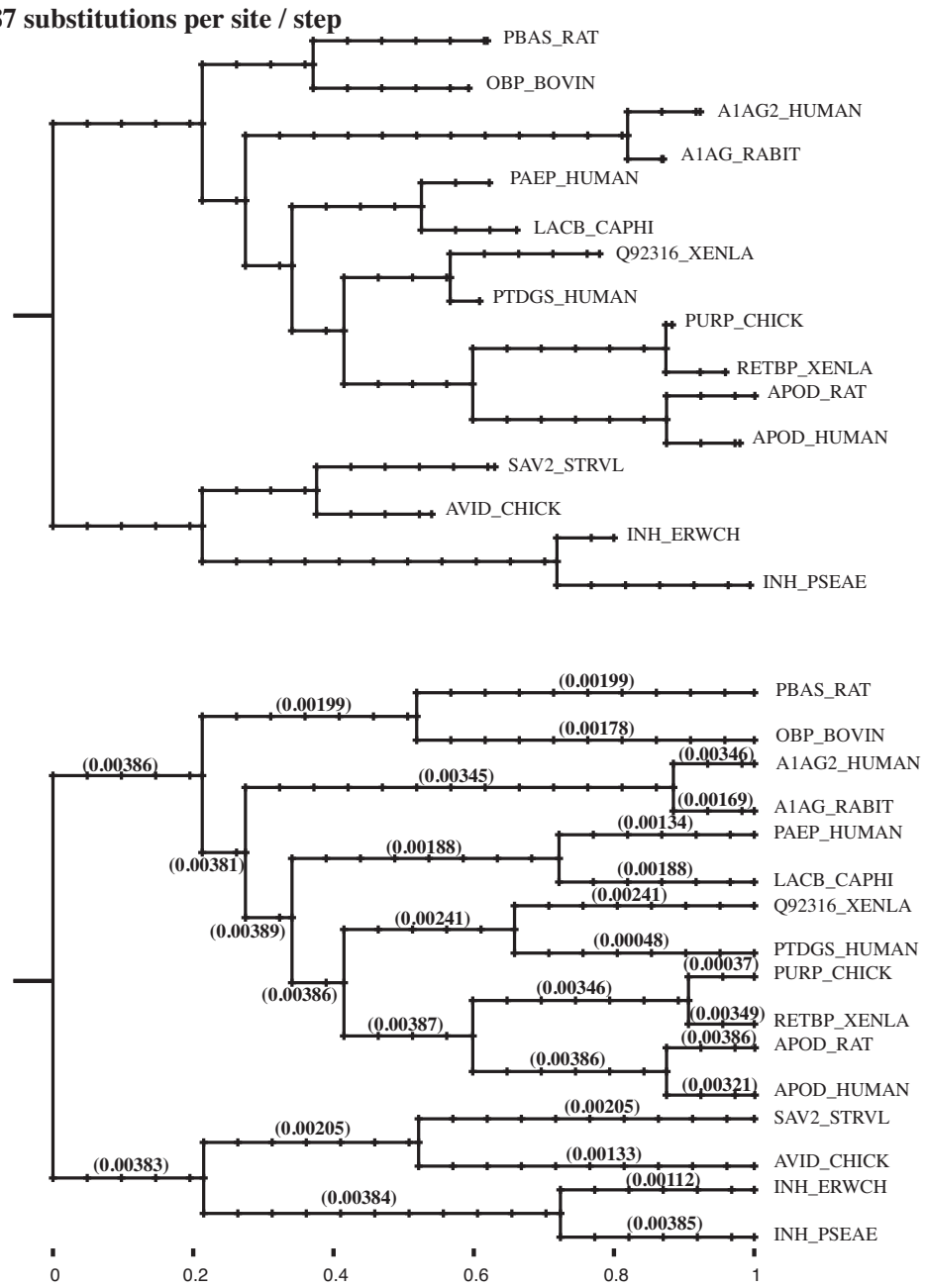



Figure 3.4: *Continued on next page.*

(C) Event tracking format:
[Event ID, (D)eletion or (I)nsertion, Relative time of occurrence, Affected taxa, location in multiple alignment]

[10,D,0.410156,1100000000000000,6,7:8:9:10:11:12]
[11,D,0.414062,1100000000000000,1,53]
[69,D,0.416992,0000000000001100,2,6:7]
[70,D,0.418945,0000000000001100,1,11]
[39,I,0.419922,0000111100000000,2,13:14]

```
INH_PSEAE E----V-------KLY--------RFARGYYCNF-GI--AVSPRQRLIVAG--RACDPF
INH_ERWCH P----T-------KVY--------RFADASFCSF-QN--TLS-KSLSRVNG--RSCHLA
AVID_CHIC K----EY---LL--MFF--------QG---ITFSD-LC--AI----------TVRETK--
SAV2_STRV E----ERMYILL--PMF--------QC---NPWGE-TC--FS----------MVIVDNQD
PURP_CHIC D----SAIHATLRFVMF--------FD---TFKGK-AL--------------GGEPNCPN
RETBP_XEN I----SAVHRSLRPVIL--------FD---QLKNK-AT--QK----------SGVVTHPD
APOD_RAT  G----SAVDIVHLPVMH--------LD---DSEPP-------------------TSSPGD
APOD_HUMA G----SAVDIMPQPVMV--------LE---SCEPA-AL--PD----------IGASAPGN
Q92136_XE G----TALDAII--IVS--------ID---LFGGR-NL--NN----------APTQSKLN
PTGDS_HUM G----EVLNTVY--FVS--------ID---MEGAN-NL--QK----------IPAQNKTN
PAEP_HUMA D----TEDELRV--MSG--------LD---LYRGH-RQ--SQ----------VHATLELE
LACB_CAPH D----LDEDLVA--MGG--------LP---LYRGH-KQ--SK----------VHVTQQVN
A1AG2_HUM N------DSV-G--VLYYVPLELTYLD---TV---IKLQCK--------------LMRQ
A1AG_RABI A------DTS-G--LLYNLPLKLAYLE---TV---IRVQSK--------------LMRE
PBAS_RAT  VALCSENRNLYI--MMI--------ID---RHCLL-KG--AL----------VNNKLQLN
OBP_BOVIN Q----LPHVVYI----V--------VH---LRCGT-K-----------------KMELN
```

Figure 3.4: An example of the discrete evolutionary steps and time relative steps representations for the calycin superfamily example. (A) The Newick format trees for each representation. (B) The guide trees as represented in (A), where the top is DES, and the bottom tree is TRS. Each tick mark along the tree represents 50 steps of evolution. The values in parentheses in the TRS tree represent the scaled evolutionary step size. The TRS tree also contains a scale that indicates the relative times of the simulation. (C) An example of indel tracking. All events that occurred between 41% and 42% of the simulation are listed, and the location of each event is marked with the colors matching with those used with event IDs in the true MSA.

### 3.1.6 Implementation

iSGv.2.0 is written in ANSI C++. iSGv2.0 calculates substitution probabilities using the Seq-Gen [69] formulation. Only rooted trees can be used as guide trees. It has been tested on Linux, Mac OS X 10.4–10.5, and also on Windows XP running MinGW (http://www.mingw.org/), MSYS, and GNU `gzip` and `tar`. It is packaged using GNU autotools and should compile on most systems with a standard C++ compiler. The output can be in Phylip, Nexus, and FASTA formats. iSGv2.0 (executables and source codes) and its user manual describing the functionalities are freely available at http://bioinfolab.unl.edu/∼cstrope/iSG/.

### 3.1.7 Indel simulation comparison

We compared seven indel-capable simulation methods, including iSGv1.0 and iSGv2.0. These methods are listed in Table 1.1. Two tests were performed to examine the indel formulation of each method. In the first test, we analyzed the over- and underestimation of insertions and deletions individually for each simulation method using guide trees with varying numbers of internal nodes. In the second test, similar analysis was done, varying the relative rates of insertions versus deletions.

Note that the indel scheme implemented in EvolveAGene3 [36] is very different from other methods. EvolveAGene3 simulates codon evolution based on empirical models obtained from *E. coli*. EvolveAGene3 calculates indel probabilities with two spectra: the first spectrum determines the event to take place, with probabilities to be 0.6284, 0.0744, or 0.2972 for a substitution, insertion, or deletion, respectively. In the second spectrum, EvolveAGene3 determines the indel length, rejecting any event that is not a multiple of 3. For the second spectrum, we summed up the probabilities of all acceptable lengths to obtain single accepting probabilities for insertions

and deletions: 0.144 and 0.261, respectively, and used them for each type of event regardless of the length. "Selection against deletions and insertions" were both set to 1 (no selection). Thus, with EvolveAGene3, the insertion and deletion probabilities are $0.0107 = 0.0744 \times 0.144 \times 1$ and $0.0776 = 0.2972 \times 0.261 \times 1$, respectively ("event probability" $\times$ "accepting probability" $\times$ "selection against insertions or deletions"). We also modified the EvolveAGene3 code to allow frame-shifting indel mutations, in order to create similar indel-generation conditions with other methods.

#### 3.1.7.1 Experimental setup

For our tests, we set the total tree length to be 8 substitutions per site, with an indel rate of 1 insertion or deletion per 50 substitutions. For EvolveAGene3, insertion and deletion rates were 0.535 and 3.88 per 50 substitutions, respectively, as explained above. We simulated with random root sequences of 1000 characters. DNA sequences were generated by DAWG, EvolveAGene3, and MySSP, and protein sequences were generated by ROSE, SIMPROT, iSGv1.0, and iSGv2.0. The difference in character sets (DNA or protein) is of no consequence in our tests for two reasons: (1) we used the same indel length distributions for both sets and (2) we set the probability of insertion and deletion occurrence equally regardless of the character set. Figure 3.5 illustrates the four simple guide trees with varied numbers of internal nodes. At each node the external branch was set to zero length, as shown in the Newick format in Figure 3.5, effectively making it a leaf node, so that the direct effect of the different number of nodes can be examined. We performed two tests: (1) simulating insertions alone or deletions alone and (2) simulating both insertions and deletions with varying relative rates. Test 1 is intended to show the effect of the indel placement paradigms of each sequence simulation method. Test 2 is intended to show the effect of the indel methods when both insertions and deletions are generated. If insertions and deletions

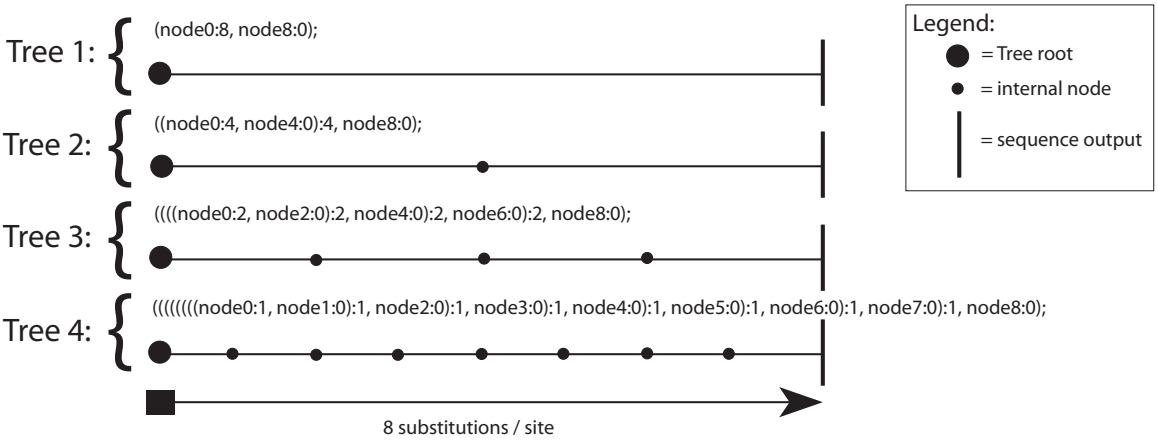



Figure 3.5: The four simple guide trees and their corresponding Newick formats used to test indel simulation schemes. These have 0, 1, 3, and 7 branching points for Trees 1, 2, 3, and 4, respectively. Note that at each branching point (or node), one branch is given a zero length branch, as shown in the Newick format. The total length of the guide tree is set to 8 substitutions per site. Branching points are named from "node0" (at the root) to "node8." During the simulation, sequences are saved at each internal node as well as terminal nodes, and used for indel analysis.

are simulated properly, all simulation runs are expected to return similar numbers of insertions or deletions regardless of the number of internal nodes, since all four guide trees have identical lengths. Differing results between guide trees implies that adding branching points (nodes) affects the remainder of evolution, which is clearly undesirable.

### 3.1.8 Protein superfamily comparison

To present the sequence simulation capabilities of iSGv2.0, we simulated the calycin-superfamily proteins. We performed the simulation using iSGv2.0, ROSE, and iSGv1.0, and compared their simulation results.

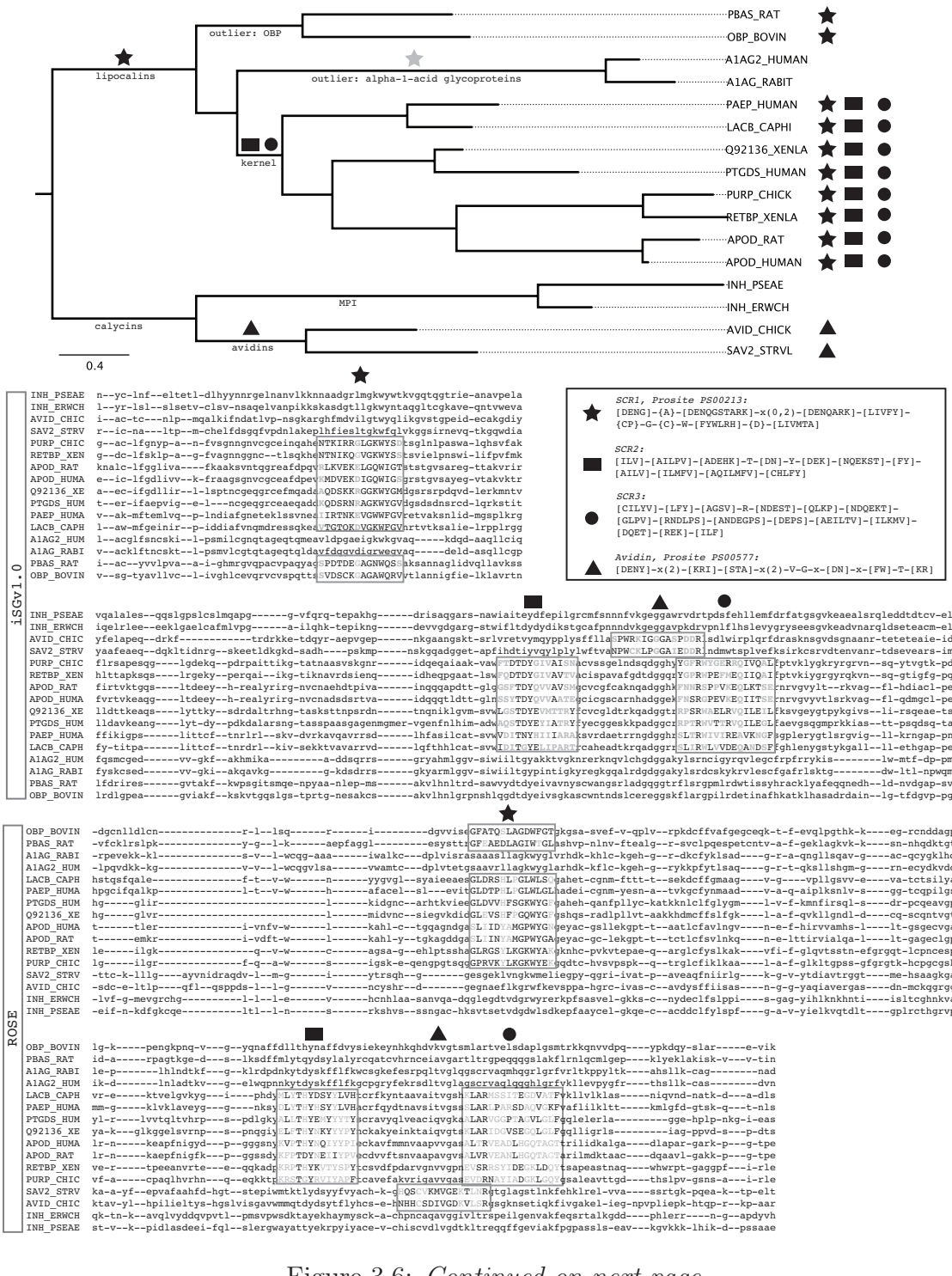



Figure 3.6: *Continued on next page.*




Figure 3.6: Simulation of the calycin protein superfamily using ROSE, iSGv1.0, and iSGv2.0. The phylogeny at the top is the guide tree used for the simulation. The signature motifs for each protein sequence (SCR1, SCR2, SCR3, and avidin) are listed by the UniProt protein IDs. Where the motifs are gained or lost are illustrated on the tree by black or gray symbols, respectively. The regular expression patterns defining these motifs are listed below the tree. In the output alignments, lowercase letters indicate non-motif positions, while uppercase letters belong to motifs. Each motif is boxed in the alignment, and the identity of the motif is given by the corresponding symbols at the top of the alignment.

#### 3.1.8.1 Experimental setup

We sampled two sequences each from the avidins, MPIs, two subfamilies of outlier lipocalins, and four subfamilies of kernel lipocalins. The base alignment of these sequences was obtained using PROMALS3D [66] with manual adjustment. The alignment, along with the annotated beta-strands and motifs, is given in Figure 3.3. Figure 3.6 shows the phylogeny reconstructed using `proml` from Phylip version 3.68 [27]. Based on the phylogeny, we determined the most likely scenario of motif gain and loss as follows: (1) SCR1 was gained before the divergence of the lipocalin family, and subsequently lost in the alpha-1-acid glycoprotein lineage; (2) SCR2 and SCR3 were gained before the divergence of the kernel lipocalin family; and (3) the avidin motif was gained after the avidin family was diverged from the MPI family. SCR1 and the avidin motifs are obtained from PROSITE regular expressions PS00213 and PS00577, respectively [79]. For SCR2 and SCR3, we gathered the motif alignments in the PRINTS database (the lipocalin family motifs 2 and 3; PR00179) [4], and calculated the percentage of each amino acid in each column of the alignment. The regular expression patterns were generated using amino acids with at least 5% representation as the acceptable residue set for each alignment position. Figure 3.6 lists these regular expressions. Using the guide tree shown in Figure 3.6, we simulated the calycin superfamily evolution. We specified the template for the input sequences based on the secondary structures of the sequence, and specified four motifs: SCR1, SCR2, SCR3, and the avidin motif. Since iSGv1.0 and ROSE do not have the ability to conserve specific lineages, we chose to conserve the motifs in the global invariable and the I $+\gamma$ arrays, respectively. We conserved fixed-length regions using the invariable array option that forbids indels between sites and held sites with single acceptable states as invariable for all methods. All specifications are available in Figure 3.3. We

simulated 100 datasets for each method.

#### 3.1.8.2 Indel statistics

To test the minimum and maximum subsequence constaints (template) in iSGv2.0, we flagged insertions and deletions in ROSE and iSGv1.0 that broke minimum and maximum subsequence constraints. In order to do this, the template constraints needed to be introduced to each method. It was possible for iSGv1.0 by simulating within the iSGv2.0 template framework. However, we were unable to incorporate ROSE in the iSGv2.0 framework. For ROSE, to detect template-breaking indels we inspected the true MSA including ancestral sequences. From this alignment, we traversed all root-to-tip paths, examining the regions corresponding to the templated subsequences. When we found an unacceptable number of residues in a region, we counted one template-breaking indel. If the descendant sequences had a region shorter or longer than the corresponding template, we counted another template-breaking indel only if the indel pattern (gap columns) was different from the ancestral sequence.

## 3.2 Results and Discussion

### 3.2.1 Test 1: Insertions alone or deletions alone

Our first test was to run insertion-only and deletion-only simulations. Indel lengths were fixed with 1, 2, 4, and 8 residues or bases. We measured the performance of each method by (1) the length of the "true" MSA for insertions, where the number of sites inserted is equal to the alignment length minus 1000, and (2) the number of characters remaining in the output sequence for deletions.

Figure 3.7 shows the test results. The number of internal nodes in the guide

tree had an adverse effect on the performance of SIMPROT, iSGv1.0, ROSE, and MySSP (only in the case of insertions). As a side effect of their continuous modeling of indels, overestimation of deletions (Figure 3.7A and B) and underestimation of insertions (Figure 3.7C and D) are clearly shown with fewer numbers of internal nodes. These methods calculate the expected number of indel events without adjusting the sequence length when an event occurs. DAWG, iSGv2.0, and EvolveAGene3 show no or very slight effects in indel numbers. For DAWG and iSGv2.0, this is because sequence lengths are adjusted dynamically along the branch. The unaffected results by EvolveAGene3 are likely due to the fact that this method treats branch lengths as the number of mutation-event tests that occur along a branch. For our purpose, we set the branch length to 8000 mutation-event tests (1000-character sequence with each site undergoing 8 substitutions). EvolveAGene3 also forbids overlapping insertions and deletions, effectively reducing the deletion rates with larger deletions. This effect can be seen in Table 3.1, where more characters are left after simulations with larger deletion sizes. Because of the constant insertion and deletion probabilities set in EvolveAGene3 and our removal of codon constraints in EvolveAGene3, the lengths of the alignments are often shorter than other simulation methods.

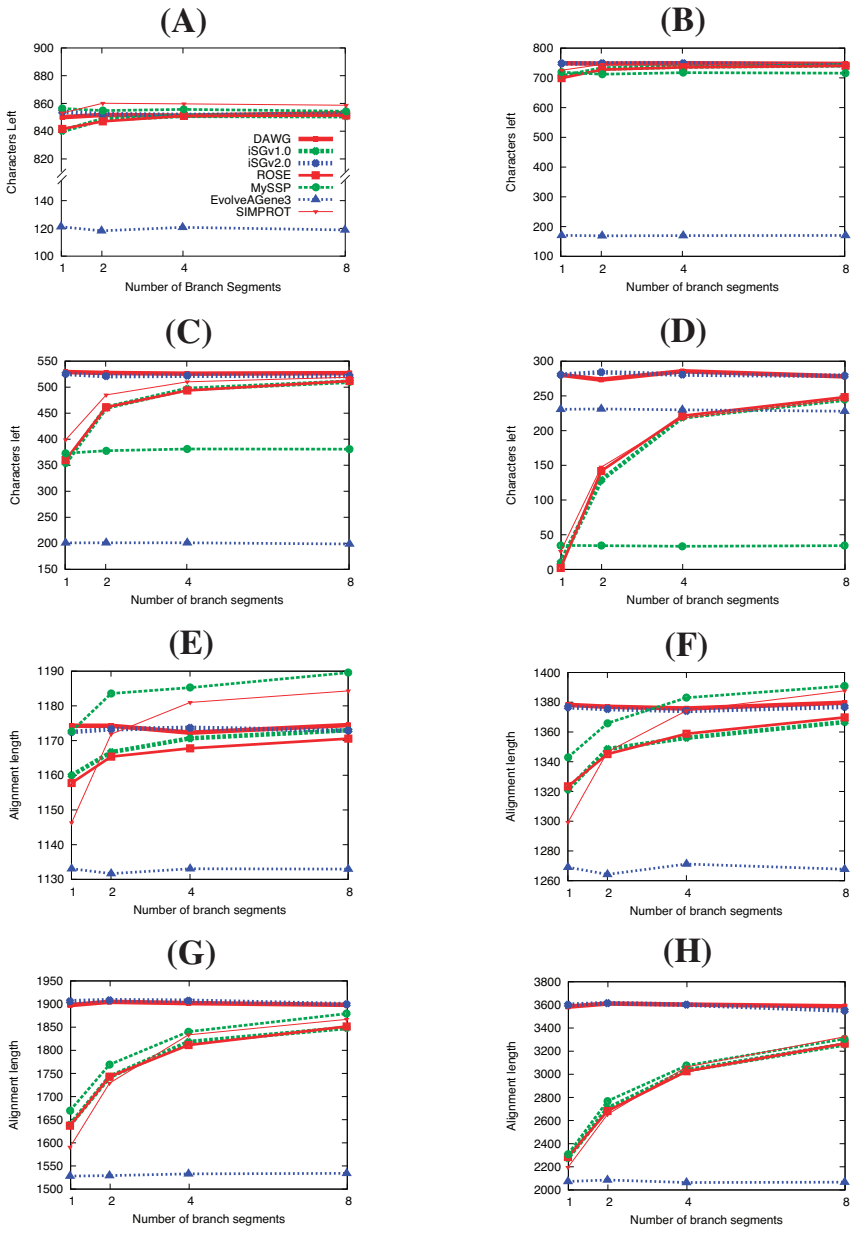



Figure 3.7: Indel simulation performance among the seven methods (Test 1). Correct simulations are expected to produce a plot with a horizontal line. (A) Size-1 deletions, (B) size-2 deletions, (C) size-4 deletions, (D) size-8 deletions, (E) size-1 insertions, (F) size-2 insertions, (G) size-4 insertions, (H) size-8 insertions. The test statistics (characters left for deletion-only tests and true alignment lengths for insertion-only tests) and the coefficients of variation are shown in Table 3.1. The standard deviations for each data point are shown in Table 3.2. The average values obtained from 100 simulations are plotted.

Table 3.1: The data points and coefficient of variation for the plots in Figure 3.7.

| | Method | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DAWG | | iSGv1 | | iSGv2 | | ROSE | | MySSP | | EAG | | SIMPROT | |
| # Branch Segments | ins | del | ins | del | ins | del | ins | del | ins | del | ins | del | ins | del |
| *indel length 1* | | | | | | | | | | | | | | |
| 1 | 1173.3 | 853.2 | 1160.2 | 841.6 | 1172.7 | 853.9 | 1159.2 | 838.6 | 1172.9 | 856.3 | 1133.3 | 121.1 | 1144.3 | 853.2 |
| 2 | 1171.8 | 851.0 | 1166.9 | 847.0 | 1173.5 | 852.1 | 1168.8 | 845.8 | 1183.8 | 854.7 | 1131.9 | 118.2 | 1172.3 | 859.7 |
| 4 | 1174.3 | 853.0 | 1170.9 | 847.5 | 1173.9 | 851.6 | 1170.7 | 847.2 | 1185.5 | 855.6 | 1133.3 | 120.7 | 1180.6 | 859.9 |
| 8 | 1174.0 | 851.9 | 1173.2 | 850.3 | 1173.1 | 851.9 | 1169.0 | 850.6 | 1189.8 | 854.3 | 1133.2 | 118.8 | 1184.8 | 859.9 |
| $\sigma^2/\mu$ | **0.001** | **0.001** | **0.021** | **0.012** | **0.000** | **0.001** | **0.017** | **0.023** | **0.033** | **0.001** | **0.000** | **0.013** | **0.213** | **0.010** |
| *indel length 2* | | | | | | | | | | | | | | |
| 1 | 1377.4 | 726.5 | 1317.2 | 681.3 | 1371.4 | 725.7 | 1323.2 | 681.5 | 1341.8 | 697.5 | 1268.0 | 148.7 | 1300.9 | 700.5 |
| 2 | 1377.7 | 728.5 | 1341.4 | 706.5 | 1376.0 | 725.9 | 1352.3 | 703.5 | 1364.8 | 690.8 | 1263.1 | 147.4 | 1347.0 | 719.8 |
| 4 | 1378.4 | 726.1 | 1364.1 | 715.8 | 1375.8 | 727.1 | 1354.8 | 714.2 | 1382.0 | 695.8 | 1270.1 | 147.9 | 1371.7 | 721.5 |
| 8 | 1382.3 | 727.6 | 1374.6 | 720.9 | 1378.4 | 726.2 | 1368.1 | 719.9 | 1389.8 | 694.1 | 1266.6 | 148.7 | 1389.6 | 731.2 |
| $\sigma^2/\mu$ | **0.003** | **0.001** | **0.362** | **0.329** | **0.005** | **0.000** | **0.199** | **0.306** | **0.247** | **0.009** | **0.005** | **0.002** | **0.821** | **0.172** |
| *indel length 4* | | | | | | | | | | | | | | |
| 1 | 1892.5 | 530.0 | 1638.3 | 357.6 | 1905.3 | 526.3 | 1644.8 | 357.1 | 1669.0 | 373.8 | 1527.4 | 201.5 | 1590.0 | 401.2 |
| 2 | 1903.4 | 524.7 | 1742.0 | 458.6 | 1907.6 | 521.9 | 1747.0 | 461.5 | 1768.6 | 378.6 | 1528.6 | 201.7 | 1729.3 | 486.8 |
| 4 | 1892.4 | 527.1 | 1817.4 | 504.0 | 1906.4 | 522.7 | 1816.4 | 494.8 | 1839.8 | 381.9 | 1532.4 | 201.7 | 1831.5 | 510.1 |
| 8 | 1893.1 | 526.6 | 1847.8 | 506.8 | 1899.0 | 522.0 | 1846.0 | 517.3 | 1878.6 | 381.6 | 1533.4 | 199.3 | 1875.5 | 519.5 |
| $\sigma^2/\mu$ | **0.011** | **0.007** | **3.710** | **7.978** | **0.006** | **0.006** | **3.398** | **8.230** | **3.552** | **0.028** | **0.004** | **0.005** | **6.868** | **4.546** |
| *indel length 8* | | | | | | | | | | | | | | |
| 1 | 3605.4 | 275.4 | 2289.8 | 8.0 | 3601.5 | 282.2 | 2282.7 | 0.0 | 2307.2 | 32.7 | 2070.9 | 228.7 | 2182.5 | 31.4 |
| 2 | 3657.0 | 277.9 | 2710.9 | 136.9 | 3570.2 | 274.7 | 2690.5 | 136.4 | 2766.2 | 32.2 | 2083.8 | 229.2 | 2645.3 | 146.9 |
| 4 | 3604.7 | 276.0 | 3041.1 | 213.6 | 3605.4 | 275.0 | 3041.8 | 213.8 | 3073.5 | 31.4 | 2062.1 | 228.0 | 3076.1 | 214.3 |
| 8 | 3630.1 | 275.0 | 3285.0 | 245.6 | 3619.2 | 282.5 | 3293.4 | 247.0 | 3304.5 | 32.3 | 2065.4 | 226.1 | 3301.4 | 245.5 |
| $\sigma^2/\mu$ | **0.13** | **0.00** | **49.23** | **55.48** | **0.09** | **0.05** | **51.16** | **60.56** | **48.69** | **0.01** | **0.03** | **0.01** | **65.40** | **42.27** |

Table 3.2: Standard deviations for each data point in Figure 3.7.

| | MySSP | | | | EvolveAGene3 | | | | SIMPROT | | | | ROSE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *deletion-only:* | | | | | | | | | | | | | | | | |
| # branch segments | (A) | (B) | (C) | (D) | (A) | (B) | (C) | (D) | (A) | (B) | (C) | (D) | (A) | (B) | (C) | (D) |
| 1 | 13.00 | 26.82 | 53.77 | 21.44 | 28.19 | 18.43 | 24.96 | 27.47 | 12.80 | 23.21 | 42.92 | 32.64 | 11.82 | 24.98 | 49.87 | 0.100 |
| 2 | 12.06 | 24.62 | 51.73 | 18.81 | 30.12 | 17.63 | 23.53 | 28.84 | 10.64 | 20.37 | 34.46 | 50.48 | 12.03 | 22.14 | 35.85 | 58.12 |
| 4 | 14.15 | 29.62 | 55.09 | 18.37 | 30.68 | 18.01 | 24.84 | 28.73 | 12.39 | 20.27 | 27.91 | 39.42 | 11.04 | 17.20 | 34.38 | 46.23 |
| 8 | 12.37 | 29.70 | 59.10 | 18.03 | 31.21 | 17.96 | 25.21 | 28.78 | 10.47 | 21.05 | 37.51 | 42.05 | 10.29 | 21.87 | 37.26 | 39.52 |
| *insertion-only:* | | | | | | | | | | | | | | | | |
| # branch segments | (E) | (F) | (G) | (H) | (E) | (F) | (G) | (H) | (E) | (F) | (G) | (H) | (E) | (F) | (G) | (H) |
| 1 | 13.71 | 30.74 | 54.27 | 114.0 | 12.43 | 20.25 | 45.46 | 92.17 | 12.28 | 22.83 | 59.06 | 95.35 | 12.03 | 24.24 | 46.99 | 106.6 |
| 2 | 13.96 | 34.04 | 72.40 | 159.3 | 12.23 | 24.06 | 41.94 | 82.51 | 17.58 | 29.18 | 58.49 | 137.5 | 12.09 | 26.03 | 59.69 | 148.1 |
| 4 | 14.16 | 39.74 | 82.19 | 172.8 | 11.60 | 21.81 | 47.86 | 82.72 | 15.77 | 34.01 | 68.70 | 199.0 | 13.54 | 30.70 | 61.85 | 181.6 |
| 8 | 16.93 | 35.02 | 80.22 | 237.5 | 11.68 | 22.18 | 48.60 | 97.57 | 17.77 | 32.05 | 77.90 | 230.8 | 13.30 | 28.44 | 78.15 | 257.7 |

| | iSGv1.0 | | | | DAWG | | | | iSGv2.0 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *deletion-only:* | | | | | | | | | | | | |
| # branch segments | (A) | (B) | (C) | (D) | (A) | (B) | (C) | (D) | (A) | (B) | (C) | (D) |
| 1 | 10.68 | 24.65 | 44.73 | 0.000 | 11.55 | 19.92 | 33.88 | 40.38 | 11.72 | 19.34 | 30.03 | 40.30 |
| 2 | 10.46 | 20.52 | 38.43 | 46.00 | 11.45 | 18.09 | 31.08 | 32.81 | 13.00 | 20.31 | 28.95 | 43.40 |
| 4 | 11.52 | 21.40 | 34.19 | 42.49 | 11.36 | 19.77 | 31.78 | 43.19 | 10.44 | 20.92 | 31.77 | 38.81 |
| 8 | 10.56 | 19.21 | 35.89 | 35.86 | 11.27 | 17.59 | 30.59 | 38.41 | 12.30 | 20.48 | 32.66 | 41.66 |
| *insertion-only:* | | | | | | | | | | | | |
| # branch segments | (E) | (F) | (G) | (H) | (E) | (F) | (G) | (H) | (E) | (F) | (G) | (H) |
| 1 | 10.40 | 21.88 | 45.52 | 87.20 | 14.70 | 32.06 | 83.93 | 286.0 | 13.96 | 33.29 | 79.31 | 286.0 |
| 2 | 11.76 | 26.01 | 59.44 | 151.9 | 14.46 | 32.91 | 89.25 | 261.0 | 12.23 | 28.26 | 84.11 | 258.5 |
| 4 | 12.81 | 28.75 | 65.68 | 180.2 | 14.23 | 33.45 | 75.62 | 257.1 | 14.05 | 32.87 | 75.36 | 250.5 |
| 8 | 15.28 | 30.34 | 83.02 | 225.2 | 13.51 | 32.48 | 82.34 | 280.7 | 13.55 | 29.50 | 91.90 | 287.1 |

These results show that iSGv2.0 and DAWG performed appropriately, producing consistent results regardless of the number of internal nodes. EvolveAGene3 also behaved appropriately according to its own indel model. iSGv1.0, ROSE, SIMPROT, and MySSP are all affected by the number of internal nodes, producing artificially high or low rates for deletions or insertions, respectively.

### 3.2.2 Test2: Including both insertions and deletions with various $P_{ins}$ and $P_{del}$

To further examine the effect of indel models, we simulated both insertions and deletions using the Zipfian distribution [15] with five methods: DAWG, ROSE, SIMPROT, iSGv1.0, and iSGv2.0. We simulated three scenarios: (1) $P_{ins} = 0.01$ and $P_{del} = 0.03$, (2) $P_{ins} = 0.02$ and $P_{del} = 0.02$, and (3) $P_{ins} = 0.03$ and $P_{del} = 0.01$, where $P_{ins}$ and $P_{del}$ are the number of insertions and deletions per substitution, respectively. We chose to use the Zipfian distribution since it is an empirically determined length distribution for insertion and deletion events. For MySSP, which implements a length distribution that is normally distributed based on the mean indel length given by the user, we used the expected indel length of 2.082, which is based on the Zipfian distribution with a maximum indel size of 10. EvolveAGene3 was excluded from this test because changing $P_{ins}$ and $P_{del}$ fundamentally alters the indel creation method in EvolveAGene3. In this test, an event counter reporting the numbers of insertions and deletions that occurred during the simulation was added to each method. Since we were unable to obtain the source code for MySSP, we calculated the number of events as follows. Each gap in the root sequence in the true MSA is the effect of an insertion in the descendant sequences, and likewise, each gap in the tip sequence is the result of a deletion in the ancestral sequence. To obtain the number of insertion and

deletion events, we tallied the total number of gaps in the root and tip sequences, respectively, and divided that number by the mean indel size. We measured the quality of indel simulation by comparing the numbers of insertions and deletions generated. We calculated the coefficient of variation ($\sigma^2/\mu$), which is dimensionless and makes results from different simulation methods comparable. If $\sigma^2/\mu \approx 0$, it means that the simulation method behaved similarly between the guide trees (no effect of different number of nodes). A larger $\sigma^2/\mu$ suggests that the simulation method performed differently between the guide trees with different number of nodes.

Figure 3.8 and Table 3.3 show the results of this test. As expected, the numbers of insertions and deletions generated are affected by the insertion and deletion probabilities. For iSGv1.0 and ROSE, when $P_{ins} \neq P_{del}$, the number of internal nodes affects both the numbers of insertions and deletions. SIMPROT, as a result of their multiple-hit correction feature, shows much more drastic effects with the internal-node numbers than iSGv1.0 and ROSE when the insertion rate is larger than the deletion rate ($P_{ins} = 0.03$ and $P_{del} = 0.01$). Such effects are not shown when the deletion rate is larger ($P_{ins} = 0.01$ and $P_{del} = 0.03$). MySSP shows increasing numbers of indels for each test, with the most drastic change occurring when $P_{ins} = 0.03$ and $P_{del} = 0.01$. This behavior can be better understood using the results of Test 1, where in the case of insertions, the sequence length grows as more internal nodes are added, but the sequence length is stable for deletions under the same conditions.

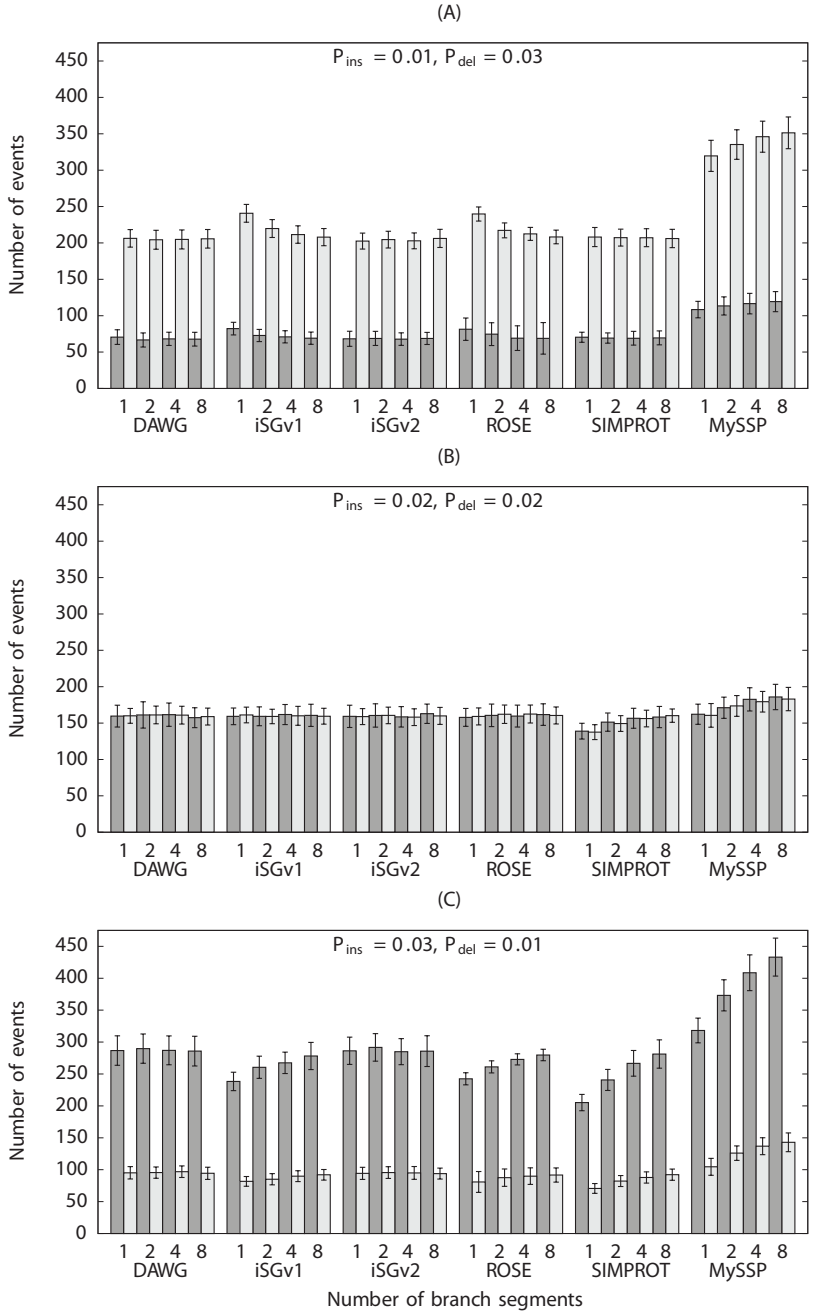



Figure 3.8: Test 2 results with different insertion/deletion probability ratios. For each method, the total numbers of insertions (dark bars) and deletions (light bars) generated are shown for simulation experiments using the guide trees with different numbers of segments (see Figure 3.5. Note that for MySSP we used the average expected value of the Zipfian distribution to obtain the results. For all methods, the average values obtained from 100 simulations are used.

Table 3.3: The effects of internal-node numbers with varying insertion and deletion rates among methods[a].

| | $\sigma^2/\mu$ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MySSP | | SIMPROT | | DAWG | | iSGv1.0 | | iSGv2.0 | | ROSE | |
| | ins | del | ins | del | ins | del | ins | del | ins | del | ins | del |
| (A) $P_{ins} = 0.01$, $P_{del} = 0.03$ | 0.145 | 0.433 | 0.003 | 0.012 | 0.015 | 0.004 | 0.216 | 0.726 | 0.021 | 0.002 | 0.226 | 0.734 |
| (B) $P_{ins} = 0.02$, $P_{del} = 0.02$ | 0.507 | 0.417 | 0.380 | 0.435 | 0.014 | 0.002 | 0.004 | 0.001 | 0.022 | 0.007 | 0.003 | 0.000 |
| (C) $P_{ins} = 0.03$, $P_{del} = 0.01$ | 4.871 | 1.675 | 2.736 | 0.757 | 0.006 | 0.003 | 0.734 | 0.295 | 0.005 | 0.031 | 0.816 | 0.265 |

[a] ins: insertion, del: deletion.

Table 3.3 summarizes the degree of variation ($\sigma^2/\mu$) in the numbers of insertions and deletions among Test 2 experiments. For iSGv1.0 and ROSE, we note that the variation in the number of insertion and deletion events is higher trending towards the dominant event-probability as shown in Figure 3.8, while MySSP shows high variability in all tests, although it is most pronounced when the relative insertion rate is high. SIMPROT also shows high variation when $P_{ins} = 0.03$ and $P_{del} = 0.01$ or $P_{ins} = P_{del}$, although it is comparable with iSGv2.0 and DAWG when $P_{ins} = 0.01$ and $P_{del} = 0.03$. When $P_{ins} = P_{del}$, the indel models of iSGv1.0 and ROSE appear to be affected very little by internal node numbers. Note that when $P_{ins} = P_{del}$, iSGv2.0 and DAWG show slightly larger values than iSGv1.0 and ROSE. This is a consequence of the larger number of steps with indel events and sequence length evaluations performed by these methods. Simulation as a random walk increases the variance in sequence length at each step. DAWG and iSGv2.0 take much larger numbers of steps (600 and 1024, respectively) compared to at most eight steps taken by all other simulators. We confirmed this result by varying the number of steps taken by iSGv2.0 (data not shown).

### 3.2.3 Example application for a protein superfamily simulation

Figure 3.6 shows examples of "true alignment" output obtained from ROSE, iSGv1.0 and iSGv2.0. As shown in Table 3.4, iSGv2.0 correctly conserved 80.98% of the sequence positions, while iSGv1.0 and ROSE, both of which cannot conserve sets of characters, correctly modeled only 61.82% and 61.88% of the positions, respectively. Of the different motifs, iSGv2.0 perfectly conserved all sites of SCR1 (see Figure 3.6). This is because this motif was present in the root alignment and conserved from

the beginning of the simulation run along the lipocalin lineage. For other motifs, all substitutions were accepted until the motif became effective later in the tree, after which only substitutions conforming to the position-specific constraints were accepted.

Table 3.4: Performance comparison among iSGv1.0, ROSE, and iSGv2.0 for the calycin superfamily simulation[a].

| | iSGv1.0 | ROSE | iSGv2.0 | Input alignment |
|---|---|---|---|---|
| Percent sequence identity | 19.78 | 17.72 | 14.62 | 15.65 |
| Percent motif positions conserved | 61.82 | 61.88 | 80.98 | – |
| Number of template violating indels[b] | 13.18 | 19.91 | 0 | – |
| Number of rejected indels[c] | NA | NA | 0.38 | – |

[a] Statistics for iSGv1.0, ROSE, and iSGv2.0 are averages from 100 simulations.
[b] The number of indels that produced subsequences that were larger or smaller than the maximum or minimum values given by the template specified for iSGv2.0.
[c] A rejected indel occurs when a scan of the sequence returns no positions in which an indel can be placed because of subsequence size constraints imposed by the template. NA: not available.
[d] Approximate values inferred from the sequences as given in the true MSA including internal nodes in the guide tree.

We observed multiple side effects due to the restrictions imposed by the quaternary invariable and I $+\gamma$ arrays of iSGv1.0 and ROSE, respectively. As seen in Figure 3.6, conserved motifs in the MSA appeared as "islands" where indels were absent. Additionally, invariable sites such as the GXW region of SCR1 were conserved for the entire column of the alignment, despite the fact that it should only be conserved among kernel lipocalins and the outlier lipocalin family of odorant-binding proteins (OBPs). iSGv1.0 also simulated fewer indels, which is a side effect of the number of alignment positions that contain motif positions, reducing the number of accepting positions for indels. It appears that ROSE uses the absolute number of residues in the sequence to calculate the overall probability of an indel for a branch,

regardless of the number of non-accepting sites for indels. Differences in the indel placements between iSGv1.0 and ROSE versus iSGv2.0 are also evident. As shown in Figure 3.6, iSGv2.0 has a much higher number of indels along the N- and C-terminal regions of the alignment. This is because these regions had only weak constraints on their sizes: The N-terminal was constrained to 10 to 43 residues and the C-terminal 10 to 30 residues. During the simulation process, iSGv2.0 determined the size of the indel, and based on both template and motif constraints searched the sequences to find regions that could accept the indel. Most of the larger indels tended to fall in the least constrained regions. Since neither iSGv1.0 nor ROSE has such constraint capabilities, indels were placed wherever they were not forbidden by the quaternary invariable and I $+\gamma$ arrays. Furthermore, the superfamily fold could not be modeled by either iSGv1.0 or ROSE. They placed an average of 13.18 and 19.91 template-breaking indels, respectively (Table 3.4). iSGv2.0 upheld the template restrictions. On average 0.35 indels per simulation run were rejected using iSGv2.0 because there were limited acceptable positions for indels due to template constraints.

The input MSA (Figure 3.3) had an average pairwise sequence identity of 15.65% (Table 3.4). The 20–35% range of sequence identity or lower is the so-called twilight zone of sequence identity [72], which is often seen among proteins belonging to highly divergent families. iSGv1.0, ROSE, and iSGv2.0 simulated datasets in this range, with the average values over 100 runs of 19.78%, 17.72%, and 14.62%, respectively. The difference in sequence identities between iSGv1.0 and ROSE versus iSGv2.0 is explained by the global conservation of invariable sites by iSGv1.0 and ROSE, even for sequences without the lineage-specific motifs. iSGv1.0 and ROSE both showed a lower percent motif positions conserved (Table 3.4) indicating that some positions were conserved by them even if they did not conform to the residue constraints for different protein families.

## 3.3 Conclusion

Good sequence evolution simulation requires not only realistic event simulation through substitution, insertion, and deletion, but also needs realistic constraint enforcement and heterogeneous evolution among between subsequences and among subtrees. In this study, we showed that while many of current simulation methods introduce insertion and deletion events, only iSGv2.0 and DAWG have robust models. We introduced a formal model of functional constraints on substitution and indel events. We improved the modeling of sequence evolution by fixing indel evolution, incorporating novel functional constraints for motif conservation and subsequence length preservation, and improved heterogeneous sequence evolution and lineage-specific evolution. iSGv2.0 also added modeling of coding and non-coding DNA evolution.

We showed that the majority of indel-simulating programs incorporate indel models that do not account for sequence-length variations during the branch evolution. They introduce bias into the results of the sequence simulation, although such biases are not evident when insertion and deletion frequencies are equal. We also showed that adding subsequence-length constraints and motif constraints allows iSGv2.0 to correctly model superfamily evolution in the twilight zone of sequence similarity.

# Chapter 4

# Gap Profiling

Gaps occurring in a MSA are interpreted as either (1) a non-homologous gain of characters (insertion) in a sequence other than the sequence containing the gap, or (2) the loss of a subsequence (deletion) in the sequence containing the gap. In either case, we infer that an event occurred in a single, discrete position for each sequence; i.e., between two characters in the sequence containing the gap, regardless of the length of the insertions or deletions. In the alignment of a gap region, the ungapped subsequence is aligned with $i$ gap characters. In order to constrain gap sizes to one character, we introduce a binary array that represents the positions between consecutive characters in the sequence. In this array, we represent the existence of gaps between each pair of consecutive characters for each sequence in an MSA in a technique we refer to as "gap profiling." The advantage of this representation is that the lengths of gaps do not affect the representation, as all gaps are represented as a single character state.

After creating the representation of gaps for both true and reconstructed MSAs, the reconstructed MSA gap profiles can be compared to the gap profiles of the true MSA. From this we can calculate both the specificity and the sensitivity of the gap

placements. We then develop *gap profile score* by rewarding sensitive gap placements, while penalizing low specificity.

The gap profile score is a novel representation of indel placement in MSAs, and is a first step in assessing the gap penalty schemes implemented in alignment methods, potentially leading to improved schemes, ultimately leading to better MSA method performance. Along with better MSA performance, gap profiling can improve secondary analyses by providing a starting point for manual alignment – a scoring metric based on the accuracy of placing gaps in the alignment. The value of the quality measures produced by MSA scoring metrics may lead to better alignments, but their value in secondary studies, such as phylogenetic tree reconstruction, is unknown. The lack of knowledge of the phylogenetic history of sequences in the benchmark datasets has blocked the ability to test the values returned by MSA method scoring metrics versus the reliability of phylogenetic topological inferences.

In this chapter, we begin by simulating a new benchmark dataset to assess the gap profile score using indel-Seq-Gen version 2.0 [86] to obtain a variety of datasets that have a known evolutionary history and the true MSA with known indel locations. We compare the gap profile score against commonly used MSA scoring metrics, the sum of pairs (SPS, or $Q$) and total column (TC) scores [93], the shift score [18], and the Modeler score ($f_M$) [75]. Finally, we compare the scores returned by the $Q$ score and the gap profile score against the phylogenetic reconstruction accuracy to determine if there is a correlation between the MSA accuracy (with respect to the scoring method metric) and phylogenetic tree reconstruction.

## 4.1 Methods

Calculating gap profile scores, like other MSA quality scores, requires both a true MSA, $M^{True}$, and a reconstructed MSA, $M^{Recon}$. The calculation of gap profiles for $M^{True}$ is no different than $M^{Recon}$; thus, for the following discussion, we will include the superscripts only when necessary, otherwise referring only to the MSA $M$.

### 4.1.1 Construction of Benchmark Datasets

In order to test MSA scoring metrics, we require a benchmark dataset that has no uncertainty in the placement of characters, or in the case of the gap profile score, gaps. Current benchmarks using data derived from extant taxa are commonly judged based on well-conserved (non-gapped) regions, making them unsuitable for this purpose. To rectify this shortcoming, we create a nucleotide sequence benchmark dataset using the sequence simulation method indel-Seq-Gen version 2.0 [86]. We simulated non-coding nucleotides with low substitution rates with high insertion and deletion probabilities, using the generalized time reversible substitution matrix with among-site rate heterogeneity modeled by the Gamma density function ($\mathrm{GTR} + \Gamma$). We set the nucleotide frequencies equal to those found in the Nematode Tree of Life project [101], as used in Liu *et al.* [53]. In the simulation framework, we vary five parameters: The insertion and deletion rates, the tree topology, tree length, indel size, and number of taxa, as listed in Table 4.1. We use the balanced and pectinate tree topologies, in order to cover the two extremes of evolutionary relatedness between taxa: i.e., any particular random tree falls somewhere in between these two tree types. The balanced tree gives a view of gene-family-like evolution, where each branching point is a new subfamily. In such a tree, nearly half of the insertions and deletions will be shared by more than one descendant sequence. In the ladder-like pectinate tree, fewer indels will be

shared. Table 4.2 shows the comparison of balanced and pectinate trees used in this study.

Table 4.1: List of parameters and reconstruction methods used for dataset creation.

| Parameter | Values taken |
|---|---|
| Tree type | balanced or pectinate (both ultrametric) |
| Tree length[a] | 0.25, 0.5, 0.75, 1.0 |
| Number of taxa | 8, 16, 32, 64 |
| $P_{ins}$:$P_{del}$[b] | 0.0:0.2, 0.05:0.15, 0.1:0.1, 0.15:0.05, 0.2:0.0 |
| Indel size | 1, 2, 5, 10 |
| Methods[c] | ClustalW2.0 [50], MAFFT [48],[d] Muscle3.7 [24], ProbCons [20] |

[a] Tree length is the root to tip length of the tree, where the length is in substitutions per site.

[b] $P_{ins}$ and $P_{del}$ are the probabilities of insertion and deletion, respectively, per substitution.

[c] We use the default settings of each method, unless otherwise noted.

[d] MAFFT is run using the `--linsi` option.

Table 4.2: Comparison of balanced and pectinate trees, as shown in Figure 1.6.

| | Number of taxa in tree | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8[b] | | 16[b] | | 32[b] | | 64[b] | |
| | bal | pec | bal | pec | bal | pec | bal | pec |
| *Tree length*[a] | | | | | | | | |
| 0.25 | 1.167 | 1.250 | 1.875 | 2.250 | 3.100 | 4.250 | 5.250 | 8.250 |
| 0.50 | 2.330 | 2.500 | 3.750 | 4.500 | 6.200 | 8.500 | 10.50 | 16.50 |
| 0.75 | 3.500 | 3.750 | 5.625 | 6.750 | 9.300 | 12.75 | 15.75 | 24.75 |
| 1.00 | 4.670 | 5.000 | 7.500 | 9.000 | 12.40 | 17.00 | 21.00 | 33.00 |
| Percent length difference | **0.933**[c] | | **0.833**[c] | | **0.729**[c] | | **0.636**[c] | |
| *Ratio of evolution performed in internal tree branches*[d] | 0.429 | 0.171 | 0.467 | 0.104 | 0.484 | 0.057 | 0.492 | 0.030 |

[a] The tree length is defined as the sum of the branch lengths for root-to-tip paths.

[b] Number of taxa in the balanced (bal) and pectinate (pec) trees.

[c] The ratio of length difference is calculated by dividing the summed length of all branches in the balanced tree by the summed length of all branches in the pectinate tree.

[d] This measure gives the ratio of the simulation where indels will be shared by more than one taxon (homologous indels created), calculated as the sum of branches with more than one descendant divided by the total length of all branches in the tree.

### 4.1.2 Gap Profile Score

Calculating the gap profile score is done in three steps. First, the gap profiles need to be calculated for $M^{True}$ and $M^{Recon}$. After all gap profiles are calculated, we measure the sensitivity and specificity of the $M^{Recon}$ versus $M^{True}$. Finally, we the calculate gap profile scores.

#### 4.1.2.1 Profiling Gaps

The gap profiling method is designed to work on pairwise alignments of sequences. Given the MSA $M$ with $N$ sequences $(S_1...S_N)$, we profile gaps on all pairs of sequences $(S_i, S_j)$, where $i, j = 1...N$. In this section, we will profile the pairwise alignment between two general sequences, sequence $S_i$ and sequence $S_j$, where $i \neq j$.

The gap profile for sequence $S_i$ when aligned to sequence $S_j$, $GP^M_{S_i,S_j} = b_0^{S_i,S_j} \ldots b_{|S_i|}^{S_i,S_j}$ is a binary array defined as

$$b_x^{S_i,S_j} = \begin{cases} 1 & \text{if a `} - \text{' appears between } S_i(x-1) \text{ and } S_i(x) \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

for each position $x \in S_i$, where $S_i(x)$ denotes the $x$th character in sequence $S_i$. An example of the calculation of gap profiles $GP^M_{S_i,S_j}$ and $GP^M_{S_j,S_i}$ is shown in Figure 4.1.

We repeat the process for all pairs of sequences, obtaining the set of $N \times N$ gap profiles for the MSA $M$, as shown in Figure 4.2.

#### 4.1.2.2 Sensitivity and Specificity

After calculating the gap profiles, we measure the gap placement accuracy of each $M^{Recon}$ versus $M^{True}$ for each pair of sequences by comparing $GP^{Recon}_{S_i,S_j}$ against $GP^{True}_{S_i,S_j}$. To accomplish this, we define two measures: *gap placement sensitivity* and *gap place-*

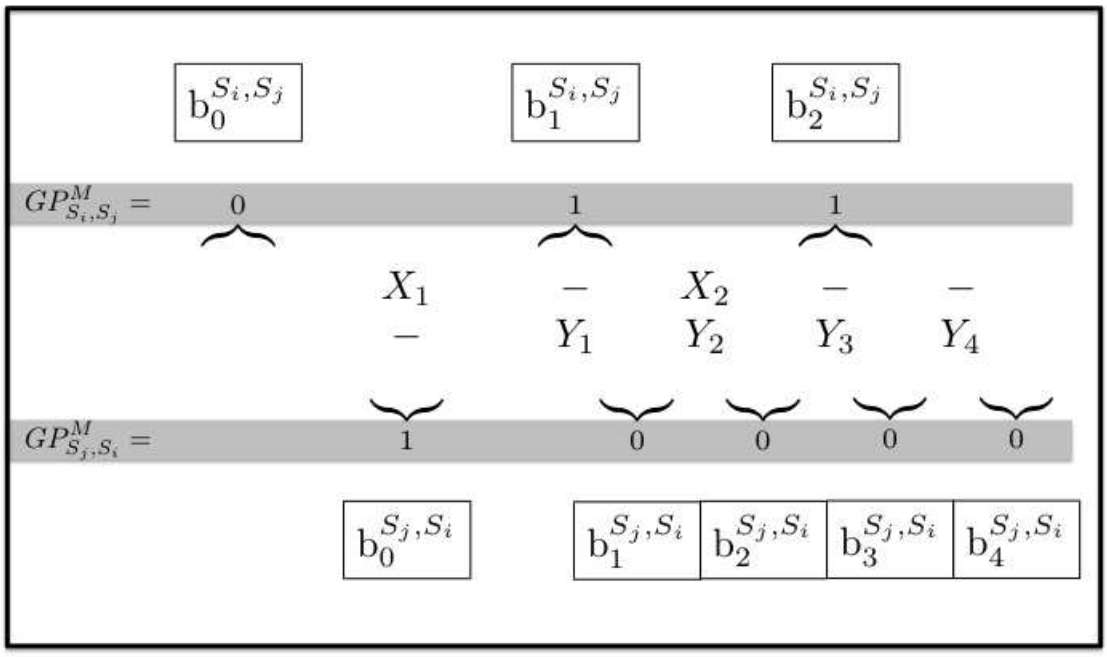



Figure 4.1: Counting the two gap profiles for the pairwise alignment of sequences $S_i$ and $S_j$ from the MSA $M$.

$$M = \begin{bmatrix} GP^M_{S_1,S_1} & GP^M_{S_1,S_2} & \cdots & GP^M_{S_1,S_N} \\ GP^M_{S_2,S_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ GP^M_{S_N,S_1} & \cdots & \cdots & GP^M_{S_N,S_N} \end{bmatrix}$$

Figure 4.2: The $N \times N$ gap profiles that explain gapping pattern for the MSA $M$.

*ment specificity* as follows. Sensitivity measures proximity of gapping pattern $M^{Recon}$ against $M^{True}$, while the specificity measures the over- or underabundance of gaps in $M^{Recon}$ against $M^{True}$.

To calculate the sensitivity, we examine each $b_x^{S_i,S_j} \in GP_{S_i,S_j}^{True}$ where $b_x^{S_i,S_j} = 1$ and $x = 0 \ldots |S_i|$. We search for the minimum value of $d$ such that $\hat{b}_{x\pm d}^{S_i,S_j} = 1$, $d = 0, 1, \ldots, w, > w$, where $\hat{b}_{x\pm d}^{S_i,S_j} \in GP_{S_i,S_j}^{Recon}$ and $w$ is the maximum search distance($w = 2$ in Figure 4.3C). Once the closest gap is located in $GP_{S_i,S_j}^{Recon}$, we increment the corresponding bin $d$ in the sensitivity array. Likewise, for the specificity array, we examine each $\hat{b}_x^{S_i,S_j} \in GP_{S_i,S_j}^{Recon}$ such that $\hat{b}_x^{S_i,S_j} = 1$, search for the minimum value of $d$ in $b_{x\pm d}^{S_i,S_j} \in GP_{S_i,S_j}^{True}$ such that $b_{x\pm d}^{S_i,S_j} = 1$, and increment the $d$th bin in the specificity array. Figure 4.3 depicts the process of calculating the sensitivity and specificity from the true and reconstructed gap profiles of sequence $S_i$ when aligned with $S_j$. This process is repeated for all gap profile pairs in $M$. All counts for gap profile comparisons are summed in the sensitivity and specificity arrays.

#### 4.1.2.3 Calculating Gap Profile Scores

We measure the ability of MSA reconstruction methods to correctly place indels by plotting the sensitivity versus the specificity similar to ROC described in 1.4.2: Using decision lines, placed before $d = 0$ in the sensitivity histogram ($DL_0^{sn}$) and after $d = w$ for the specificity histogram ($DL_w^{sp}$), we plot the proportion of all counts before $DL_0^{sn}$ and after $DL_w^{sp}$, i.e., the first pair $(DL_w^{sp}, DL_0^{sn})$ is $(0, 0)$. We proceed by sliding from $DL_0^{sn} \rightarrow DL_1^{sn}$ and $DL_w^{sp} \rightarrow DL_{w-1}^{sp}$, where '$\rightarrow$' indicates the shift in position of the decision line. After this shift, we plot the proportion of counts above $DL_1^{sn}$ and below $DL_{w-1}^{sp}$, repeating this process until we reach $DL_w^{sn}$ and $DL_0^{sp}$ (last plotted pair, $(1, 1)$), yielding a ROC-like plot of the sensitivity and specificity, as shown for $w = 5, 10, 25, 50, 100, 500$, and 1000 in the left column of Figure 4.4. We calculate the

(A)

| Gap Profiles, $S_i$ vs. $S_j$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $x =$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $GP^{True}_{S_i,S_j}$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $GP^{Recon}_{S_i,S_j}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

(B)

| *Sensitivity calculation* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $x =$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Gap 1, $x = 4$ | | | | 1 | 0 | *1* | | | |
| Gap 2, $x = 7$ | | | | | | | | *0* | |

| *Specificity calculation* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $x =$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Gap 1, $x = 2$ | 1 | 0 | 1 | 2 | | | | | |
| Gap 2, $x = 5$ | | | | | *1* | 0 | 1 | | |
| Gap 3, $x = 7$ | | | | | | | | *0* | |

(C)

| $d =$ | 0 | 1 | 2 | $> 2$ |
|---|---|---|---|---|
| Sensitivity | 1 | 1 | 0 | 0 |
| Specificity | 1 | 1 | 0 | 1 |

Figure 4.3: Calculating specificity and sensitivity. (A) The gap profiles for $M^{True}$ $M^{Recon}$ for sequence $S_i$ when compared to $S_j$ ($GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$). (B) Counting the distance of gaps from $GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$ for sensitivity and specificity calculation. Positions of the search space correspond with (A). A distance $d = 0$ starts where the gap is found. If the corresponding position in the comparison gap profile is not one, $d$ is incremented, and the gap profile values at positions $x \pm d$ are checked, and so on until a corresponding position either contains a value of one or $d = w$. Numbers in italics denote that a corresponding gap has been found, ending the search. (C) The sensitivity and specificity arrays after finishing the comparison between $GP^{True}_{S_i,S_j}$ and $GP^{Recon}_{S_i,S_j}$.

area under the curve to obtain the gap profile score. To make the gap profile score more sensitive/specific, we increase/decrease the size of $w$, respectively, as shown in Figure 4.4. We refer to the gap profile score using a maximum gap distance of $w$ as $\text{GP}_w$ for the remainder of this study.

### 4.1.3 MSA Scoring Technique Comparison

All MSA scoring metrics compare reconstructed MSAs against benchmark datasets. As described in Chapter 1, currently available MSA scoring metrics fall into three categories based on the metric criteria: (1) pairwise character matching, (2) columnar matching, and (3) structural matching. Since we use simulated datasets that do not have structural information, we do not use MSA scoring metrics that require structural information. We compare performances between the $\text{GP}_w$ score and four commonly used MSA scoring metrics: the sum of pairs (SPS; or $Q$) score, total column score (TC), the shift-score [18], and the Modelers score ($f_M$) [75].

Pairwise character matching ($Q$, shift, and $f_M$ scores) and pairwise indel matching (the $\text{GP}_w$ score) examine each pair of sequences and output the statistics based on how well the pairs are matched. The shift score is a pairwise proximity-based measure that assigns positive scores to sequences that have a small shift in homologous pairs, but also allows for negative scores to be assigned for placing homologous pairs very distantly in the alignment. The TC is the columnar matching score. The TC score is a strict measure that assigns a correct score to a match only if all nucleotides in the column are correctly matched. Figure 4.5 shows a matrix representation of the different MSA-scoring metric characterizations. Note that the majority of scores fall in the pairwise character-based category, while there has yet to be an columnar indel-based method.

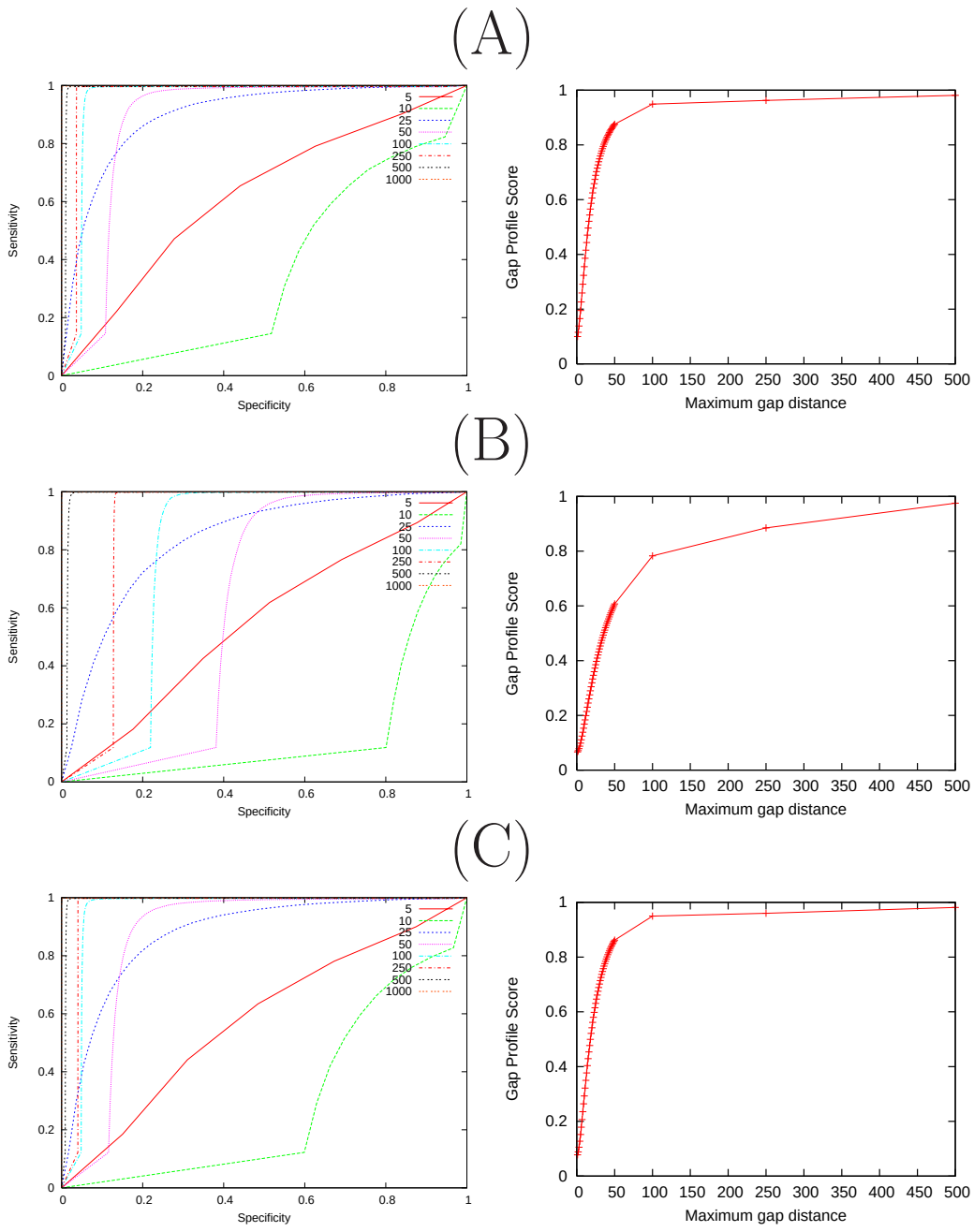



Figure 4.4: *Continued on next page.*

(D)

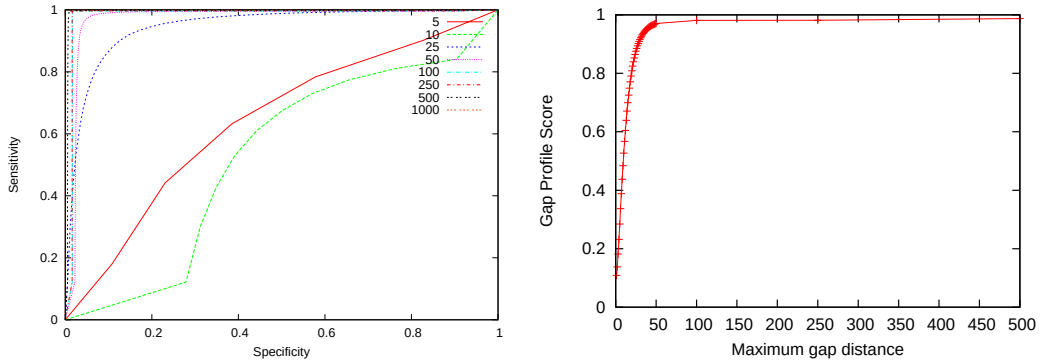



Figure 4.4: Comparison of the gap profile scores with respect to the maximum gap distance ($w$) for (A) ClustalW2.0, (B) MAFFT-linsi, (C) Muscle3.7, and (D) ProbCons. *Left*: Sensitivity versus specificity curves for different window sizes. Note that maximum gap distances of 500 and 1000 may not be visible, as they return near perfect scores, and as such are superimposed with the axes. *Right*: gap profile score for maximum gap distances 1...50, 100, 250, and 500. The represented benchmark condition of this example is a balanced tree of 16 taxa and a length of 0.75 substitutions per site, $P_{ins} = P_{del} = 0.1$, with indel lengths equal to 5.

| | | Counting Method | |
|---|---|---|---|
| | | *Pairwise* | *Column* |
| Data Type | *Characters* | $Q$<br>$f_M$<br>shift | Total Column |
| | *Indels* | gap profile score | |

Figure 4.5: Categories of MSA-scoring metrics.

#### 4.1.3.1 Scoring Method Implementation

We use two implementations of the scoring methods: `baliscore` [97], which implements the SPS and the TC score (called the "Column score" in the implemented version) and `qscore` [23], which implements the $Q$ score (same as SPS), the TC score, the shift score, and the Modelers score ($f_M$).

Both packages contain implementations of the SPS and TC score. These implementations do not perform identically, which will negatively affect our analyses. The `baliscore` implementation of the SPS metric will, in instances, report a value of less than 1.0 (the perfect score) for alignments that are perfect [46]. The `qscore` package fixes this issue [22]. The `qscore` implementation of the TC score, however, counts a column as correct if and only if all the characters in the reference alignment are aligned in the test alignment, while the `baliscore` TC score implementation does not. In `baliscore`, a column that is gapped in the reference alignment but not in the test alignment is counted as correct [22]. Since these packages have been used to score MSA methods in their publications, we first examine how these inconsistencies affect the alignments.

#### 4.1.3.2 Scoring Metric Comparison

New scoring metrics add utility only if they measure different parameters than the existing metrics. For example, if all residues are aligned correctly, then all gaps must also be correctly aligned. In such cases, the SPS is a sufficient alignment goodness measure, and the $\mathrm{GP}_w$ score would only confirm what is already known. In order to evaluate how different scoring metrics quantify the accuracy of alignments differently, we calculate the scores over all reconstructed MSAs using different scoring metrics and examine the correlation.

### 4.1.4 Multiple Sequence Alignment Programs

We compare the values returned by the MSA scoring metrics using four MSA reconstruction methods: ClustalW2.0 [50], MAFFT L-INS-i [47], Muscle3.7 [24], and ProbCons [20]. As shown in Figure 1.2, the progressive alignment schemes employed by each of these methods produce significantly different gapping patterns. While we suspect that these gapping patterns produce significant differences to the $GP_w$ metric, we also examine how the different alignments affect the scores returned by character-based scoring metrics.

#### 4.1.4.1 Statistics Calculation

We calculate the statistics listed in Table 4.3 for each alignment dataset (20 replicates per dataset). The Hamming distances are calculated by counting all pairwise, non-gap matches in each column and dividing it by the number of comparisons made. We also gather the length of contiguous indels and the length of contiguous characters. The length of contiguous characters is defined as the length of a non-gapped stretch of characters in the MSA. The length of contiguous indels is defined likewise for stretches of indels. The statistics for each dataset is calculated over the 20 replicates. Using these measures, we gain a general understanding of MSA gapping patterns without having to examine each MSA individually.

Table 4.3: Statistics gathered from the alignments generated by different methods for each dataset.

| Value | Statistics | | | | |
|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | Median | Minimum | Maximum |
| Number of actual insertions[a] | + | + | | | |
| Number of actual deletions[a] | + | + | | | |
| Total number of insertion characters | + | + | | | |
| Total number deletion characters | + | + | | | |
| Alignment length | + | + | | + | + |
| Percent alignment gaps[g] | + | + | | + | + |
| Average normalized hamming distance | + | + | | + | + |
| Length of contiguous indels | + | + | + | + | + |
| Length of contiguous characters | + | + | + | + | + |
| Tree reconstruction accuracy[b] | + | | | | |
| $GP_5$ | + | + | | | |
| $GP_{10}$ | + | + | | | |
| $GP_{25}$ | + | + | | | |
| $GP_{50}$ | + | + | | | |
| $GP_{100}$ | + | + | | | |
| $GP_{250}$ | + | + | | | |
| $GP_{500}$ | + | + | | | |
| BAliBASE total column[c] | + | + | | + | + |
| $Q$[d] | + | + | | + | + |
| Modeler[d,e] | + | + | | + | + |
| Shift[d,f] | + | + | | + | + |

[a] Actual indels are the actual events, taken from the trace files of iSGv2.0.

[b] Accuracy of the maximum likelihood reconstructed trees using RAxML [83] against the guide tree for the alignment.

[c] Computed using the `baliscore` package from BAliBASE 3.0 [92].

[d] Computed using the `qscore` package [24].

[e] The Modeler score, $f_M$, from [75].

[f] From [18].

[g] Percent alignment gaps is the proportion of characters in the MSA that are gap characters.

### 4.1.5 MSA Scoring Methods versus Phylogenetic Accuracies

We examine the correlation of the accuracies of MSA reconstruction methods (based on scoring metric values) to the secondary analyses accuracies of phylogenetic trees built on the MSAs. Phylogenetic trees are reconstructed using the maximum likelihood method implemented in RAxML [83]. We set the substitution paramaters in RAxML to the generalised time-reversible (GTR) model [91] using among site rates of change follow that Gamma density function (`GTRGAMMA` for the 8, 16, and 32 taxon datasets, while using `GTRMIX` for the 64 taxon dataset; recommended settings in the RAxML manual). We compare the topologies of the reconstructed tree to the guide trees used to create the benchmark datasets. We count the topological accuracy as the number of correctly reconstructed tree bipartitions [70], as illustrated in Figure 4.6. All branches that lead to a taxon (terminal branches) are always correct (the bipartition of one taxon against all other taxa); therefore such bipartitions are excluded.

## 4.2 Results and Discussion

The $GP_w$ scoring metric is a unique scoring metric in that it can be very stringent (with a small $w$) or it can be very lenient (with a very large $w$), as seen in Figure 4.4. Thus, the choice of $w$ leads to a large difference in the outcomes of the scoring method. For this reason, we will restrict our results to the $GP_5$, $GP_{10}$, and $GP_{25}$ scores.

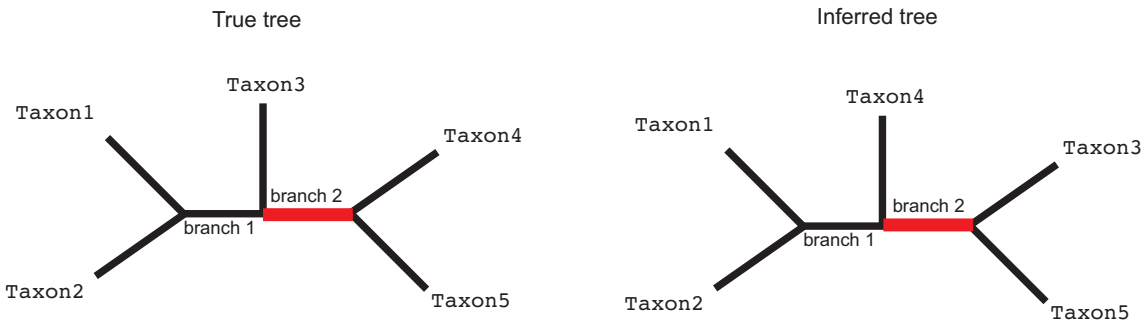



Figure 4.6: Counting topological accuracy. The true tree is the guide tree used for simulating the dataset. The inferred tree is the maximum likelihood tree inferred from reconstructed multiple sequence alignments. Two internal branches, "branch 1" and "branch 2," are non-trivial to infer: branch 1 creates the bipartition {{`Taxon1`, `Taxon2`}, {`Taxon3`, `Taxon4`, `Taxon5`}} in both the true and inferred tree, while branch 2 creates an incorrect bipartition of {{`Taxon1`, `Taxon2`, `Taxon4`}, {`Taxon3`, `Taxon5`}} in the inferred tree compared to the true bipartition {{`Taxon1`, `Taxon2`, `Taxon3`}, {`Taxon4`, `Taxon5`}}. Thus, the topological accuracy for this example is 0.5 (one correct branch divided by two internal branches).

### 4.2.1 Scoring Method Comparison

We begin by examining the differences in measuring the SPS and TC score metrics between the `qscore` and `baliscore` packages, in particular paying attention to the parameters in the benchmark dataset that affect the scoring method. We follow this by examining the correlations between scoring method metrics, with a focus on whether the $GP_w$ score we introduced provides new information.

#### 4.2.1.1 Scoring Method Inconsistencies

**Sum of pairs score (SPS):** The SPS implementations in `baliscore` has a tendency to assign higher scores to datasets than the `qscore` implementation. The primary cause for this difference is the indel size of the dataset. Figure 4.7 illustrates the two sum of pairs metrics when the indel size used to create the benchmark is varied. The datasets far from the diagonal are those with indel sizes of 10 (purple

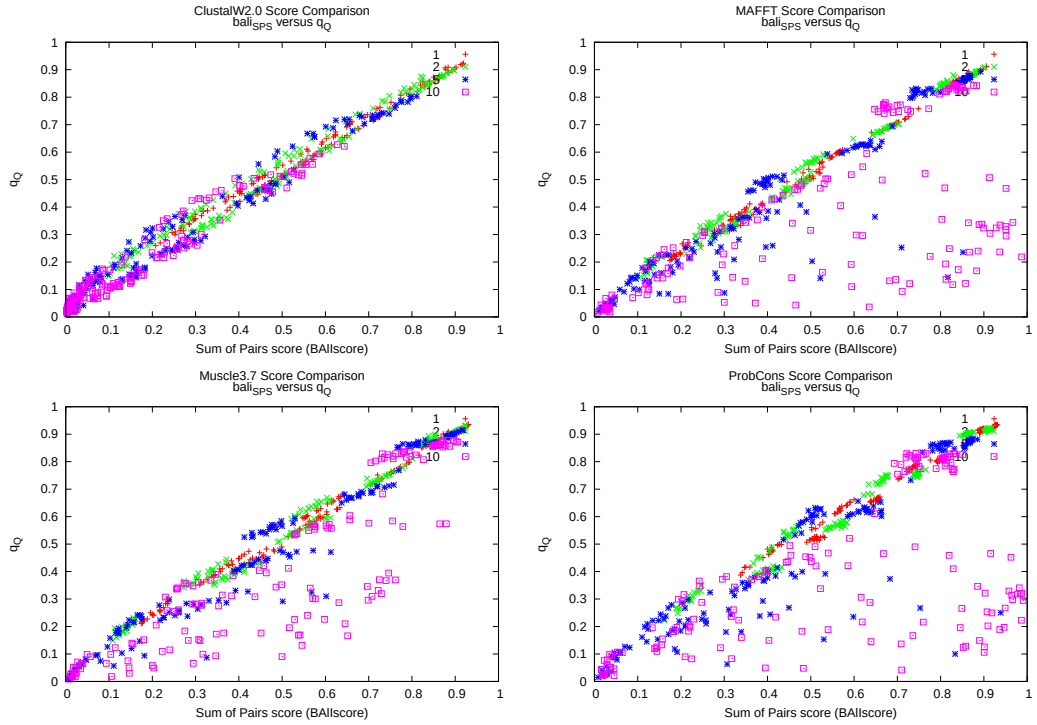



Figure 4.7: Comparison of the SPS scores from the `baliscore` ($x$-axis) and `qscore` ($y$-axis) packages. Alignment dataset points are plotted based on the four different indel sizes used to create the benchmark alignments: indel sizes 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡').

squares) and 5 (blue '*'), where the SPS score is higher than the $Q$ score for Muscle3.7, MAFFT, and ProbCons. ClustalW2.0 does not display this behavior. Tree lengths also affect the distance of each point off of the diagonal (smaller tree lengths lead to fewer indels and are closer to the diagonal), as do the $P_{ins}$:$P_{del}$ ratios (enrichment in deletions over insertions are more distant from the diagonal; results in Appendix B).

These observations lead us to believe that the `baliscore` package assigns a score to character-to-gap matches to judge how well a MSA method performs. As such, longer trees and larger indels tend to have a much larger effect on SPS calculation, as they will increase the proportion of gaps in the resulting alignment. Datasets enriched in deletions also result in higher SPS scores, since deletions remove the potential

number of positions of character-character matches, giving higher ratios of support to the noise introduced by character-to-gap scores. This effect is observable when we compare the plots using different $P_{ins}$:$P_{del}$ ratios. Higher deletion rates correlate with higher scores for `baliscore` SPS score, while higher insertion rates tend to assign the same value as the `qscore` $Q$ score.

Figure 4.8 shows the results plotted for the contiguous indel lengths versus the `baliscore` and `qscore` metrics for each of the MSA reconstruction methods. In these figures, "clumps" of points are created based on the indel size and the tree length properties of the benchmark dataset for Muscle3.7, MAFFT, and ProbCons; e.g., smaller tree lengths (e.g., easily alignable datasets) have a clump of data points that have high $Q$ scores and cluster around the true length of the indel size (i.e., for the indel size of 10, smaller tree lengths have contiguous indel length clusters that fall near size 10). ClustalW2.0 does not create clumps of data points for benchmark datasets with similar properties. Rather, ClustalW2.0 spreads the indel sizes continuously with respect to tree size. The clumps of datapoints, from high $Q$ score to low $Q$ score, are tree lengths of 0.25, 0.5, 0.75, and 1, respectively. These results are shown in Appendix B.

**Total Column Score (TC):** The TC score between the `baliscore` and `qscore` methods is more difficult to interpret, as two cases arise: `baliscore` $\ll$ `qscore` and `baliscore` $\gg$ `qscore`. We will address each of these separately. In Figures 4.9 and 4.10, the TC scores are compared with varied parameters.

`baliscore` $\ll$ `qscore`**:** This type of discrepency in the TC score is mostly dependent on interplay between the tree type and the $P_{ins}$:$P_{del}$ ratios. The `qscore` implementation tends to give artificially high scores to datasets built using balanced

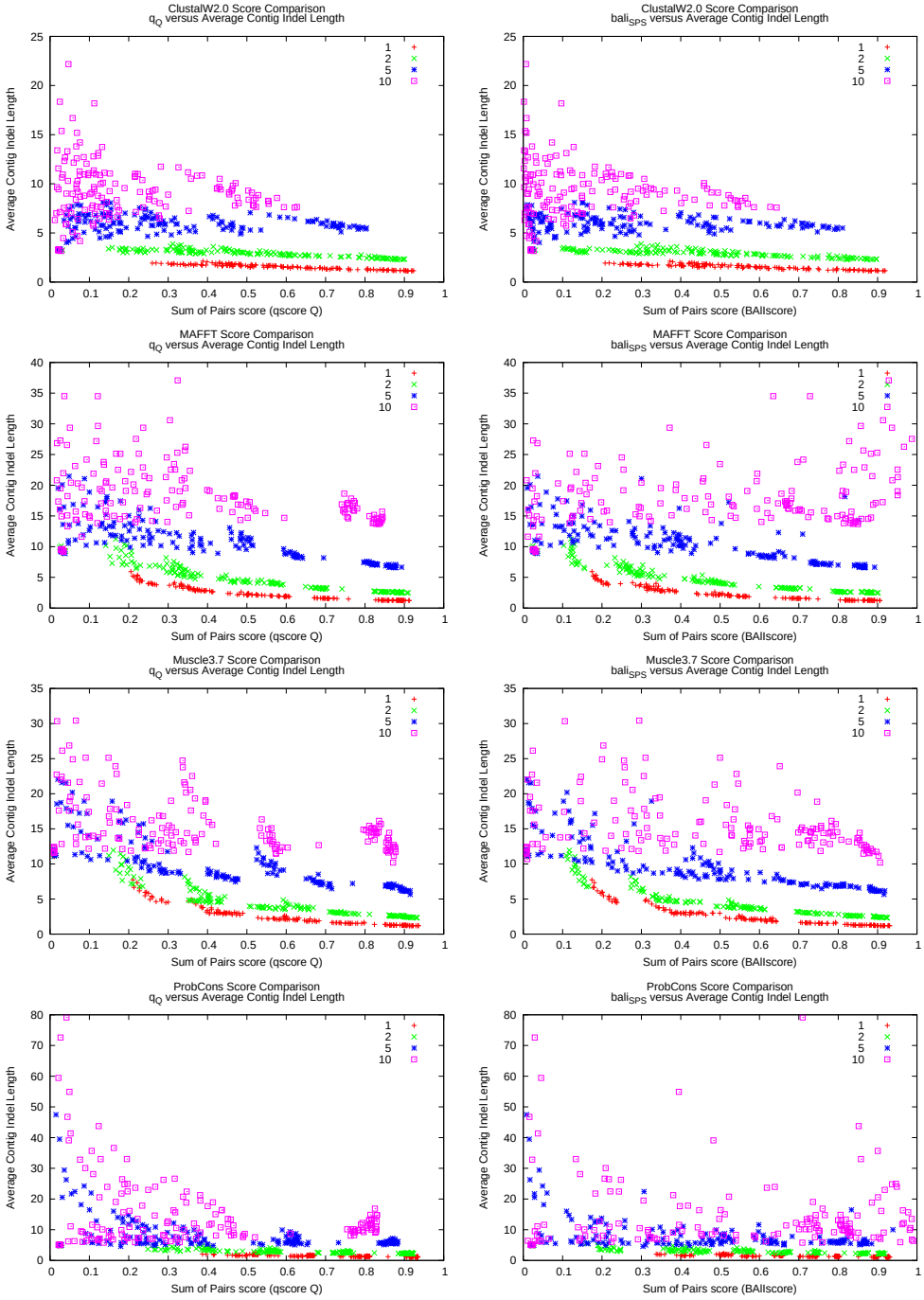



Figure 4.8: Relationships between the length of contiguous indels and the SPS values in scoring alignments by different MSA reconstruction methods. *Left:* The $Q$ score metric. *Right:* The `baliscore` SPS metric. Alignment dataset points are plotted based on the four different indel sizes used to create the benchmark alignments: indel sizes 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡').

trees with high insertion rates, as such trees will have many columns that share gaps.

`baliscore` $\gg$ `qscore`: For these datasets, the tree type does not seem to matter. Rather, it is a complex interplay between three values: tree length (except in Muscle3.7), indel size, and $P_{ins}$:$P_{del}$ ratio. The MSAs that display higher `baliscore` values than `qscore` values are made up of high ratios of deletions. This provides a much easier task for the `baliscore` TC score, in that correct column will contain many fewer residues that need to be aligned.

The `baliscore` and `qscore` package show significant differences in their implementation of SPS and TC scores. However, these differences are small in datasets with few indels. Each measure is primarily character-based; as such, we exclude methods that incorporate gap information, i.e., `baliscore` implementation of the SPS (scores character-to-gap matches) and the `qscore` implementation of the TC score (counts columns that share gaps). The measures we use, `qscore` $Q$ and `baliscore` TC, will be referred to as $\mathrm{q}_Q$ and $\mathrm{bali}_{TC}$ for the remainder of this study.

#### 4.2.1.2 Independence of Measured Values

Each different scoring metric should show differences in the returned value, depending on the type of alignments, e.g., minimal information can be gained about alignment goodness if two scoring metrics measure very similar attributes of a given dataset. On the other hand, if two scoring metrics measure complementary attributes, it may be possible to combine the scoring metrics to gain a better understanding of the alignment goodness. It is also useful to know if a particular scoring metric is sensitive to changes in an attribute that will affect its performance regardless of other attributes.

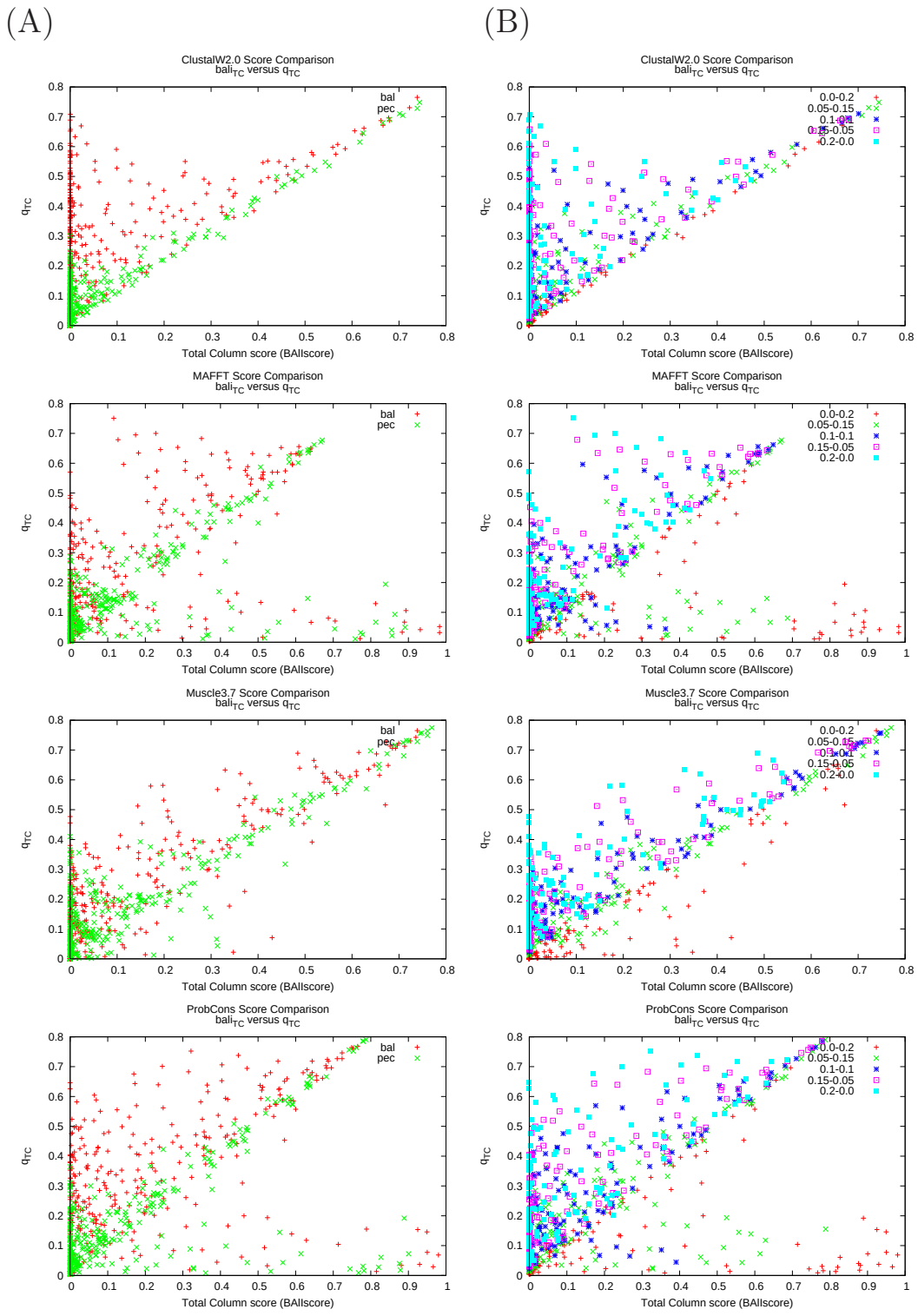



Figure 4.9: Comparison of the Total Column scores from the `baliscore` and `qscore` packages, with respect to (A) tree type: balanced (red '+') and pectinate (green 'x'), and (B) $P_{ins}$:$P_{del}$ ratio: 0.0-0.2 (red '+'), 0.05-0.15 (green 'x'), 0.1-0.1 (blue '*'), 0.15-0.05 (purple '⊡'), 0.2-0.0 (cyan '■').

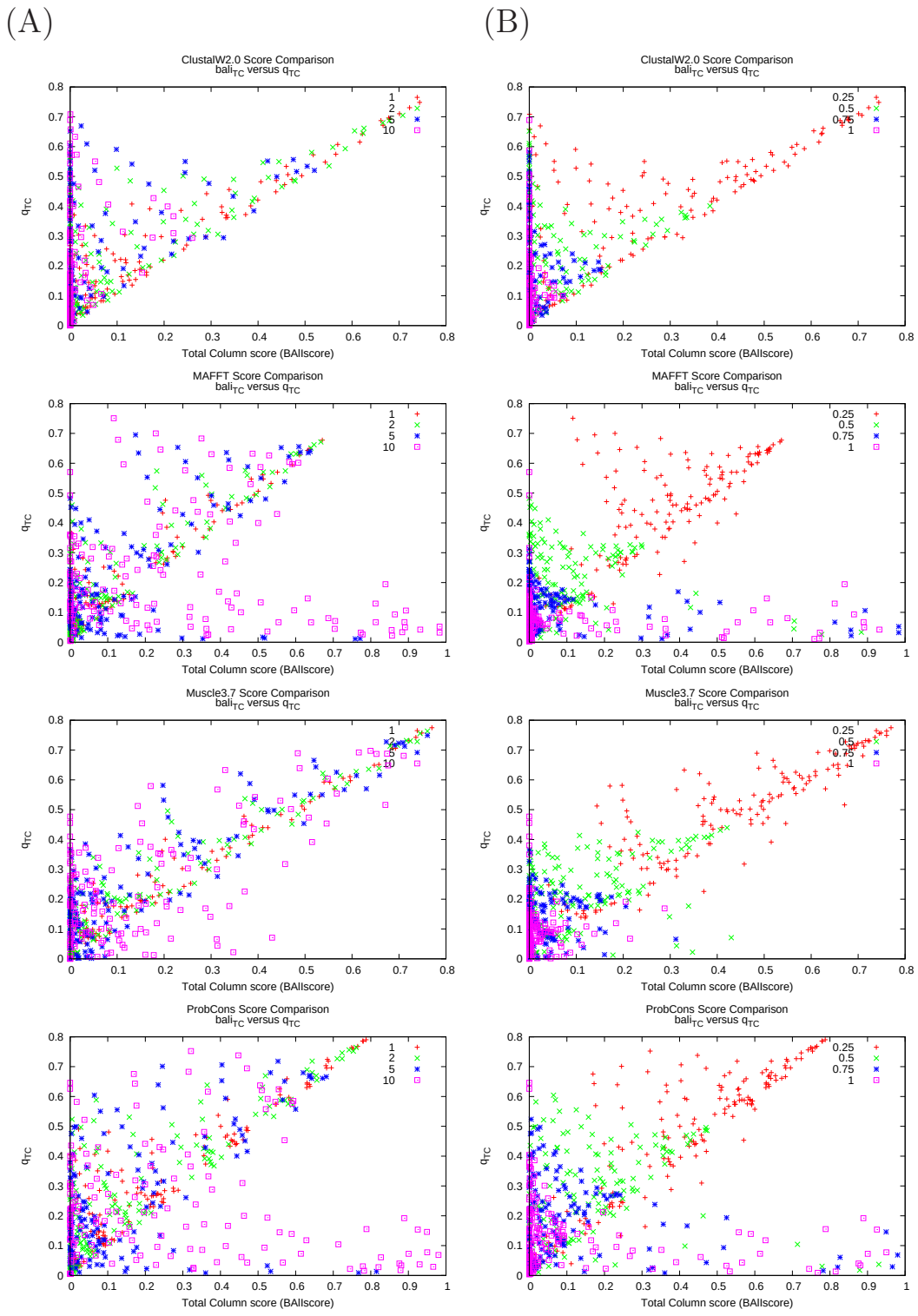
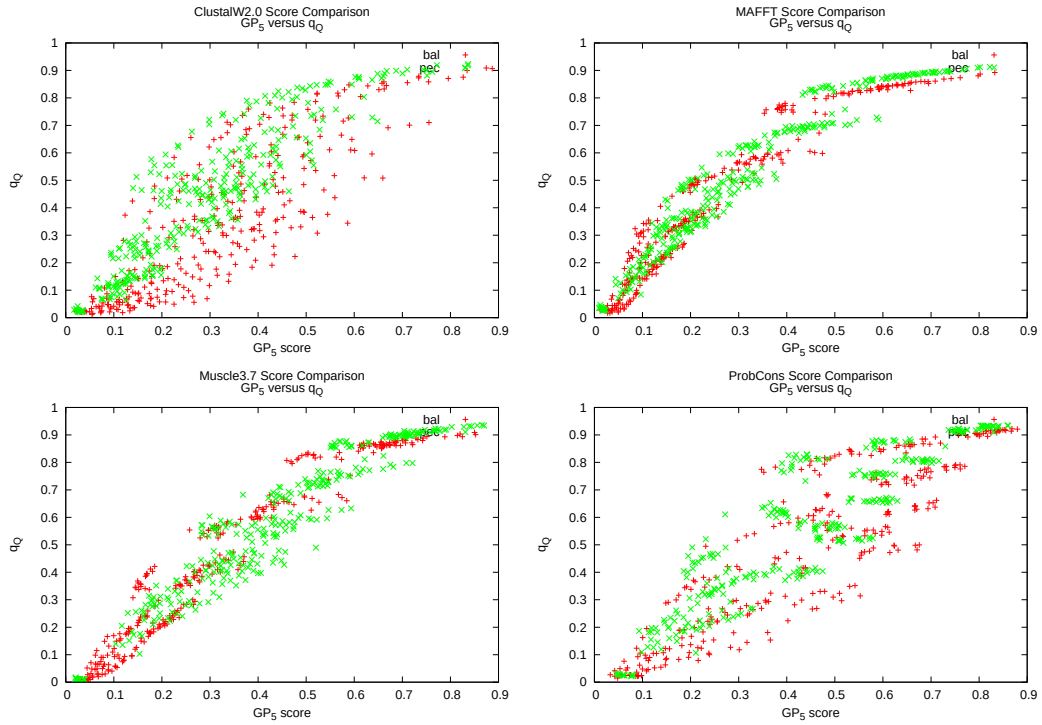



Figure 4.10: Comparison of the Total Column scores from the `baliscore` and `qscore` packages, with respect to (A) indel size: 1 (red '+'), 2 (green 'x'), 5 (blue '*'), and 10 (purple '⊡') and (B) tree length: 0.25 (red '+'), 0.5 (green 'x'), 0.75 (blue '*'), and 1 (purple '⊡').

**Correlated Measures:** If two scoring metrics provide the same measurements for conditions, we expect to see the returned metric values on the datasets fall on the line from $(0, 0)$ to $(1, 1)$ in a plot with each returned metric score on each axis. Such is the case for the $\mathrm{q}_Q$ score and $f_M$ (Figure 4.11D). There is also a general, although weaker, correlation between all metrics except the $\mathrm{bali}_{TC}$ score (Figure 4.11A–E). For these metrics, it appears that the $\mathrm{GP}_w$ score shows the lowest correlation with the other methods, suggesting that the $\mathrm{GP}_w$ score is affected by different attributes than the $\mathrm{q}_Q$ score, shift score, and $f_M$.

(A)




(B)

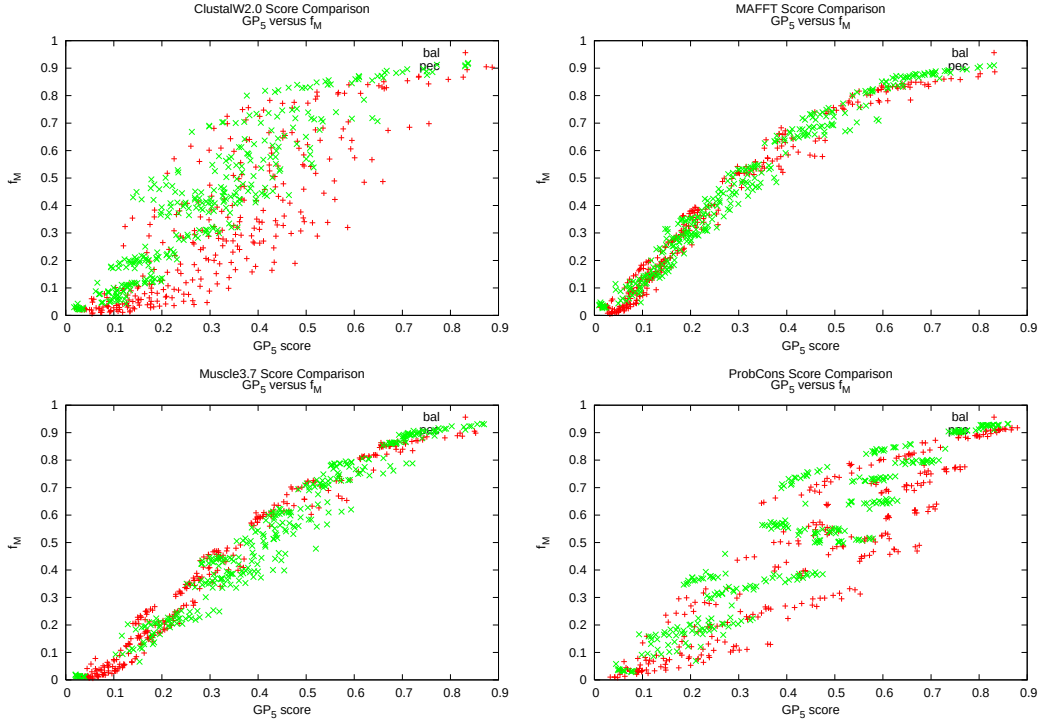
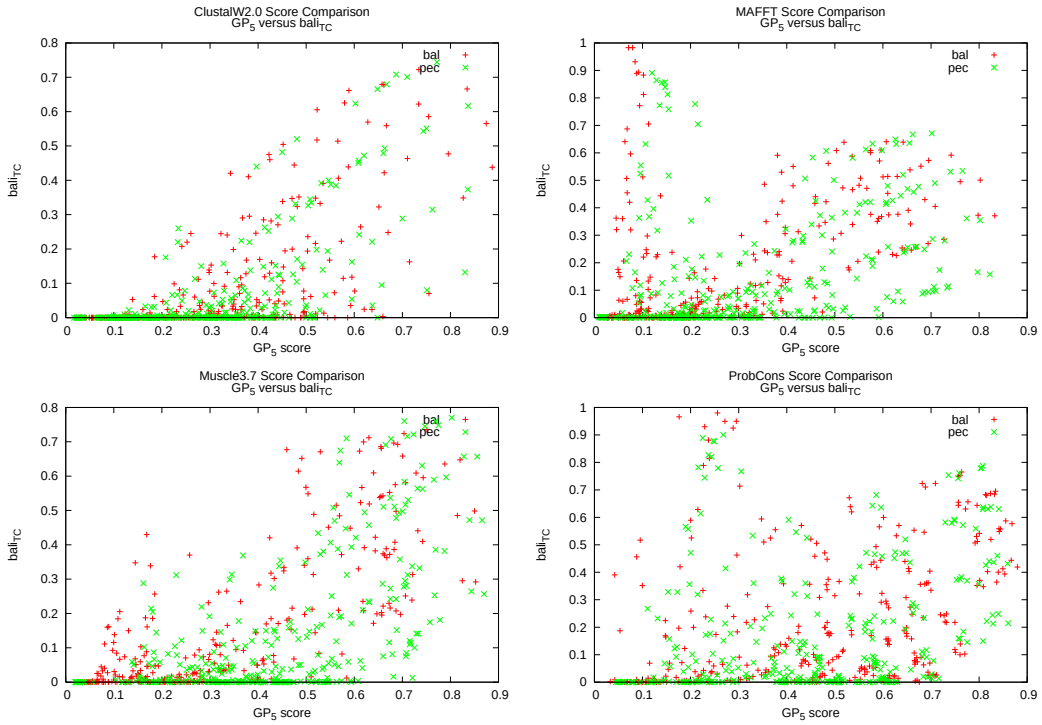



Figure 4.11: *Continued on next page.*

(C)




(D)

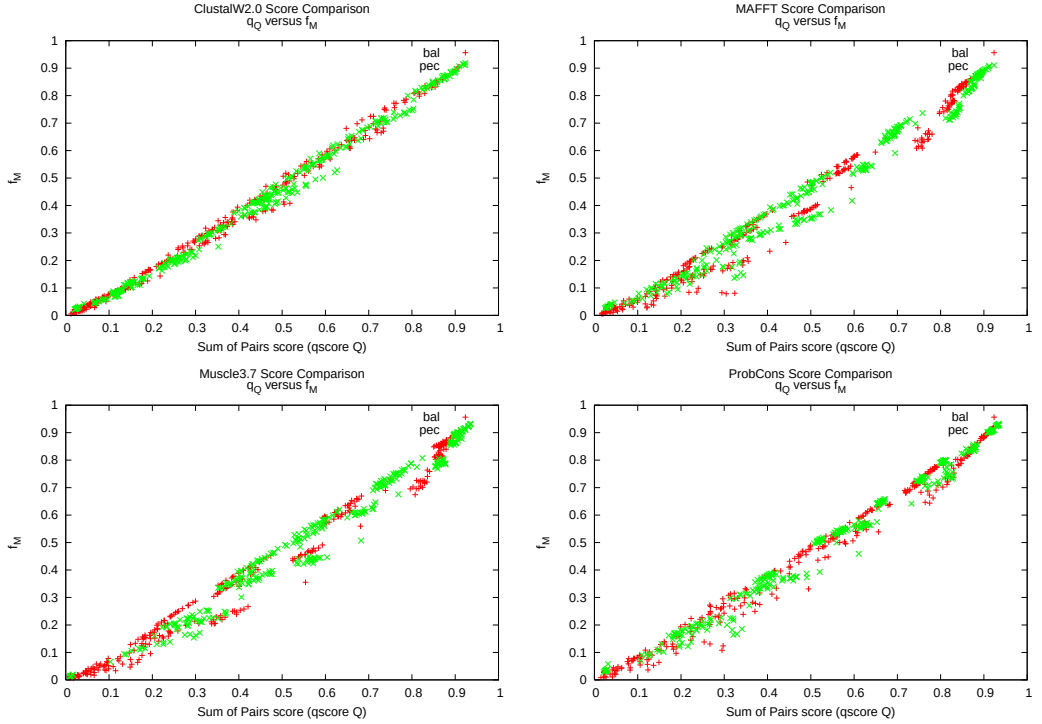



Figure 4.11: *Continued on next page.*

(E)

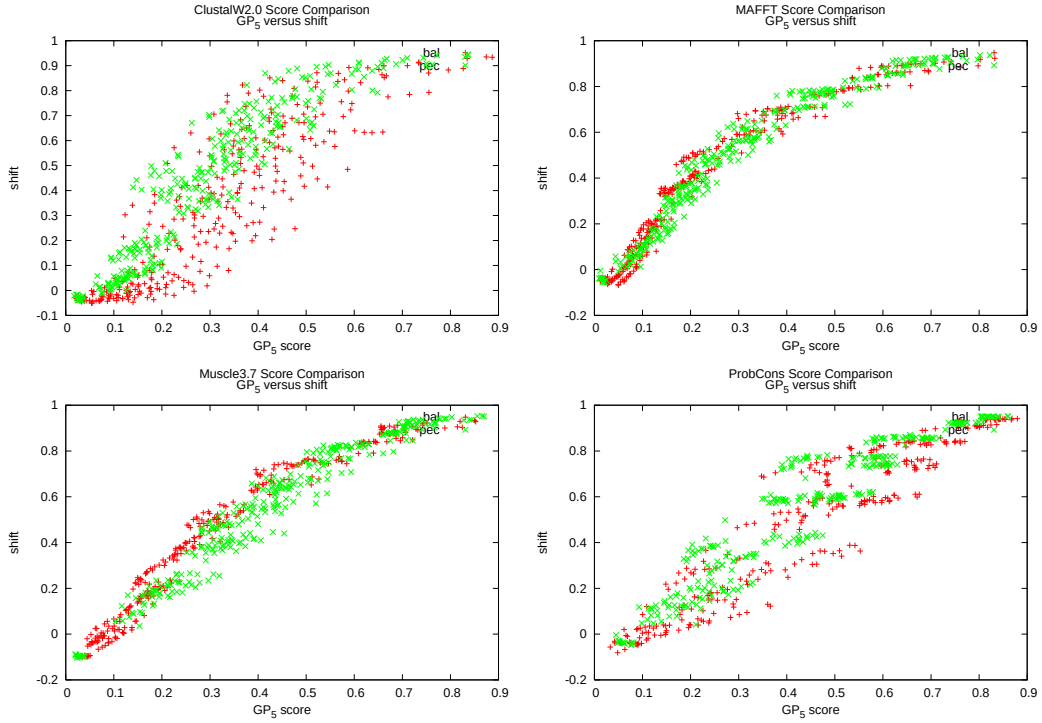



Figure 4.11: Comparison of scoring method performances on the benchmark dataset for ClustalW2.0 (top left), MAFFT-linsi (top right), Muscle3.7 (lower left), and ProbCons (lower right). (A) Gap profile score vs. $Q$ score. (B) gap profile score vs. $f_M$ score. (C) gap profile score vs. total column score. (D) $Q$ score vs. $f_M$ score. (E) gap profile score vs. shift score.

Table 4.4: The average root mean square deviation values for scoring metrics on reconstructed MSAs[a].

| | MSA method | $GP_5$ | $GP_{10}$ | $GP_{25}$ | $bali_{TC}$ | $q_Q$ | $f_M$ | shift |
|---|---|---|---|---|---|---|---|---|
| | ClustalW2.0 | – | 0.163 | **0.415** | 0.270 | 0.166 | *0.153* | 0.185 |
| $GP_5$ | MAFFT | – | *0.123* | **0.331** | 0.262 | 0.199 | 0.150 | 0.203 |
| | Muscle3.7 | – | 0.156 | **0.398** | 0.287 | 0.157 | *0.118* | 0.160 |
| | ProbCons | – | 0.180 | **0.405** | 0.363 | 0.150 | *0.124* | 0.145 |
| | ClustalW2.0 | *0.163* | – | 0.257 | **0.425** | 0.165 | 0.177 | 0.173 |
| $GP_{10}$ | MAFFT | 0.123 | – | 0.214 | **0.354** | 0.095 | *0.057* | 0.098 |
| | Muscle3.7 | 0.156 | – | 0.247 | **0.429** | *0.099* | 0.110 | 0.109 |
| | ProbCons | 0.180 | – | 0.234 | **0.503** | *0.175* | 0.192 | 0.177 |
| | ClustalW2.0 | 0.415 | *0.257* | – | **0.662** | 0.362 | 0.388 | 0.372 |
| $GP_{25}$ | MAFFT | 0.331 | 0.214 | – | **0.525** | *0.176* | 0.217 | 0.185 |
| | Muscle3.7 | 0.398 | *0.247* | – | **0.648** | 0.301 | 0.339 | 0.318 |
| | ProbCons | 0.405 | *0.234* | – | **0.683** | 0.363 | 0.395 | 0.379 |
| | ClustalW2.0 | *0.270* | 0.425 | **0.662** | – | 0.367 | 0.343 | 0.388 |
| $bali_{TC}$ | MAFFT | *0.262* | 0.354 | **0.525** | – | 0.400 | 0.375 | 0.422 |
| | Muscle3.7 | *0.287* | 0.429 | **0.648** | – | 0.410 | 0.377 | 0.417 |
| | ProbCons | *0.363* | 0.503 | **0.683** | – | 0.431 | 0.412 | 0.448 |
| | ClustalW2.0 | 0.166 | 0.165 | 0.362 | **0.367** | – | *0.037* | 0.071 |
| $q_Q$ | MAFFT | 0.199 | 0.095 | 0.176 | **0.400** | – | *0.067* | 0.080 |
| | Muscle3.7 | 0.157 | 0.099 | 0.301 | **0.410** | – | *0.059* | 0.068 |
| | ProbCons | 0.150 | 0.175 | 0.363 | **0.431** | – | *0.047* | 0.058 |
| | ClustalW2.0 | 0.153 | 0.177 | **0.388** | 0.343 | *0.037* | – | 0.067 |
| $f_M$ | MAFFT | 0.150 | *0.057* | 0.217 | **0.375** | 0.067 | – | 0.070 |
| | Muscle3.7 | 0.118 | 0.110 | 0.339 | **0.377** | 0.059 | – | *0.057* |
| | ProbCons | 0.124 | 0.192 | 0.395 | **0.412** | *0.047* | – | 0.053 |
| | ClustalW2.0 | 0.185 | 0.173 | 0.372 | **0.388** | 0.071 | *0.067* | – |
| shift | MAFFT | 0.203 | 0.098 | 0.185 | **0.422** | 0.080 | *0.070* | – |
| | Muscle3.7 | 0.160 | 0.109 | 0.318 | **0.417** | 0.068 | *0.057* | – |
| | ProbCons | 0.145 | 0.177 | 0.379 | **0.448** | 0.058 | *0.053* | – |

[a] The minimum and maximum RMSD values for each MSA-method/scoring-metric combination are italicized and boldfaced, respectively.

Table 4.4 shows the root mean square deviation (RMSD) values between scoring metrics over all MSA datasets. RMSD shows the level of correlations among scoring metrics. RMSD values near zero suggest that metric return values are very similar across all simulation conditions (correlated), while RMSD values near 1.0 suggest that metric return values are very dissimilar across all simulation conditions (not correlated). Many interesting observations can be made. (1) The column-based metric ($\text{bali}_{TC}$) tend to correlate with the strict pairwise indel metric ($\text{GP}_5$ score). (2) Character-based metrics tend to correlate with other character-based metrics. (3) Pairwise metrics tend to correlate with other pairwise-based metrics.

**Column-based measures tend to cluster with strict pairwise measures:** The $\text{GP}_5$ score and the $\text{bali}_{TC}$ score appear to be roughly equivalent; both metrics are maximally deviated from the most lenient score, $\text{GP}_{25}$ score. However, although $\text{bali}_{TC}$ is the least deviated from the $\text{GP}_5$ score, the $\text{GP}_5$ score is least deviated from the $f_M$ score (except for MAFFT). This suggests that the $\text{GP}_5$ score is a middle ground between the character-based methods and the column-based methods.

**Pairwise character-based metrics tend to correlate with other pairwise character-based metrics:** The differences between the pairwise character-based metrics ($\text{q}_Q$, $f_M$, and shift scores) is minimal. The $\text{bali}_{TC}$ metric consistently returns values that are the most deviated from these pairwise character-based metrics, with the exception of the $f_M$ metric on the ClustalW2.0 MSAs. The consistently low differences between the pairwise character-based metrics may reflect the global conditions of the benchmark datasets used, e.g., nucleotide data and global alignment.

**Pairwise metrics tend to correlate with other pairwise-based metrics:** All character-based metrics are highly deviated from the $\text{bali}_{TC}$ score, and the indel-

based metrics, except for $GP_5$, are highly deviated from the $bali_{TC}$ score. Indel-based metrics with low values of $w$ are generally most correlated with the character-based metrics. Although the pairwise character-based metrics tend to be lenient scoring measures, they do not correlate with the more lenient indel-based metrics. Rather, the pairwise character-based metrics score MSA alignments very differently than the lenient indel-based metrics, even though they are closely correlated with indel-based metrics with a low $w$. This suggests that the indel-based metrics are sensitive to different parameters with changing maximum gap distances.

As expected, three categories of scoring methods – pairwise character-based matching ($q_Q$, shift, and $f_M$ scores), pairwise indel-based matching (the $GP_w$ score), and columnar-based matching ($bali_{TC}$) – showed differences in scoring MSAs. Moreover, pairwise-based metrics, regardless of character-based or indel-based, performed similarly to column-based metrics.

**Attributes affecting scoring methods:** Here we examine MSA-scoring metric performances comparing against the parameters used to create benchmark datasets. Such comparisons should let us gain an understanding about the particular characteristics that affect scoring metrics.

The $GP_w$ score is highly sensitive to the tree type when analyzing ClustalW2.0 alignments (Figure 4.11E. This points to the potential impact of the higher number of indels that share evolutionary paths in balanced trees versus pectinate trees (from Table 4.2), combined with the tendency of ClustalW2.0 placing smaller indels in the MSA. Other MSA methods do not display this trend. We believe that this points towards ClustalW2.0 overfitting the character-based MSA scoring methods. This is reinforced by previous examples showing ClustalW2.0 as the only MSA method

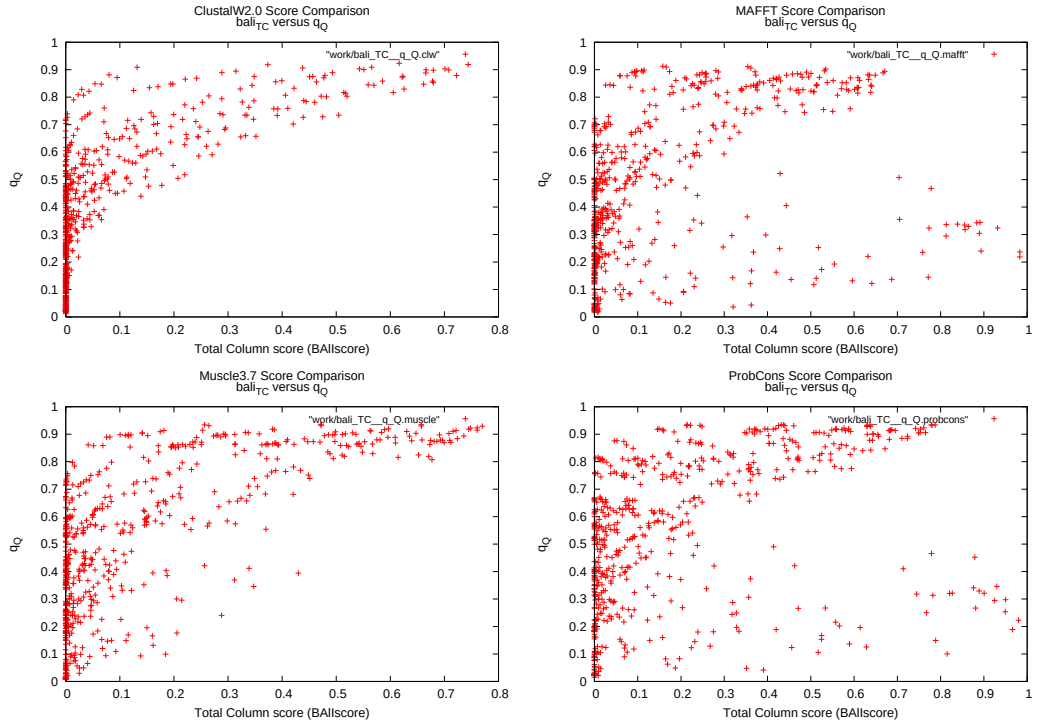



Figure 4.12: Comparison of the total column score ($\text{bali}_{TC}$) and the $Q$ score ($\text{q}_Q$).

that was less affected by the particular SPS and TC implementation being used (Figures 4.7,and 4.9 4.10, respectively).

Although the $\text{q}_Q$ score is assumed to be much more sensitive than the $\text{bali}_{TC}$ score, it is interesting to note that when compared to each other, the MAFFT and ProbCons MSAs display cases in which the $\text{bali}_{TC}$ score is high for very low values of the $\text{q}_Q$ score (Figure 4.12). This case occurs for datasets that are highly enriched for deletions, i.e., datasets afflicted with long tree lengths, large indels, and enriched in deletions. With such conditions, far more columns are preserved, while relatively few pairs of characters may be available to measure.

The gap distance thresholds ($w$) for $\text{GP}_w$ scores are irrelevant when either extremely difficult or extremely easy datasets are analyzed, where the metric value returned is 0 or 1 regardless of $w$, respectively. However, increasing values of $w$ for

other datasets increases the $GP_w$ score. Figure 4.13 shows the effects of the larger $w$ value.

### 4.2.2 Gap Profile and $Q$ scores versus Phylogenetic Reconstruction

Figure 4.14 shows the phylogeny reconstruction accuracies as a function of the $GP_5$ and the $q_Q$ score for the MSAs reconstructed by the four different MSA methods. The best case for the scoring metrics, when compared to topological accuracy, is to fall in the lower left or upper right quarters of the plot, which would show that the scoring metric correlated with the phylogenetic reconstruction accuracy. A slightly worse case is to fall in the lower right corner, meaning that low scores for the metric can lead to good phylogenetic reconstructions. The worst case is for the metric/phylogenetic-accuracy pair to fall in the upper left corner, meaning that the a good score for the metric leads to terrible trees.

The $GP_w$ score and $q_Q$ scores both show three tendencies. (1) In general, MSA accuracy and phylogenetic accuracy correlate. (2) The perfect phylogenetic reconstructions (phylogenetic reconstruction accuracy is 1) do not seem to correlate with the scoring metric. (3) Few datasets fall in the worst category (inaccurate phylogenies built from highly scored MSAs), and those that do tend to be towards the middle of the plot.

The $GP_5$ is more conservative than the $q_Q$ score. This means that far fewer points with $GP_5$ score fall in the upper left and upper right quadrants than those with the $q_Q$ score. The dataset points that fall in the upper left quadrant tend to be the difficult cases: 64 taxa, pectinate trees, enriched in deletions. Also of interest is the cluster of points around the $(0.6, 0.05)$ mark of the graph. These datasets are 8 taxon datasets

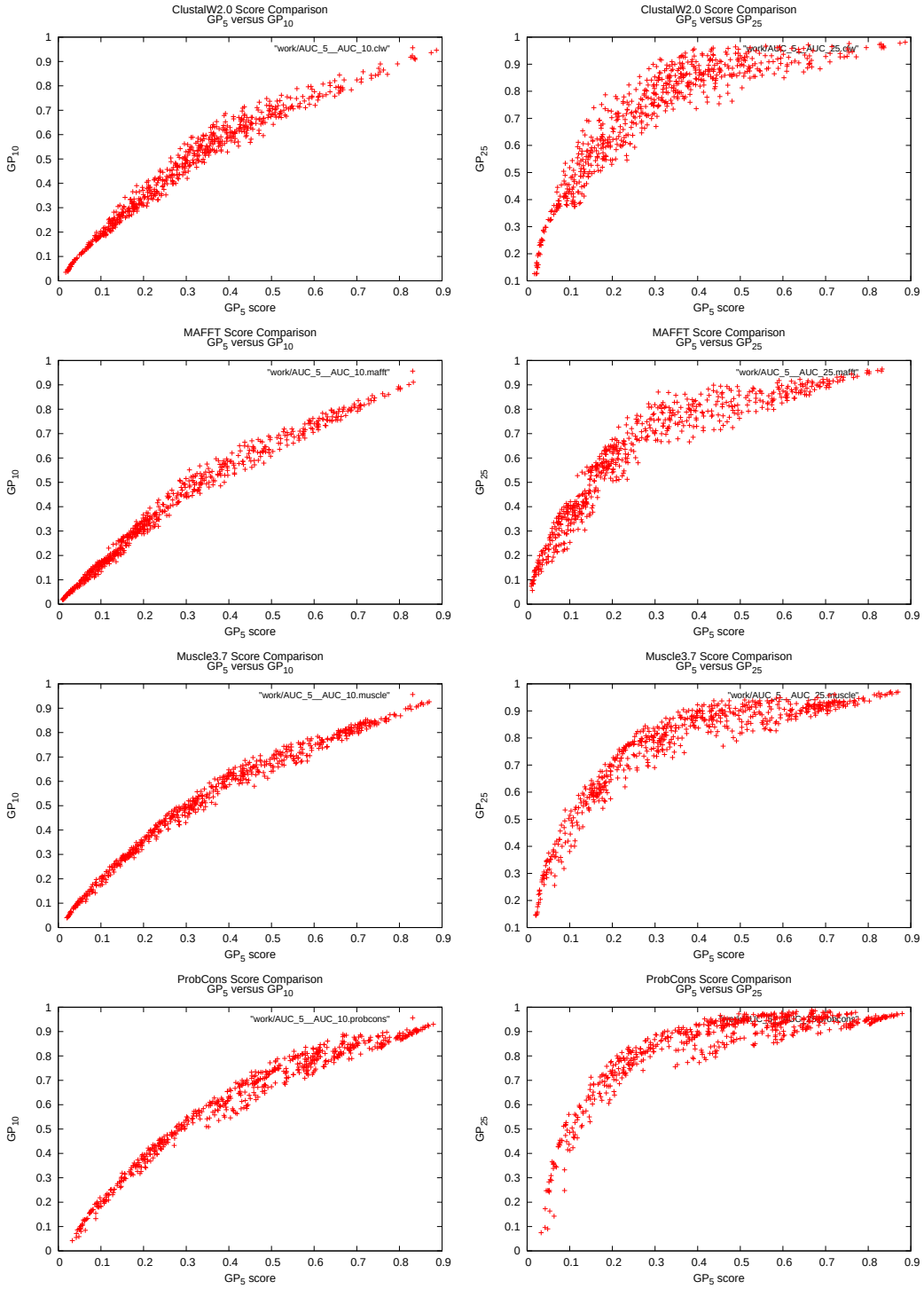



Figure 4.13: Comparison of the gap profile scores using different window sizes, $w$. *Left:* $w = 5$ vs. $w = 10$. *Right:* $w = 5$ vs. $w = 25$. *Figure continued on next page.*

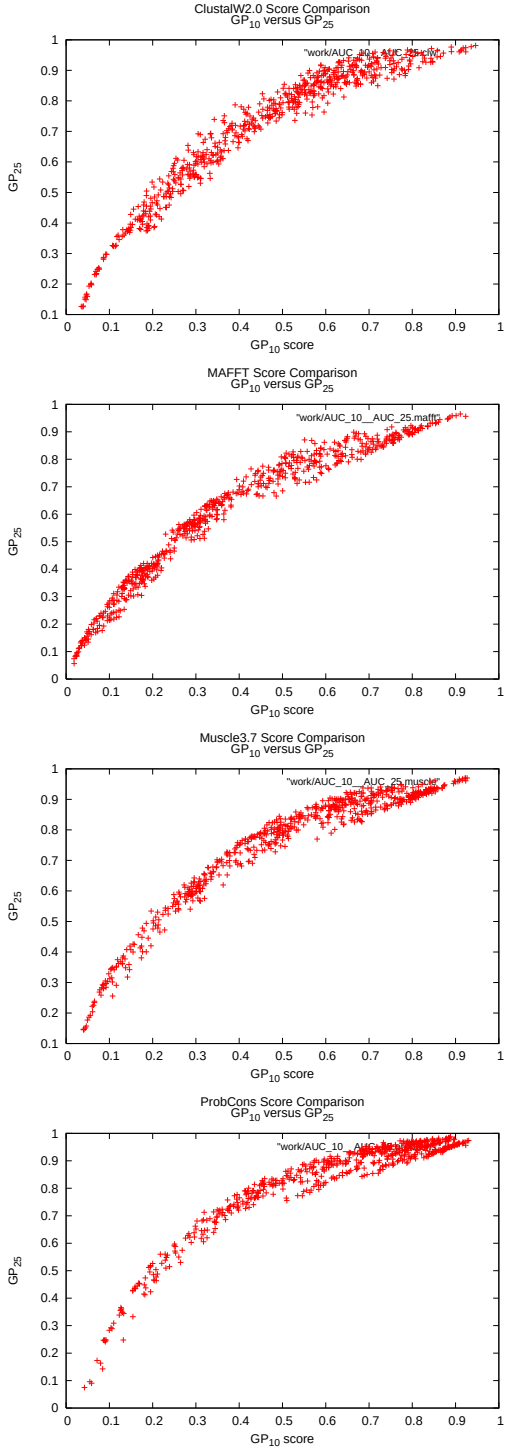



Figure 4.13: Comparison of the gap profile scores using different $w$ values. $w = 10$ vs. $w = 25$.

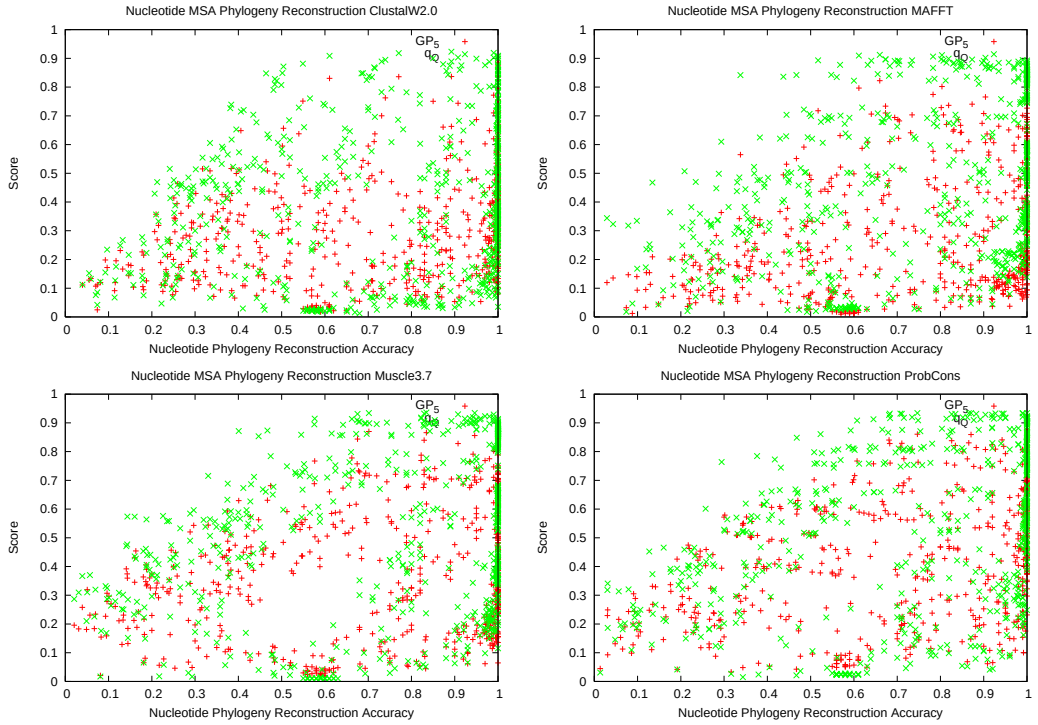



Figure 4.14: Comparison between the $GP_5$ score (red '+') and the $q_Q$ score (green 'x') versus the phylogenetic reconstruction accuracy for the MSAs reconstructed by different methods.

that are enriched in deletions, with tree lengths between 0.25–0.75 in length. These datasets are difficult for the $GP_5$ score since there are a limited number of indels to calculate the score. With respect to the $q_Q$ score, these datasets may contain fewer informative sites because of the proliferation of deletions and the limited number of taxa.

The $GP_5$ score can be helpful in judging accuracies of phylogenetic reconstructions because it returns conservate values. For example, if we get a $GP_5$ score of 0.8 for a MSA, we can expect the accuracy of the phylogenetic tree reconstructed to be around 0.6, 0.63, 0.7, and 0.7 for the MSA reconstructions provided by ClustalW2.0, MAFFT, Muscle3.7, and ProbCons, respectively. To achieve the same amount of certainty with the $q_Q$ score metric, we would have to have $q_Q$ scores to be 0.9 or higher. If we limited

the $q_Q$ score to be greater than 0.8, we can only expect phylogenetic reconstruction accuracies of 0.48, 0.33, 0.38, and 0.45, i.e., a difference of around 0.12, 0.3, 0.32, and 0.25 percent lower accuracies than the $GP_5$ score for ClustalW2.0, MAFFT, Muscle3.7, and ProbCons, respectively.

## 4.3 Conclusion

Current scoring metrics fall into two categories: character-based matching and columnar matching. Within the category, metrics tend to be highly correlated, and do not measure different characteristics of the MSAs. In this study, we have introduced another type of scoring metric, indel-based matching, specifically the gap profile ($GP_w$) score. The $GP_w$ score is only weakly correlated with character-based matching, but more importantly, is sensitive to different properties of MSAs. The introduction of a new scoring metric enables improved MSA goodness evaluation by combining different types of metrics.

The $GP_w$ score is a very flexible metric, allowing the tester to fine-tune the sensitivity of the $GP_w$ score by changing the value of $w$; thus, for conserved MSAs, one can set a very strict metric by setting $w$ to a low value. For harder MSAs, one can set a lenient metric by setting $w$ to a higher value. One weakness of the $GP_w$ score, however, is that it requires a gap-dense MSA in order to make a good judgement on how well a MSA method performs.

We showed that the $GP_5$ is a more specific measure than the SPS with respect to phylogenetic reconstruction accuracies. The $GP_5$ score has the added benefit of being able to tune the sensitivity by changing the value of $w$, either identifying more reconstructed MSAs that produce high quality phylogenies by seting $w > 5$, or identifying only those reconstructed MSAs that can be confidently predicted to generate

high quality phylogenies by setting $w$ to lower values.

Future versions of the $GP_w$ score can be aimed at changing the metric so that it is not context-free. In our current work, only the pairwise alignments were inspected, without the context of either the MSA or where the sequences are in the phylogeny. This context-free approach counted insertions and deletions equally in the scoring measure, the effect of which being that insertions and deletions will affect the score disproportionately depending on where they occur within the simulation (insertions add a gap site to all sequences in sister lineages, whereas deletions add a gap to all descendant sequences). Additionally, the $GP_w$ score also does not take into account start and end points in gaps, which could remove important data from gaps that occur in the same region. The $GP_w$ score, as presented here, however, still provides a useful, flexible metric that measures different parameters of datasets.

# Chapter 5

# Conclusion

Reconstructing the evolutionary history of biological sequences using a multiple sequence alignment (MSA) is the first step in almost all molecular evolutionary studies. Knowing the accuracy of MSA methods is necessary to be confident in the results from the secondary analyses such as phylogenetic reconstruction. In order to test the goodness of a MSA reconstruction, benchmark datasets are necessary. Unfortunately, current benchmark datasets built on real sequence data suffer from multiple shortcomings, and also do not contain the evolutionary relationships of the sequences in the dataset. In order to rectify these shortcomings, one must rely on simulated sequence datasets. Simulated datasets, in general, suffer from lack of realism. In order to introduce more realism into sequence simulation, we introduced a new method, indel-Seq-Gen.

To begin creating more realistic protein simulations than those available at the time, we created indel-Seq-Gen version 1.0 (iSGv1.0). iSGv1.0 incorporated indel simulations, and added the unique ability to simulate heterogeneous evolution of multidomain protein families through the addition of heterogeneous sequence simulation, the ability to change parameters such as the amino acid frequencies, substitution

matrices, proportion of invariable sites, indel probabilities, indel length distributions, and differing guide tree topologies and branch lengths. iSGv1.0 also added the unique ability to conserve sequences that require a certain length to conserve function, such as the *CXXC* motif in thioredoxin-fold proteins, while not restricting the sequence to undergo slower substitution evolution at the same time. We showcased the ability to improve sequence realism by simulating the transmembrane regions in G protein-coupled receptors (GPCRs) and the $\beta$-strands in the lipocalin superfamily. We further demonstrated the ability of iSGv1.0 to create highly-divergent sequences that maintain the sequence traits required for a family of proteins by examining the effects of high indel rates in phylogenetic reconstructions and using simulated sequences as query strings for BLAST and Pfam searches. With iSGv1.0, testing the performance of various molecular evolution and bioinformatics methods when they are applied to extremely divergent heterogeneous protein analysis became feasible.

Further improvement of the realism in generating highly diverged sequence superfamilies requires site- and lineage-specific evolution. To accomplish this, we upgraded iSGv1.0 to indel-Seq-Gen version 2.0 (iSGv2.0). iSGv2.0, in addition to incorporating DNA simulation, added lineage-specific evolution, motif conservation using PROSITE-like regular expressions, indel tracking, and a richer representation of subsequence length constraints that allows for sequences to change length, but constricts the sequence to a minimum and maximum length. In addition to the above improvements, sequence constraints imposed by insertions, deletions, and substitutions were formalized and implemented in iSGv2.0. We also uncovered, and fixed, a flaw in the indel models of most sequence simulators that biases simulation results for hypotheses involving indels, by introducing discrete steps model in iSGv2.0. We presented examples of the improvements in iSGv2.0 by using the calycin superfamily. We showed the capabilities of iSGv2.0 to represent realistic superfamilies by modeling the calycin

superfamily using two other protein simulation methods (iSGv1.0 and ROSE). We simulated the three structurally conserved regions among the lipocalins and the motif of another calycin subfamily, the avidins. We compared the true MSAs generated by the three methods and showed how iSGv2.0 maintained the subsequeunce length constraints. With iSGv2.0, we were able to create realistic datasets that can be used to test hypotheses involving MSA methodologies.

iSGv2.0 also enabled us to test the scoring metrics that measure the goodness of MSA reconstructions. Current MSA scoring metrics fall into two character-based categories: pairwise matching and columnar matching. Possessing the true indel histories and positions in the true MSA allowed us to introduce a new indel-based pairwise matching metric, the gap profile score. To test metric performances, we created a benchmark MSA dataset using iSGv2.0 that covers a large range of parameterizations, and populated the spectrum of datasets, from "easy" to "hard." The created benchmark varied five parameters: (1) tree shape, (2) number of taxa, (3) tree root-to-tip length, (4) $P_{ins}$:$P_{del}$ ratio, and (5) indel length.

We compared the gap profile scoring metric to the character-based scoring methods as implemented in the packages `qscore` and `baliscore`. We first showed the unexpected result that the implementations of the total column score in `qscore` and the sum of pairs score in `baliscore` were not consistent. We then showed that the gap profile score is affected in a different manner than both the pairwise and columnwise character-based scoring methods, though it is closer to a character-based method. Using the gap profile score to predict phylogenetic reconstruction accuracy for reconstructed MSAs showed that the gap profile score with $w = 5$ is a more specific measure than the the sum of pairs score, or $Q$ score. The tests conducted showed that the $Q$ score is more sensitive than the gap profile score with $w = 5$. However, the gap profile score is flexible and can be made more sensitive by increasing $w$.

To conclude, we introduced a hypothesis-testing toolkit for evaluating biological informativeness of indels. We introduced two versions of indel-Seq-Gen, each version improving on the state-of-the-art methods abilities to create realistic simulated sequences. We further introduced the first multiple sequence alignment scoring metric that use indels as the only informative data. Future versions of the gap profile score can be aimed at making the metric context-based. Currently, only the pairwise alignments were inspected, without the context of either the MSA or where the sequences are in the phylogeny. This context-free approach counted insertions and deletions equally in the scoring measure. With this approach, insertions and deletions affect the score disproportionately depending on where they occur within the simulation. Finally, the gap profile score can be improved by taking into account start and end points in gaps, conserving important information of gaps that occur in the same region.

# Bibliography

[1] J. Adachi and M. Hasegawa. MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood. *Computer Science Monographs*, 28:1–150, 1996.

[2] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:33893402, 1997.

[3] F. Armougom, S. Moretti, V. Keduas, and C. Notredame. The iRMSD: a local measure of sequence alignment accuracy using structural information. *Bioinformatics*, 22:e35–e39, 2006.

[4] T.K. Attwood, M.E. Beck, A.J. Bleasby, and D.J. Parry-Smith. PRINTS – a database of protein motif fingerprints. *Nucleic Acids Research*, 22:3590–3596, 1994.

[5] A. Bairoch, R. Apweiler, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, and L.S. Yeh. The universal protein resource (UniProt). *Nucleic Acids Research*, 33:D154–D159, 2005.

[6] A. Bairoch and B. Boeckmann. The SWISS-PROT protein sequence data bank. *Nucleic Acids Research*, 19 Suppl:2247–2249, 1991.

[7] A. Bateman, L. Coin, R. Durbin, R.D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E.L.L. Sonnhammer, D.J. Studholme, C. Yeats, and S.R. Eddy. The Pfam protein families database. *Nucleic Acids Research*, 32:D138–D141, 2004.

[8] S.A. Benner, M. Cohen, and G. Gonnet. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *Journal of Molecular Biology*, 229:1065–1082, 1993.

[9] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shyndyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[10] G. Blackshields, I.M. Wallace, M. Larkin, and D.G. Higgins. Analysis and comparison of benchmarks for multiple sequence alignment. *In Silico Biology*, 6, 2006.

[11] N.S. Boutonnet, M.J. Rooman, M.E. Ochagavia, J. Richelle, and S.J. Wodak. Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. *Protein Engineering Design and Selection*, 8:647–662, 1995.

[12] R.K. Bradley and I.H. Holmes. Transducers: an emerging probabilistic framework for modeling indels on trees. *Bioinformatics*, 23:3258–3262, 2007.

[13] L. Lo Cante, B. Ailey, T.J.P. Hubbard, S.E. Brenner, A.G. Murzin, and C. Chothia. SCOP: a Structural Classification of Proteins database. *Nucleic Acids Research*, 28:257–259, 2000.

[14] R.A. Cartwright. DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics*, 21:iii31–iii38, 2005.

[15] Mike S. S. Chang and S.A. Benner. Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *Journal of Molecular Biology*, 341:617–631, 2004.

[16] P.T. Chivers, M.C. Laboissière, and R.T. Raines. The CXXC motif: imperatives for the formation of native disulfide bonds in the cell. *EMBO Journal*, 15:2659–2667, 1996.

[17] P.Y. Chou and G.D. Fasman. Prediction of protein conformation. *Biochemistry*, 13:222–245, 1974.

[18] M. Cline, R. Hughey, and K. Karplus. Predicting reliable regions in protein sequence alignments. *Bioinformatics*, 18:306–314, 2002.

[19] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model for evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, volume 5, pages 345–352, Washington DC, 1978. National Biochemical Research Foundation.

[20] C.B. Do, M.S. Mahabhashyam, M. Brudno, and S. Batzoglou. ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15:330–340, 2005.

[21] S.R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14:755–763, 1998.

[22] R. Edgar. personal communication.

[23] R. Edgar. http://www.drive5.com/qscore/.

[24] R.C. Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5, 2004.

[25] R.C. Edgar and K. Sjölander. COACH: profile-profile alignment of protein families using hidden Markov models. *Bioinformatics*, 20:1309–1318, 2004.

[26] J. Felsenstein. PHYLIP Phylogeny Inference Package Version 3.65, 2005. Distributed by the author. Department of Genetics, University of Washington, Seattle.

[27] J. Felsenstein. PHYLIP Phylogeny Inference Package Version 3.68, 2008. Distributed by the author. Department of Genetics, University of Washington, Seattle.

[28] D.-F. Feng and R.F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[29] D.R. Flower, A.C.T. North, and C.E. Sansom. The lipocalin protein family: structural and sequence overview. *Biochimica et Biophysica Acta*, 1482:9–24, 2000.

[30] T. Fuchs, G. Glusman, S. Horn-Saban, D. Lancet, and Y. Pilpel. The human olfactory subgenome: from sequence to structure and evolution. *Human Genetics*, 108:1–13, 2001.

[31] Y. Gilad, O. Man, and G. Glusman. A comparison of the human and chimpanzee olfactory receptor gene repertoires. *Genome Research*, 15:224–230, 2005.

[32] T. Golubchik, M.J. Wise, S. Easteal, and L.S. Jermiin. Mind the gaps: evidence of bias in estimates of multiple sequence alignments. *Molecular Biology and Evolution*, 24:2433–2442, 2007.

[33] N.C.W. Goonesekere and B. Lee. Frequency of gaps observed in a structurally aligned protein pair database suggests a simple gap penalty function. *Nucleic Acids Research*, 32:2838–2843, 2004.

[34] O. Gotoh. A weighting system and algorithm for aligning many phylogenetically related sequences. *Computer Applications in the Biosciences*, 9:361–370, 1995.

[35] N.C. Grassly, J. Adachi, and A. Rambaut. PSeq-Gen: an application for the monte carlo simulation of protein sequence evolution along phylogenetic trees. *Bioinformatics*, 13:559–560, 1997.

[36] B.G. Hall. Simulating DNA coding sequence evolution with EvolveAGene 3. *Molecular Biology and Evolution*, 25:688–695, 2008.

[37] M. Hasegawa, H. Kishino, and T. Yano. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22:672–677, 1985.

[38] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceeds from the National Academy of Sciences, USA*, 15:10915–10919, 1992.

[39] H. Hermjakob, F. Lang, and R. Apweiler. SPTR – a comprehensive, non-redundant and up-to-date view of the protein sequence world. *CCP11 Newsletter*, 2, 1998.

[40] M. H ohl and M.A. Ragan. Is multiple-sequence alignment required for accurate inference of phylogeny. *Systematic Biology*, 56:206–221, 2007.

[41] T.J. Hubbard, B. Ailey, S.E. Brenner, A.G. Murzin, and C. Chothia. SCOP: a structural classification of proteins database. *Nucleic Acids Research*, 27:254–256, 1999.

[42] N. Jardine and D. McKenzie. Continental drift and the dispersal and evolution of organisms. *Nature*, 235:20–24, 1972.

[43] D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999.

[44] D.T. Jones, W.R. Taylor, and J.M. Thornton. The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences*, 8:275–282, 1992.

[45] P. Joost and A. Methner. Phylogenetic analysis of 277 human G-protein-coupled receptors as a tool for the prediction of orphan receptor ligands. *Genome Biology*, 3:research0063.1–0063.16, 2002.

[46] K. Karplus and B. Hu. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics*, 17:713–720, 2001.

[47] K. Katoh, K. Kuma, T. Miyata, and H. Toh. Improvement in the accuracy of multiple sequence alignment program MAFFT. *Genome Informatics*, 16:22–33, 2005.

[48] K. Katoh, K. Misawa, K. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment base on fast Fourier transform. *Nucleic Acids Research*, 30:3059–3066, 2002.

[49] M. Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.

[50] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.G. Gibson, and D.G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23:2947–2948, 2007.

[51] T. Lassmann and E. Sonnhammer. Quality assessment of multiple alignment programs. *FEBS letters*, 529:126–130, 2002.

[52] D.J. Lipman. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

[53] K. Liu, S. Raghavan, S. Nelesen, C.R. Linder, and T. Warnow. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324:1561–1564, 2009.

[54] A Lötynoja and N. Goldman. Phylogeny-aware gap placement prevens errors in sequence alignment and evolutionary analysis. *Science*, 320:1632–1635, 2008.

[55] K. Mizuguchi, C.M. Deane, T.L. Blundell, and J.P. Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

[56] E.N. Moriyama. http://bioinfolab.unl.edu/emlab/index.html.

[57] D.A. Morrison. Why would phylogeneticists ignore computerized sequence alignment. *Systematic Biology*, 58:150–158, 2009.

[58] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[59] P.C. Ng, J.G. Henikoff, and S. Henikoff. PHAT: a transmembrane-specific substitution matrix. *Bioinformatics*, 16(9):760–766, 2000.

[60] C. Notredame. Recent evolutions of multiple sequence alignment algorithms. *PLoS Computational Biology*, 3:1405–1408, 2007.

[61] C. Notredame, D.G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302:205–217, 2000.

[62] O. O'Sullivan, M. Zehnder, D. Higgins, P. Bucher, A. Grosdidier, and C. Notredame. APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, 19:i215–i221, 2003.

[63] A.R. Panchenko and T. Madej. Structural similarity of loops in protein families: toward the understanding of protein evolution. *BMC Evolutionary Biology*, 5, 2005.

[64] A. Pang, A.D. Smith, P.A.S. Nuin, and E.R.M. Tillier. SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. *BMC Bioinformatics*, 6:236, 2006.

[65] S. Pascarella and P. Argos. Analysis of insertions/deletions in protein structures. *Journal of Molecular Biology*, 224:461–471, 1992.

[66] J. Pei, B.-H. Kim, and N.V. Grishin. PROMALS3D: a tool for multiple sequence and structure alignment. *Nucleic Acids Research*, 36:2295–2300, 2008.

[67] B. Qian and R.A. Goldstein. Distribution of indel lengths. *Proteins: Structure, Function, and Bioinformatics*, 45:102–104, 2001.

[68] G.P.S. Raghava, S.M.J. Searle, P.C. Audley, J.D. Barber, and G.J. Barton. OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, 4, 2003.

[69] A. Rambaut and N.C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Bioinformatics*, 13:235–238, 1997.

[70] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[71] M.S. Rosenberg. MySSP: Non-stationary evolutionary sequence simulation, including indels. *Evolutionary Bioinformatics Online*, 1:81–83, 2005.

[72] B. Rost. Twilight zone of protein sequence alignments. *Protein Engineering*, 12:85–94, 1999.

[73] R.B. Russell and G.J. Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, 14:309–323, 1992.

[74] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[75] J.M. Sauder, J.W. Arthur, and R.L. Dunbrack, Jr. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, 40:6–22, 2000.

[76] J. Schultz, R.R. Copley, T. Doerks, C.P. Ponting, and P. Bork. SMART: a web-based tool for the study of genetically mobile domains. *Nucleic Acids Research*, 28:231–234, 2000.

[77] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering Design and Selection*, 11:739–747, 1998.

[78] A.S. Siddiqui, U. Dengler, and G.J. Barton. 3Dee: a database of protein structural domains. *Bioinformatics*, 17:200–201, 2001.

[79] C.J.A. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P. Bucher. PROSITE: a documented database using patterns and profiles as motif descriptors. *Briefings in Bioinformatics*, 3:265–274, 2002.

[80] D. Sánchez, M.D. Ganfornina, G. Gutiérrez, and A. Marín. Exon-intron structure and evolution of the lipocalin gene family. *Molecular Biology and Evolution*, 20:775–783, 2003.

[81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[82] K.A. Spackman. Signal detection theory: valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160–163, San Mateo, CA, 1989. Morgan Kaufmann.

[83] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22:2688–2690, 2006.

[84] J. Stoye, D. Evers, and F. Meyer. Generating benchmarks for multiple sequence alignments and phylogenetic reconstructions. In *Proceedings International Conference on Intelligent Systems for Molecular Biology*, pages 303–306, 1997.

[85] J. Stoye, D. Evers, and F. Meyer. ROSE: generating sequence families. *Bioinformatics*, 14:157–163, 1998.

[86] C.L. Strope, K. Abel, S.D. Scott, and E.N. Moriyama. Biological sequence simulation for complex evolutionary hypotheses with indel-Seq-Gen version 2. *Molecular Biology and Evolution*, 26:2581–2593, 2009.

[87] C.L. Strope, S.D. Scott, and E.N. Moriyama. indel-Seq-Gen: A new protein family simulator incorporating domains, motifs, and indels. *Molecular Biology and Evolution*, 24:640–649, 2007.

[88] A.R. Subramanian, J. Weyer-Menkhoff, M. Kaufmann, and B. Morgenstern. DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, 6, 2005.

[89] D. Swofford. *PAUP*: Phylogenetic analysis using parsimony (and other methods)*. Sinauer Associates, Sunderland, MA, 2003. Version 4.0.

[90] G. Talavera and J. Castrensana. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic Biology*, 56:564–577, 2007.

[91] S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *American Mathematical Society: Lectures on mathematics in the life sciences*, 17:57–86, 1986.

[92] J.D. Thompson, P. Koehl, R. Ripp, and O. Poch. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, 61:127–136, 2005.

[93] J.D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27:2682–2690, 1999.

[94] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[95] J. Thorne. Models of protein sequence evolution and their applications. *Current Opinion in Genetics and Development*, 10:602–605, 2000.

[96] G.E. Tusnády and I. Simon. The HMMTOP transmembrane topology prediction server. *Bioinformatics*, 17:849–850, 2001.

[97] BAliBASE v3.0. http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE/bali_score.c.

[98] I. van Walle, I. Lasters, and L. Wyns. SABmark – a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21:1267–1268, 2005.

[99] A. Varadarajan, R.K. Bradley, and I.H. Holmes. Tools for simulating evolution of aligned genomic regions with integrated parameter estimation. *Genome Biology*, 9:R147, 2008.

[100] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1:337–348, 1994.

[101] NemATOL website. (nematode branch of the assembling the tree of life projects), 2008. `http://nematol.unh.edu/tree/tree1/vIch20ct682_c1w1.aln`.

[102] X. Xia and Z. Xie. Protein structure, neighbor effect, and a new index of amino acid dissimilarities. *Molecular Biology and Evolution*, 19:58–67, 2002.

[103] Z. Yang. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10:1396–1401, 1993.

[104] Z. Yang. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39:105–111, 1994.

[105] Z. Yang. PAML: a program package for phylogenetic analysis by maximum likelihood. *Computer Applications in the Biosciences*, 13(5):555–556, 1997.

[106] Z. Yang and S. Kumar. Approximate methods for estimating the pattern of nucleotide substitution and the variation of substitution rates among sites. *Molecular Biology and Evolution*, 13(5):650–659, 1996.

# Appendix A

# indel-Seq-Gen version 1.0 Supplementary Material

## A.1 Template Alignments

a.

```
          0000000000000000000000000000000000000000000000000000000000002000
OR4C6_HUMAN  ------------------ENQN--NV--TEF--ILLGLTENLELWK-IFSAVFLVMYVAT
OR4N4_HUMAN  ----------------KIANNT--VV--TEF--ILLGLTQSQDIQL-LVFVLILIFYLII
O12D3_HUMAN  ------------------ENVT--TM--NEF--LLLGLTGVQELQP-FFFGIFLIIYLIN
O12D2_HUMAN  ------------------LNTT--SV--TEF--LLLGVTDIQELQP-FLFVVFLTIYFIS
O1078_RAT    ----------------DSSNRT--RV--SEF--LLLGFVENKDLQP-LIYGLFLSMYLVT
OR7AH_HUMAN  ----------------EPENDT--GI--SEF--VLLGLSEEPELQP-FLFGLFLSMYLVT
OR1G1_PANTR  ----------------EGKNLT--SI--SEF--FLLGFSEQLEEQK-ALFGSFLFMYLVM
OLFI8_RAT    ------------------NNKT--VI--THF--LLLGLPIPPEHQQ-LFFALFLIMYLTT
OR6J1_HUMAN  ------------------GNWT-AAV--TEF--VLLGFSLSREVEL-LLLVLLLPTFLLT
OL287_RAT    -------------AWSTGQNLS--TP--GPF--ILLGFPGPRSMRI-GLFLLFLVMYLLT
O11A1_HUMAN  -------------EIVSTGNET---I--TEF--VLLGFYDIPELHF-LFFIVFTAVYVFI
O11H6_HUMAN  FFIIHSLVTSVFLTALGPQNRTMHFV--TEF--VLLGFHGQREMQS-CFFSFILVLYLLT
O10K1_HUMAN  ----------------EQVNKT--VV--REF--VVLGFSSLARLQQ-LLFVIFLLLYLFT
OR2K1_HUMAN  ----------------QGENFT--IW--SIF--FLEGFSQYPGLEV-VLFVFSLVMYLTT
O13C3_HUMAN  ----------------GEINQT--LV--SEF--LLLGLSGYPKIEI-VYFALILVMYLVI
OR2F1_HUMAN  ----------------GTDNQT--WV--SEF--ILLGLSSDWDTRV-SLFVLFLVMYVVT
O2A12_HUMAN  -----------------ESNQT--WI--TEV--ILLGFQVDPALEL-FLFGFFLLFYSLT
OR9G1_HUMAN  ----------------QRSNHT---V--TEF--ILLGFTTDPGMQL-GLFVVFLGVYSLT
OR9G5_HUMAN  ----------------QRSNHT---V--TEF--ILLGFTTDPGMQL-GLFVVFLGVYSLT
OLF2_CANFA   ----------------DGKNCS--SV--NEF--LLVGISNKPGVKV-TLFITFLIVYLII
O1044_MOUSE  ----------------AQINCT--QV--TEF--ILVGLTDREELKM-PLFVVFLSIYLFT
OR8D2_HUMAN  ----------------ATSNHS--SG--AEF--ILAGLTQRPELQL-PLFLLFLGIYVVT
OL502_MOUSE  -------------AFLEDGNHT--AV--TGF--ILLGLTDDPVLRV-VLFVIILCIYLVT
OR3A3_PANTR  -------------ESEAGTNRT--AV--AEF--MLLGLVQTEEMQS-VIFVLLLFAYLVT
OR3A1_HUMAN  -------------QPESGANGT--VI--AEF--ILLGLLEAPGLQP-VVFVLFLFAYLVT
O10AD_HUMAN  -----------------LRNGS--IV--TEF--ILVGFQQSSTSTRALLFALFLALYSLT
O56A4_HUMAN  ----------------ASPNNDSTAPV--SEF--LLICFPNFQSWQH-WLSLPLSLLFLLA
O51A4_HUMAN  ----------------SIINTSYVEI--TTF--FLVGMPGLEYAHI-WISIPICSMYLIA
OPSD_BOVIN   ------NGTEGPNFYVPFSNKT--GVVRSPFEAPQYYLAEPWQFSM-----LAAYMFLLI
```

Figure A.1

```
             00200020000000000000000000000001000200000000100000000000000
OR4C6_HUMAN  VLE---NLLIV-VTIITSQSLRS-PMYFFLTFLSLLDVMFSSVVAPKVIVDTLSKSTTIS
OR4N4_HUMAN  LPG---NFLII-FTIRSDPGL-TAPLYFFLGNLAFLDASYSFIVAPRMLVDFFSEKKVIS
O12D3_HUMAN  LIG---NGSIL-VMVVLEPQLHS-PMYFFLGNLSCLDISYSSVTLPKLLVNLVCSRRAIS
O12D2_HUMAN  VTG---NGAVL-MIVISDPRLHS-LMYFFLGNLSYLDICYSTVTLPKMLQNFLSTHKAIS
O1078_RAT    VIG---NISII-VAIISDPCLHT-PMYFFLSNLSFVDICFISTTVPKMLVNIQTQNNVIT
OR7AH_HUMAN  VLG---NLLII-LATISDSHLHT-PMYFFLSNLSFADICFISTTIPKMLINIQTQSRVIT
OR1G1_PANTR  VAG---NLLII-LVIITDTQLHT-PMYFFLANLSLADACFVSTTVPKMLANIQIQSQAIS
OLFI8_RAT    FLG---NLLIV-VLVQLDSHLHT-PMYLFLSNLSFSDLCFSSVTMLKLLQNIQSQVPSIS
OR6J1_HUMAN  LLG---NLLII-STVLSCSRLHT-PMYFFLCNLSILDILFTSVISPKVLANLGSRDKTIS
OL287_RAT    VVG---NLAII-SLVGAHRCLQT-PMYFFLCNLSFLEIWFTTACVPKTLATFAPRGGVIS
O11A1_HUMAN  IIG---NMLII-VAVVSSQRLHK-PMYIFLANLSFLDILYTSAVMPKMLEGFL-QEATIS
O11H6_HUMAN  LLG---NGAIV-CAVKLDRRLHT-PMYILLGNFAFLEIWYISSTVPNMLVNILSEIKTIS
O10K1_HUMAN  LGT---NAIII-STIVLDRALHT-PMYFFLAILSCSEICYTFVIVPKMLVDLLSQKKTIS
OR2K1_HUMAN  LLG---NSTLI-LITILDSRLKT-PMYLFLGNLSFMDICYTSASVPTLLVNLLSSQKTII
O13C3_HUMAN  LIG---NGVLI-IASIFDSHFHT-PMYFFLGNLSFLDICYTSSSVPSTLVSLISKKRNIS
OR2F1_HUMAN  VLG---NCLIV-LLIRLDSRLHT-PMYFFLTNLSLVDVSYATSVVPQLLAHFLAEHKAIP
O2A12_HUMAN  LMG---NGIIL-GLIYLDSRLHT-PMYVFLSHLAIVDMSYASSTVPKMLANLVMHKKVIS
OR9G1_HUMAN  VVG---NSTLI-VLICNDSCLHT-PMYFFTGNLSFLDLWYSSVYTPKILVTCISEDKSIS
OR9G5_HUMAN  VVG---NSTLI-VLICNDSHLHT-PMYFVVGNLSFLDLWYSSVYTPKILVICISEDKSIS
OLF2_CANFA   LVA---NLGMI-ILIRMDSQLHT-PMYFFLSHLSFSDARYSTAVGPRMLVGFIAKNKSIP
O1044_MOUSE  TLG---NLGLI-LVIRTDARLHT-PMYFFLSNLAFVDFCYSSVITPKMLGNFLYKQNMIS
OR8D2_HUMAN  VVG---NLGMI-FLIALSSQLYP-PVYYFLSHLSFIDLCYSSVITPKMLVNFVPEENIIS
OL502_MOUSE  VSG---NLSTI-LLIRVSSQLHH-PMYFFLSHLASADIGYSSSVTPNMLVNFLVERNTIS
OR3A3_PANTR  TGG---NLSIL-AAILVEPKLHT-PMYFFLGNLSVLDVGCITVTVPAMLGRLLSHKSTIS
OR3A1_HUMAN  VRG---NLSIL-AAVLVEPKLHT-PMYFFLGNLSVLDVGCISVTVPSMLSRLLSRKRAVP
O10AD_HUMAN  MAM---NGLIIFITSWTDPKLNS-PMYFFLGHLSLLDVCFITTTIPQMLIHLVVRDHIVS
O56A4_HUMAN  MGA---NTTLL-ITIQLEASLHQ-PLYYLLSLLSLLDIVLCLTVIPKVLAIFWFDLRSIS
O51A4_HUMAN  ILG---NGTIL-FIIKTEPSLHE-PMYYFLSMLAMSDLGLSLSSLPTVLSIFLFNAPEIS
OPSD_BOVIN   MLGFPINFLTLYVTVQ-HKKLRT-PLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFG
```

Figure A.1

```
             0001001000000000000000003333222223001010000000000000000001000
OR4C6_HUMAN  LKGCLTQLFVEHFFGGVGIILLTVMAYDRYVAICKPLHYTIIMSPR-VCCLMVGGAWV-G
OR4N4_HUMAN  YRGCITQLFFLHFLGGGEGLLLVVMAFDRYIAICRPLHCSTVMNPR-ACYAMMLALWL-G
O12D3_HUMAN  FLGCITQLHFFHFLGSTEAILLAIMAFDRFVAICNPLRYTVIMNPQ-VCILLAAAAWL-I
O12D2_HUMAN  FLGCISQLHFFHFLGSTESMLFAVMAFDLSVAICKPLRYTVIMNPQ-LCTQMAITIWV-I
O1078_RAT    YAGCITQIYFFLLFVELDNFLLTIMAYDRYVAICHPMHYTVIMNYK-LCGFLVLVSWI-V
OR7AH_HUMAN  YAGCITQMCFFVLFGGLDSLLLAVMAYDRFVAICHPLHYTVIMNPR-LCGLLVLASWM-I
OR1G1_PANTR  YSGCLLQLYFFMLFVMLEAFLLAVMAYDHYVAICHPLHYILIMSPG-LCVFLVSASWI-M
OLFI8_RAT    YAGCLTQIFFFLLFGYLGNFLLVAMAYDRYVAICFPLHYTNIMSHK-LCTCLLLVFWI-M
OR6J1_HUMAN  FAGCITQCYFYFFLGTVEFLLLTVMSYDRYATICCPLRYTTIMRPS-VCIGTVVFSWV-G
OL287_RAT    LAGCATQMYFVFSLGCTEYFLLAVMAYDRYLAICLPLRYGGIMTPG-LAMRLALGSWL-C
O11A1_HUMAN  VAGCLLQFFIFGSLATAECLLLAVMAYDRYLAICYPLHYPLLMGPR-RYMGLVVTTWL-S
O11H6_HUMAN  FSGCFLQFYFFFSLGTTECFFLSVMAYDRYLAICRPLHYPSIMTGK-FCIILVCVCWV-G
O10K1_HUMAN  FLGCAIQMFSFLFFGSSHSFLLAAMGYDRYMAICNPLRYSVLMGHG-VCMGLMAAACA-C
OR2K1_HUMAN  FSGCAVQMYLSLAMGSTECVLLAVMAYDRYVAICNPLRYSIIMNRC-VCARMATVSWV-T
O13C3_HUMAN  FSGCAVQMFFGFAMGSTECLLLGMMAFDRYVAICNPLRYPIILSKV-AYVLMASVSWL-S
OR2F1_HUMAN  FQSCAAQLFFSLALGGIEFVLLAVMAYDRYVAVCDALRYSAIMHGG-LCARLAITSWV-S
O2A12_HUMAN  FAPCILQTFLYLAFAITECLILVMMCYDRYVAICHPLQYTLIMNWR-VCTVLASTCWI-F
OR9G1_HUMAN  FAGCLCQFFFSAGLAYSECYLLAAVAYDRYVAISKPLLYAQAMSIK-LCALLVAVSYC-G
OR9G5_HUMAN  FAGCLCQFFFSAGLAYSECCLLAAMAYDRYVAISKPLLYAQAMSIK-LCALLVAVSYC-G
OLF2_CANFA   FYSCAMQWLVFCTFVDSECLLLAVMAFDRYKAISHPLLYTVSMSSR-VCSLLMAGVYL-V
O1044_MOUSE  FNACAAQLGCFLAFMTAECLLLASMAYDRYVAICNPLLYMVLMSPG-ICFQLVAAPYS-Y
OR8D2_HUMAN  FLECITQLYFFLIFVIAEGYLLTAMEYDRYVAICRPLLYNIVMSHR-VCSIMMAVVYS-L
OL502_MOUSE  YLGCGIQLGSAVFFGTVECFLLAAMAYDRFIAICSPLLYSNKMSTQ-VCVQLLVGSYI-G
OR3A3_PANTR  YDACLSQLFFFHLLAGMDCFLLTAMAYDRFLAICRPLTYSTHMNQR-VQRMLVAVSWT-C
OR3A1_HUMAN  CGACLTQLFFFHLFVGVDCFLLTAMAYDQFLAICRPLTYSTRMSQT-VQRMLVAASWA-C
O10AD_HUMAN  FVCCMTQMYFVFCVGVAECILLAFMAYDRYVAICYPLNYVPIISQK-VCVRLVGTAWF-F
O56A4_HUMAN  FPACFLQMFIMNSFLTMESCTFMVMAYDRYVAICHPLRYPSIITDQFVARAVVFVI-ARN
O51A4_HUMAN  SNACFAQEFFIHGFSVLESSVLLIMSFDRFLAIHNPLRYTSILTTV--RVAQIGIVFSFK
OPSD_BOVIN   PTGCNLEGFFATLGGEIALWSLVVLA-ERYVVVCKPMSNFRFGENH--AIMGVAFTWV-M
```

Figure A.1

```
             00000000000000000000000001000000001000000000100000000000000
OR4C6_HUMAN  GFMHAMI--QLLFMYQ---IP---FCGPNIIDHFICDLFQLLTLACTDTHILGLLVTLNS
OR4N4_HUMAN  GFVHSII--QVVLILR---LP---FCGPNQLDNFFCDVRQVIKLACTDMFVVELLMVFNS
O12D3_HUMAN  SFFYALM--HSVMTAH---LS---FCGSQKLNHFFYDVKPLLELACSDTLLNQWLLSIVT
O12D2_HUMAN  GFFHALL--HSVMTSR---LN---FCGSNRIHHFLCDIKPLLKLACGNTELNQWLLSTVT
O1078_RAT    SVLHALF--QSLMMLA---LP---FCTHLEIPHYFCEPNQVIQLTCSDAFLNDLVIYFTL
OR7AH_HUMAN  AALNSLS--QSLMVLW---LS---FCTDLEIPHFFCELNQVIHLACSDTFLNDMGMYFAA
OR1G1_PANTR  NALYSLL--HTLLMNS---LS---FCANHEIPHFFCDIDPLLSLSCADPFTNELVIFITG
OLFI8_RAT    TSSHAMM--HTLLAAR---LS---FCENNVLLNFFCDLFVLLKLACSDTYVNELMIHIMG
OR6J1_HUMAN  GFLSVLF--PTILISQ---LP---FCGSNIINHFFCDSGPLLALACADTTAIELMDFMLS
OL287_RAT    GFSAITV--PATLIAR---LS---FCGSRVINHFFCDISPWIVLSCTDTQVVELVSFGIA
O11A1_HUMAN  GFVVDGL--VVALVAQ---LR---FCGPNHIDQFYCDFMLFVGLACSDPRVAQVTTLILS
O11H6_HUMAN  GFLCYPV--PIVLISQ---LP---FCGPNIIDHLVCDPGPLFALACISAPSTELICYTFN
O10K1_HUMAN  GFTVSLV--TTSLVFH---LP---FHSSNQLHHFFCDISPVLKLASQHSGFSQLVIFMLG
OR2K1_HUMAN  GCLTALL--ETSFALQ---IP---LCG-NLIDHFTCEILAVLKLACTSSLLMNTIMLVVS
O13C3_HUMAN  GGINSAV--QTLLAMR---LP---FCGNNIINHFACEILAVLKLACADISLNIITMVISN
OR2F1_HUMAN  GFISSPV--QTAITFQ---LP---MCRNKFIDHISCELLAVVRLACVDTSSNEVTIMVSS
O2A12_HUMAN  SFLLALV-QHITLILR---LP---FCGPQKINHFFCQIMSVFKLACADTRLN-VVLFAGS
OR9G1_HUMAN  GFINSSI--ITKKTFS---FN---FCRENIIDDFFCDLLPLVELACGEKGGYKIMMYFLL
OR9G5_HUMAN  GFINSSI--ITKKTFS---FN---FCRENIIDDFFCDLLPLVELACGEKGGYKIMMYFLL
OLF2_CANFA   GIMDASV--NTILTFR---LC---FCESNVINHFFCDVPPLLLLSCSDTQVNELVIFTIF
O1044_MOUSE  SFLVALF--HAILTFR---LC---YCHSNAINHFYCDDMPLLRLTCSDTHSKQLWIFVCA
OR8D2_HUMAN  GFLWATV--HTTRMSV---LS---FCRSHTVSHYFCDILPLLTLSCSSTHINEILLFIIG
OL502_MOUSE  GFLNASS--FTLSFFS---LV---FCGPNRVNHFFCDFAPLVKLSCSDVSVPAVVPSFTA
OR3A3_PANTR  AFTNALT--HTIALTT---LN---FCGPSVINHFYCDLPQLFQLSCSSTQLNELLLFVAA
OR3A1_HUMAN  AFTNALT--HTVAMST---LN---FCGPNVINHFYCDLPQLFQLSCSSTQLNELLLFAVG
O10AD_HUMAN  GLINGIF--LEYISFR---EP---FRRDNHIESFFCEAPIVIGLSCGDPQFSLWAIFADA
O56A4_HUMAN  AFVSLP---VPMLSAR---LR---YCAGNIIKNCICSNLSVSKLSCDDITFNQLYQFVAG
O51A4_HUMAN  SMLLVLP---FPFTLR--NLR---YCKKNQLSHSYCLHQDVMKLACSDNRID-VIYGFFG
OPSD_BOVIN   ALACAAPP----LVGWSRYIPEGMQC--------SCGIDYYTP---HEETNNESFVIYMF
```

Figure A.1

```
             00000002000003200000000000000000000001000000000000000020000032
OR4C6_HUMAN  GMMCVAIFLILIASYT-VILCS-----LKSYSSKGRHKLLVTLNSGMMCVAIFLILIASY
OR4N4_HUMAN  GLMTLLCFLGLLASYA-VILC-----HVRRAASEGKNKLLMVFNSGLMTLLCFLGLLASY
O12D3_HUMAN  GSISMGAFFLTLLSCFYVIGFL----LFKNRSCRILHKWLLSIVTGSISMGAFFLTLLSC
O12D2_HUMAN  GTIAMGPFFLTLLSYFYIITYL----FFKTRSCSMLCKWLLSTVTGTIAMGPFFLTLLSY
O1078_RAT    VLLATVPLAGIFYSYFKIVSSI-----CAISSVHGKYKLVIYFTLVLLATVPLAGIFYSY
OR7AH_HUMAN  GLLAGGPLVGILCSYSKIVSSI-----RAISSAQGKYKMGMYFAAGLLAGGPLVGILCSY
OR1G1_PANTR  GLTGLICVLCLIISYTNVFSTI-----LKIPSAQGKRKLVIFITGGLTGLICVLCLIISY
OLFI8_RAT    VIIIVIPFVLIVISYAKIISSI-----LKVPSTQSIHKLMIHIMGVIIIVIPFVLIVISY
OR6J1_HUMAN  SMVILCCIVLVAYSYTYIILTI-----VRIPSASGRKKLMDFMLSSMVILCCIVLVAYSY
OL287_RAT    FCVILGSCGITLVSYAYIITTI-----IKIPSARGRHRLVSFGIAFCVILGSCGITLVSY
O11A1_HUMAN  VFCLTIPFGLILTSYARIVVAV-----LRVPAGASRRRVTTLILSVFCLTIPFGLILTSY
O11H6_HUMAN  SMIIFGPFLSILGSYTLVIRAV-----LCIPSGAGRTKLICYTFNSMIIFGPFLSILGSY
O10K1_HUMAN  VFALVIPLLLILVSYIRIISAI-----LKIPSSVGRYKLVIFMLGVFALVIPLLLILVSY
OR2K1_HUMAN  ILLLPIPMLLVCISYIFILSTI-----LRITSAEGRNKTIMLVVSILLLPIPMLLVCISY
O13C3_HUMAN  MAFLVLPLMVIFFSYMFILYTI-----LQMNSATGRRKITMVISNMAFLVLPLMVIFFSY
OR2F1_HUMAN  IVLLMTPFCLVLLSYIQIISTI-----LKIQSREGRKKVTIMVSSIVLLMTPFCLVLLSY
O2A12_HUMAN  AFILVGPLCLVLVSYLHILVAI-----LRIQSGEGRRKVVLFAGSAFILVGPLCLVLVSY
OR9G1_HUMAN  ASNVICPAVLILASYLFIITSV-----LRISSSKGYLKIMMYFLLASNVICPAVLILASY
OR9G5_HUMAN  ASNVICPAVLILASYLFIITSV-----LRISSSKGYLKIMMYFLLASNVICPAVLILASY
OLF2_CANFA   GFIELITLSGLFVSYCYIILAV-----RKINSAEGRFKLVIFTIFGFIELITLSGLFVSY
O1044_MOUSE  GIMFISSLLIVFISYTFIISAI-----LRMRSAEGRRKLWIFVCAGIMFISSLLIVFISY
OR8D2_HUMAN  GVNTLATTLAVLISYAFIFSSI-----LGIHSTEGQSKILLFIIGGVNTLATTLAVLISY
OL502_MOUSE  GSIIIVTIFVIAVSYIYILITI-----LKMRSTEGRQKVVPSFTAGSIIIVTIFVIAVSY
OR3A3_PANTR  AVMAVAPLVFISVSYAHVVAAV-----LQIHSAEGRKKLLLFVAAAVMAVAPLVFISVSY
OR3A1_HUMAN  FIMAGTPMALIVISYIHVAAAV-----LRIRSVEGRKKLLLFAVGFIMAGTPMALIVISY
O10AD_HUMAN  IVVILSPMVLTVTSYVHILATI-----LSKASSSGRGKWAIFADAIVVILSPMVLTVTSY
O56A4_HUMAN  WTLLGSDLILIVISYSFILKVV-----LRIKAEGAVAKLYQFVAGWTLLGSDLILIVISY
O51A4_HUMAN  ALCLMVDFILIAVSYTLILKTV-----LGIASKKEQLKVIYGFFGALCLMVDFILIAVSY
OPSD_BOVIN   VVHFIIPLIVIFFCYGQLVFTVKEAAAQQQESAT-TQKSFVIYMFVVHFIIPLIVIFFCY
```

Figure A.1

```
             00000000000000000000000000000000000000000033003200000000000000
OR4C6_HUMAN  T---MRPVVTHPI--DK-----------AMAVSDSIITPMLNPLIYTLRNAEVKSA-MKK
OR4N4_HUMAN  A---MCP--FRALPADK-----------MVSLFHTVIFPLMNPMIYTLRNQEVKTS-MKR
O12D3_HUMAN  FY--IRPASATSMIQDR-----------IMAIMYSAVTPVLNPLIYTLRNKEVMMA-LKK
O12D2_HUMAN  FY--IHPALESFMDQDR-----------IVAIMYTVVTPVLNPLIYTLRNKEVKGA-LGR
O1078_RAT    FK--LSSAANNSSQASA-----------TASVMYTVVTPMVNPFIYSLRNKDVKSV-LKK
OR7AH_HUMAN  SK--LTSAATHNSHTSA-----------TASVMYTVATPMLNPFIYSLRNKDIKRA-LKM
OR1G1_PANTR  TN--FSPPSTRXA--QK----DT-----VASVMYTVVTPMLNPFIYSLRNQEIKSS-LRK
OLFI8_RAT    AK--LCPSGDNFSLKGS-----------AMAMMYTVVTPMLNPFIYSLRNRDMKQA-LIR
OR6J1_HUMAN  TY--VTPSQKEYLEINK-----------IPLVLSSVVTPFLNPFIYTLRNDTVQGV-LRD
OL287_RAT    AY--VRTSVESSLDLTK-----------AITVLNTIVTPVLNPFIYTLRNKDVKEA-LRR
O11A1_HUMAN  AR--VAPSAVHSQLLSK-----------VFSLLYTVVTPLFNPVIYTMRNKEVHQA-LRK
O11H6_HUMAN  TL--VSPTSGNPAGMQK-----------IITLVYTAMTPFLNPLIYSLRNKDMKDA-LKR
O10K1_HUMAN  IR--LRPKTNYTSSQDT-----------LISVSYTILTPLFNPMIYSLRNKEFKSA-LRR
OR2K1_HUMAN  IF--LKPSSSNAQKIDK-----------IISLLYGVLTPMLNPIIYSLRNKEVKDA-MKK
O13C3_HUMAN  MF--AKPKSQDLIGEEKLQALDK-----LISLFYGVVTPMLNPILYSLRNKDVKAA-VKY
OR2F1_HUMAN  IQ--IQPHSSPSVLQEK-----------LFSVFYAILTPMLNPMIYSLRNKEVKGA-WQK
O2A12_HUMAN  LH--MAPKSSHSQERRK-----------ILSLFYSLFNPILNPLIYSLRNAEVKGA-LKR
OR9G1_HUMAN  LF--ALPRSSYSFDMDK-----------IVSTFYTVVFPMLNLMIYSLRNKDVKEA-LKK
OR9G5_HUMAN  LF--ALPRSSYSFDMDK-----------IVSTFYTVVFPMLNLMIYSLRNKDVKEA-LKK
OLF2_CANFA   CY--FRPSSSYSLDQDK-----------IISLFYSLVIPMLNPLIYSLRNKDVKEA-LKK
O1044_MOUSE  TF--LQPSSNHSLDTDK-----------MASVFYTVIIPMLNPLIYSLRNKEVKDA-LKK
OR8D2_HUMAN  AF--FKPPSSTTMEKEK-----------VSSVFYITIIPMLNPLIYSLRNKDVKNA-LKK
OL502_MOUSE  IY--VMPKSSYSTDQNK-----------VVSVFYMVVVPMLNPLIYSLRNKEIKGA-LKR
OR3A3_PANTR  AH--MRLGSVESSDKDK-----------GVGVFMTVINPMLNPLIYSLRNTDVQGA-LCQ
OR3A1_HUMAN  IH--MRLGSTKLSDKDK-----------AVGIFNTVINPMLNPIIYSFRNPDVQSA-IWR
O10AD_HUMAN  VH--MNPHSTHGPDKDK-----------PFSLLYTIITPMCNPIIYSFRNKEIKEA-MVR
O56A4_HUMAN  SF-------VITNLARKRIPPDV-PI--LLNILHHLIPPALNPIVYGVRTKEIKQG-IQN
O51A4_HUMAN  TLVVHRFARHVSPLINV-----------LMANVLLLVPPLTNPIVYCVKTKQIRVRVVAK
OPSD_BOVIN   GQ--AGVAFYIFTHQGS----DFGPIFMTIPAFFAKTSAVYNPVIYIMMNKQFRNCMVTT
```

Figure A.1

```
             00000000000000000000000000000000000000000000000000
OR4C6_HUMAN  LWMKWEALAGK-----------------------------------
OR4N4_HUMAN  LLS-RHVVCQVDFIIRN-----------------------------
O12D3_HUMAN  IFG-RKLFKDWQQHH-------------------------------
O12D2_HUMAN  VIR-RL------------------------------------------
O1078_RAT    TLCEEVIRSPPSLLHFFLVLCHLPCFIFCY---------------
OR7AH_HUMAN  SFRG-KQ-----------------------------------------
OR1G1_PANTR  LIW-VRKIHSP-------------------------------------
OLFI8_RAT    VTCSKKISLPW--------------------------------------
OR6J1_HUMAN  VWVRVRGVFEKRMRAVLRSRLSSNKDHQGRACSSPPCVYSVKLQC
OL287_RAT    TVKGK--------------------------------------------
O11A1_HUMAN  ILC-IKQTETLD-----------------------------------
O11H6_HUMAN  VLGLTVSQN--------------------------------------
O10K1_HUMAN  TIGQTFYPLS-------------------------------------
OR2K1_HUMAN  LLG-KITLHQTHEHL-------------------------------
O13C3_HUMAN  LLN-KKPIH---------------------------------------
OR2F1_HUMAN  LLWKFSGLTSKLAT---------------------------------
O2A12_HUMAN  VLW-KQRSM---------------------------------------
OR9G1_HUMAN  LLP---------------------------------------------
OR9G5_HUMAN  LLP---------------------------------------------
OLF2_CANFA   LKN-KKWFH---------------------------------------
O1044_MOUSE  LIA-SKNQMLSS-----------------------------------
OR8D2_HUMAN  MTRG-RQSS---------------------------------------
OL502_MOUSE  QLA-KNTFS---------------------------------------
OR3A3_PANTR  LLV-VKRSLT--------------------------------------
OR3A1_HUMAN  MLT-GRRSLA--------------------------------------
O10AD_HUMAN  ALG-RTRLAQPQSV---------------------------------
O56A4_HUMAN  LLKRL-------------------------------------------
O51A4_HUMAN  LCQRKI------------------------------------------
OPSD_BOVIN   LCCGKNPLGDDEASTTVSKTETSQVAPA-----------------
```

Figure A.1

b.

```
          0000000000000000000000000000000000000000000000000000000000
Ddis.Lip  VTYAYNSFKRYIPEGVHAVKPFYPEKYVGKWYEIARLYTYEKD--------LDKITAEY
Atha.OML  ---------MTEKKEMEVVKGLNVERYMGRWYEIASFPSRQPKN-------GVDTRATY
Hsap.ApoD -----QAFHLGKCPNPPVQENFDVNKYLGRWYEIEKIPTTENG---------RCIQANY
Dmel.NLaz ----AQVPFPGKCPDVKLLDTFDAEAYMGVWYEYAAYPFAEIG--------KKCIYANY
Dmel.GLaz --AGTDAYGFGRCPNYPSMPKFNMSRVLGHWYEVERSFYLEIASGCTTFQFEPYNKGEQ
Dmel.Karl YQTRRRSGPSNRCPKVGAIKNFDLERMMGCWHVVQYYASTELP---------EYACMRS
Same.Laz  ------AQETMGCADRSAINDFNATLYMGKWYEYAKMGSMYEE--------GGVCVTAE
Msex.IcyA ----GDIFYPGYCPEVKPVDDFDLSAFAGAWHEIAKLPLEENE--------GKCTVAEY
Rnor.RBP  -------ERDCRVSSFRVKENFDKARFSGLWYAIAKKDPELFL--------QDNIIAEF
Btau.BLB  ------------LIVTQTMKGLDIQKVAGTWYSLAMAASDISL--------LDAQSAPL
Hsap.Glyc ---------VPAMDIPQTKQDLELPKLAGTWHSMAMATNNISL--------MATLKAPL
Meug.BL   ------------VENIRSKNDLGVEKFVGSWYLREAAKT-------------MFSIPLF
Rnor.PGDS ------------QGHDTVQPNFQQDKFLGRWYSAGLASNSSW---------FREKKELL
Hsap.NGAL QDSTSDLIPAPPLSKVPLQQNFQDNQFQGKWYVVGLAGNAI----------LREDKDPQ
Mmus.A1mg -------DPASTLPDIQVQENFSESRIYGKWYNLAVGSTCWLS-------RIKDKMSVQ
Hsap.C8GC QKPQRPRRPASPISTIQPKANFDAQQFAGTWLLVAVGSACFLQ---------EQGHRAE
Rnor.ERBP ----------------AVVKDFDISKFLGFWYEIAFASKM------------TGLAHKE
Ggal.QS21 -----------------AATVPDSSEVAGKWYIVALASNTS----------FLREKGKM
Mmus.MUP1 ----------HAEEASSTGRNFNVEKINGEWHTIILASDKEKI---------EDNGNFR
Sscr.VEG  -------------AQEFPAVGQPLQDLLGRWYLKAMTSDPIP----------GKKPESV
Maur.Aphr ----------------------QDFAELQGKWYTIVIAADNEKI---------EEGGPLR
Hsap.OBP2 ------------------SFTLEEEDITGTWYVKAMVVDKFPE---------DRRPRKV
Hsap.a1G2 ------QIPLCANLVPVPITNATLDRITGKWFYIASAFRNEYN-------KSVQEIQAT
```

Figure A.1

```
          00000000000000000000000000000000000000000000222222220000000
Ddis.Lip  SINKDGSITVVNSGYNYKKKKRE-----AKGIAYFVNGSD----EGMLKVSFFG-----
Atha.OML  TLNPDGTIHVLNETWS-NGKRGF-----EGSAYKADPKSDEAKLKVKFYVPPFLPI---
Hsap.ApoD SLMENGKIKVLNQELRADGTVNQ-----EGEATPVNLTEP---AKLEVKFSWFMP----
Dmel.NLaz SLIDNSTVSVVNAAINRFTGQPS-----VTGQAKVLGPGQ-----LAVAFYPTQP---L
Dmel.GLaz SKFSNFKLAVAIKNINRITGNPNVNIGYAPENSRSSIMDF--KFTTRFPFPDVIARLLP
Dmel.Karl HFSFSKEDQHITMNFSYIFAEDPL--RKLVGNITWMIPKFQEPGHWQHTEDIYEG----
Same.Laz  YSMSSNNITVVNSMKDNTTHEVNT---TTGWAEFASEL-HT-DGKLSVHFPNSPSVG--
Msex.IcyA KYDGKKASVYNSFVINGVKEYME----GDLEIAPDAKLTK--QGKYVMTFKFGPR---V
Rnor.RBP  SVDEKGHMSATAKGRVRLLSNWE----VCADMVGTFTDTE-DPAKFKMKYWGVASFLQR
Btau.BLB  RVYVEELKPTPEGDLEILLQKWE----NGECAQKKIIAEKTKIPAVFKIDAL-------
Hsap.Glyc RVHITSLLPTPEDNLEIVLHRWE----NNSCVEKKVLGEKTGNPKKFKINYTVA-----
Meug.BL   DMDIKEVNLTPEGNLELVLLEKA-----DRCVEKKLLLKKTQKPTEFEIYISSES----
Rnor.PGDS FMCQTVVAPSTEGGLNLTSTFLR----KNQCETKVMVLQPAGVPGQYTYNSPHWG----
Hsap.NGAL KMYATIYELKEDKSYNVTSVLFR----KKKCDYWIRTFVPGCQPGEFTLGNIKSYP-GL
Mmus.A1mg TLVLQEGATETEISMTSTRWRRG----VCEEITGAYQKTD--IDGKFLYHKSKWN----
Hsap.C8GC ATTLHVAPQGTAMAVSTFRKLDG----ICWQVRQLYGDTG--VLGRFLLQARGAR----
Rnor.ERBP EKMGAMVVELKENLLALTTTYYS---DHCVLEKVTATEGDG-PAKFQVTRLSG------
Ggal.QS21 KMVMARISFLGEDELEVSYAAPS---KGCRKWETTF-KKTS--DDGELYYSEEA-----
Mmus.MUP1 LFLEQIHVLENSLVLKFHTVRDE----ECSELSMVADKTEK--AGEYSVTYDG------
Sscr.VEG  TPLILKALEGGDLEAQITFLIDG----QCQDVTLVLKKTN---QPFTFTAYDG------
Maur.Aphr FYFRHIDCYKNCSEMEITFYVIT----NNQCSKTTVIGYLK-GNGTYQTQFEG------
Hsap.OBP2 SPVKVTALGGGNLEATFTFMRED----RCIQKKILMRKTEE--PGKYSAYGGR-----K
Hsap.a1G2 FFYFTPNKTEDTIFLREYQTRQN----QCFYNSSYLNVQRE--NGTVSRYEGG-----R
```

Figure A.1

b.

```
          00002222222222222222220000000222222222000000000000000000000
Ddis.Lip  ---PFYSGYNVIAIDPDYKYAIAGQ----SFDYMWILSKEPTIPEKIKNSYLELAKSVG
Atha.OML  --IPVTGDYWVLYIDPDYQHAIGQP----SRSYLWILSRTAQMEEETYKQLVEKAVEEG
Hsap.ApoD -SAPYWILATDYENYAVYSCTCIIQ--LFHVDFAWILARNPNLPPETVDSLKNILTSNN
Dmel.NLaz TKANYLVLGTDYESYAVYSCTSVTP--LANFKIVWILTRQREPSAEAVDAARKILEDND
Dmel.GLaz GSGKYQVLYTDYENFALWSCGSIGS--LGHSDQIWILGRDRDFEVDIRSKVYDVLKRLS
Dmel.Karl -IYNTYVLDTDYDTWGVMHCAEKKK--QPRYLSALLLSRKTSLADNEISFLRGKLPQD-
Same.Laz  ---NYWILSTDYDNYSVWSCVKRP-DSAASTEISWILLRSRNSSNMTLERVEDELKNLQ
Msex.IcyA VVQVPWVLATDYKNYANYNCNYHP-DKKAHSIHAWVLSRNKVLEGNTKEVVDNVLKTFS
Rnor.RBP  GNDDHWIIDTDYDTFAQYSCRLQNLDGTCADSYSFVFSRDPNGLTPETRRLVRQRQEEL
Btau.BLB  NENKVLVLDTDYKKYLFCMENSAE---PEQSLACQCLVRTPEVDDEALEKFDKALKALP
Hsap.Glyc --NEATLLDTDYDNFLLCLQDTTTP---IQSMMCQYLARVLVEDDEIMQGFIRAFRPLP
Meug.BL   ASYTFSVMETDYDSYFFCLYNISDR----EKMACAHYVRRIE-ENKGMNEFKKILRTLA
Rnor.PGDS SFHSLSVVETDYDEYALFSKGTKGP--GQDFRMATLYSRAQLLKEELKEKFITFSKDQG
Hsap.NGAL TSYLVRVVSTNYNQHAVFFKKVSQN---REYFKITLYGRTKELTSELKENFIRFSKSLG
Mmus.A1mg ITLESYVVHTNYDEYAFLTKKSSHH--HGLTITAKLYGREPQLRDSLLQEFKDVALNVG
Hsap.C8GC GAVHVVVAETDYQSFALYLERAG-------QLSVKLYARSLPVSDSVLSGFEQRVQEAH
Rnor.ERBP -KKEVVVEATDYLTYAIDITSLVAG---AVHRTMKLYSRSLDDNGEALYNFRKITSDHG
Ggal.QS21 -EKTVEVLDTDYKSYAIFATRVKDG---RTLHMMRLYSRSREVSPTAMAIFRKLARERN
Mmus.MUP1 -FNTFTIPKTDYDNFLAHLINEKDG---ETFQLMGLYGREPDLSSDIKERFAQLCEKHG
Sscr.VEG  KRVVYILPSKVKDHYIYCEGELDG----QEVRMAKLVGRDPENNPEALEEFKEVARAKG
Maur.Aphr -NNIFQPLYITSDKIFTNKNMDRAG---QETNMIVVAGKGNALTPEENEILVQFAHEKK
Hsap.OBP2 LIYLQELPGTDDYVFYKDQRRGGL------RYMGKLVGRNPNTNNLEALEEFKKLVQHK
Hsap.a1G2 EHVAHLLFLRDTKTLMGSYLDDEKN-----WGLSFYADKPETTKEQLGEFYEALDCLCI
```

Figure A.1

```
          00000000000000000000000000000000000000
Ddis.Lip  YDITK-LIWSKQNENEN--------------------
Atha.OML  YDISK-LHKTPQSDTPPESNTAPEDSKGVWWFKSLFG
Hsap.ApoD IDVKK-MTVTDQVNCPKLS------------------
Dmel.NLaz VSQAF-LIDTVQKNCPRLDGNGTGLAGEDGLDVDDFV
Dmel.GLaz LDPER-LIISKNKQCPEAL------------------
Dmel.Karl IDTSF-MFNIGQESCDNLMESSRDDPLAYVVNGRQRA
Same.Laz  LDLNKYKTEQSAKYCA--------------------
Msex.IcyA HLIDASFMSNEFSEAACQYSTTYSLTGPDRH------
Rnor.RBP  CLERQ-RWIEHNGYCQSRPSRNSL-------------
Btau.BLB  MHIRLSNPTQLEEQCHI--------------------
Hsap.Glyc RHLWY-DLKQMEEPCRF--------------------
Meug.BL   MPYTV-IEVRTRDMCHV--------------------
Rnor.PGDS LTEED-VFLPQPDKCIQE------------------
Hsap.NGAL LPENH-VFPVPIDQCIDG------------------
Mmus.A1mg ISENS-IFMPDRGECVPGDREVEPTSIAR--------
Hsap.C8GC LTEDQ-FYFPKYGFCEAADQFHVLDEVRR--------
Rnor.ERBP FSETD-YILKHDLTCVKVLQSAAESRP----------
Ggal.QS21 YTDEM-AVLPSQEECSVDEV-----------------
Mmus.MUP1 ILREN-IDLSNANRCLQARE-----------------
Sscr.VEG  L-NPD-VRPQQSETCSPGGN-----------------
Maur.Aphr IPVEN-LNILATDTCPE--------------------
Hsap.OBP2 GLSEEDFTPLQTGSCVLEH------------------
Hsap.a1G2 PRSDV-YTDWKKDKCEPLEKQHEKERKQEEGES----
```

Figure A.1: The template multiple alignments of the 28 vertebrate olfactory receptors (plus one outgroup sequence OPSD_BOVIN; a.) and the 23 lipocalin sequences (b). On top of each alignment the invariable arrays used with indel-Seq-Gen are shown. The GPCR invariable array contains the motif patterns (shown with '1', '2', and '3') identified by Fuchs et al. [30].

## A.2 Tree file inputs to indel-Seq-Gen

a.

#fgpcr_extra.freq,b0.0454545455#[:ec1(,6,r)]{8,0.04,gpcr}
(OR4C6_HUM:7,OR4N4_HUM:9,((O12D3_HUM:6,O12D2_HUM:2):4,(((((((O1078_RAT:9,OR7AH_HUM:3):0,
OR1G1_PAN:9):3,OLFI8_RAT:8):2,((O56A4_HUM:6,O51A4_HUM:14):7,OPSD_BOV:14):5):9,(OR3A3_PAN:11,
OR3A1_HUM:4):2):11,(((((OR6J1_HUM:3,OL287_RAT:12):5,(O11A1_HUM:10,O11H6_HUM:5):7):5,(O10K1_HUM:12,
((((OR2K1_HUM:7,O13C3_HUM:7):4,OR2F1_HUM:6):5,O2A12_HUM:4):3,O10AD_HUM:9):3):3):5,(((OR9G1_HUM:0,
OR9G5_HUM:0):5,OL502_MOU:5):3,(OLF2_CAN:9,(O1044_MOU:5,OR8D2_HUM:7):3):5):3):6):2):1);

#fgpcr_tm.freq,b0.0526315789#[:tm1(,,)]{2,0.01,gpcr}
(OR4C6_HUM:10,OR4N4_HUM:6,((O12D3_HUM:3,O12D2_HUM:7):5,(((((((O1078_RAT:5,OR7AH_HUM:1):0,
OR1G1_PAN:5):1,OLFI8_RAT:5):1,((O56A4_HUM:9,O51A4_HUM:4):5,OPSD_BOV:3):10):4,(OR3A3_PAN:2,
OR3A1_HUM:2):3):2,((((OR6J1_HUM:5,OL287_RAT:5):2,(O11A1_HUM:10,O11H6_HUM:5):3):1,(O10K1_HUM:6,
((((OR2K1_HUM:4,O13C3_HUM:5):4,OR2F1_HUM:4):5,O2A12_HUM:5):2,O10AD_HUM:5):1):0):6,(((OR9G1_HUM:0,
OR9G5_HUM:0):5,OL502_MOU:6):0,(OLF2_CAN:9,(O1044_MOU:5,OR8D2_HUM:3):0):3):4):2):2):4);

#fgpcr_cyto.freq,b0.1#[:cy1(,,)]{2,0.02,gpcr}
(OR4C6_HUM:5,OR4N4_HUM:4,((O12D3_HUM:2,O12D2_HUM:4):2,(((((((O1078_RAT:3,OR7AH_HUM:1):0,
OR1G1_PAN:4):3,OLFI8_RAT:2):3,((O56A4_HUM:3,O51A4_HUM:3):3,OPSD_BOV:4):1):1,(OR3A3_PAN:0,
OR3A1_HUM:1):2):4,((((OR6J1_HUM:2,OL287_RAT:5):2,(O11A1_HUM:4,O11H6_HUM:2):2):2,(O10K1_HUM:1,
((((OR2K1_HUM:3,O13C3_HUM:4):2,OR2F1_HUM:1):1,O2A12_HUM:1):2,O10AD_HUM:7):3):0):2,(((OR9G1_HUM:0,
OR9G5_HUM:1):3,OL502_MOU:3):1,(OLF2_CAN:2,(O1044_MOU:4,OR8D2_HUM:5):0):0):3):2):1):2);

#fgpcr_tm.freq,b0.0666666667#[:tm2(,,)]{2,0.01,gpcr}
(OR4C6_HUM:6,OR4N4_HUM:5,((O12D3_HUM:1,O12D2_HUM:3):1,(((((((O1078_RAT:1,OR7AH_HUM:0):1,
OR1G1_PAN:3):2,OLFI8_RAT:3):2,((O56A4_HUM:5,O51A4_HUM:5):4,OPSD_BOV:7):5):4,(OR3A3_PAN:1,
OR3A1_HUM:0):2):3,((((OR6J1_HUM:2,OL287_RAT:3):2,(O11A1_HUM:1,O11H6_HUM:7):1):3,(O10K1_HUM:5,
((((OR2K1_HUM:3,O13C3_HUM:0):5,OR2F1_HUM:2):1,O2A12_HUM:5):2,O10AD_HUM:4):4):1):2,(((OR9G1_HUM:0,
OR9G5_HUM:2):4,OL502_MOU:5):1,(OLF2_CAN:5,(O1044_MOU:4,OR8D2_HUM:2):0):1):1):1):1):1);

#fgpcr_extra.freq,b0.05#[:ec2(,,)]{2,0.02,gpcr}
(OR4C6_HUM:9,OR4N4_HUM:5,((O12D3_HUM:6,O12D2_HUM:3):4,((((((O1078_RAT:3,OR7AH_HUM:1):5,
OR1G1_PAN:3):2,OLFI8_RAT:6):4,((O56A4_HUM:5,O51A4_HUM:7):4,OPSD_BOV:15):10):2,(OR3A3_PAN:2,
OR3A1_HUM:8):7):5,((((OR6J1_HUM:6,OL287_RAT:6):5,(O11A1_HUM:8,O11H6_HUM:4):3):0,(O10K1_HUM:5,
((((OR2K1_HUM:4,O13C3_HUM:5):5,OR2F1_HUM:8):4,O2A12_HUM:7):1,O10AD_HUM:12):3):2):0,(((OR9G1_HUM:1,
OR9G5_HUM:0):7,OL502_MOU:7):2,(OLF2_CAN:9,(O1044_MOU:6,OR8D2_HUM:4):4):4):4):1):2):3);

#fgpcr_tm.freq,b0.0555555556#[:tm3(,,)]{2,0.01,gpcr}
(OR4C6_HUM:6,OR4N4_HUM:5,((O12D3_HUM:1,O12D2_HUM:3):3,((((((O1078_RAT:2,OR7AH_HUM:7):1,
OR1G1_PAN:4):3,OLFI8_RAT:3):3,((O56A4_HUM:8,O51A4_HUM:7):7,OPSD_BOV:7):3):4,(OR3A3_PAN:1,
OR3A1_HUM:2):4):3,((((OR6J1_HUM:7,OL287_RAT:2):2,(O11A1_HUM:6,O11H6_HUM:2):1):4,(O10K1_HUM:6,
((((OR2K1_HUM:2,O13C3_HUM:6):1,OR2F1_HUM:4):2,O2A12_HUM:9):1,O10AD_HUM:8):3):1):1,(((OR9G1_HUM:2,
OR9G5_HUM:0):8,OL502_MOU:3):2,(OLF2_CAN:8,(O1044_MOU:3,OR8D2_HUM:7):1):3):3):2):3):2);

#fgpcr_cyto.freq,b0.0714285714#[:cy2(,,)]{2,0.02,gpcr}
(OR4C6_HUM:1,OR4N4_HUM:6,((O12D3_HUM:1,O12D2_HUM:2):1,((((((O1078_RAT:2,OR7AH_HUM:2):1,
OR1G1_PAN:4):0,OLFI8_RAT:3):2,((O56A4_HUM:1,O51A4_HUM:7):3,OPSD_BOV:11):4):1,(OR3A3_PAN:1,
OR3A1_HUM:3):8):0,((((OR6J1_HUM:5,OL287_RAT:2):2,(O11A1_HUM:4,O11H6_HUM:2):3):2,(O10K1_HUM:5,
((((OR2K1_HUM:1,O13C3_HUM:4):3,OR2F1_HUM:4):1,O2A12_HUM:5):1,O10AD_HUM:7):0):3):2,(((OR9G1_HUM:0,
OR9G5_HUM:0):3,OL502_MOU:6):4,(OLF2_CAN:5,(O1044_MOU:2,OR8D2_HUM:4):1):2):2):0):2):3);

#fgpcr_tm.freq,b0.0526315789#[:tm4(,,)]{2,0.01,gpcr}
(OR4C6_HUM:5,OR4N4_HUM:8,((O12D3_HUM:3,O12D2_HUM:8):2,((((((O1078_RAT:5,OR7AH_HUM:4):3,
OR1G1_PAN:3):3,OLFI8_RAT:8):5,((O56A4_HUM:7,O51A4_HUM:9):7,OPSD_BOV:7):5):1,(OR3A3_PAN:1,
OR3A1_HUM:1):5):4,((((OR6J1_HUM:5,OL287_RAT:10):3,(O11A1_HUM:7,O11H6_HUM:7):3):1,(O10K1_HUM:5,
((((OR2K1_HUM:6,O13C3_HUM:7):3,OR2F1_HUM:4):2,O2A12_HUM:7):3,O10AD_HUM:6):5):4):5,(((OR9G1_HUM:0,
OR9G5_HUM:0):4,OL502_MOU:6):3,(OLF2_CAN:5,(O1044_MOU:9,OR8D2_HUM:3):3):4):2):2):4):4);

Figure A.2

#fgpcr_extra.freq,b0.0294117647#[:ec3(,,)]{6,0.02,gpcr}
(OR4C6_HUM:12,OR4N4_HUM:13,((O12D3_HUM:8,O12D2_HUM:6):5,((((((O1078_RAT:7,OR7AH_HUM:2):11,
OR1G1_PAN:9):6,OLFI8_RAT:8):4,((O56A4_HUM:9,O51A4_HUM:15):6,OPSD_BOV:14):15):6,(OR3A3_PAN:3,
OR3A1_HUM:1):8):5,((((OR6J1_HUM:4,OL287_RAT:13):2,(O11A1_HUM:16,O11H6_HUM:7):9):5,(O10K1_HUM:16,
((((OR2K1_HUM:12,O13C3_HUM:4):2,OR2F1_HUM:15):7,O2A12_HUM:8):4,O10AD_HUM:21):2):4):4,(((OR9G1_HUM:0,
OR9G5_HUM:0):13,OL502_MOU:10):10,(OLF2_CAN:5,(O1044_MOU:10,OR8D2_HUM:13):3):4):2):3):3):9);

#fgpcr_tm.freq,b0.0454545455#[:tm5(,,)]{2,0.01,gpcr}
(OR4C6_HUM:10,OR4N4_HUM:4,((O12D3_HUM:3,O12D2_HUM:2):9,((((((O1078_RAT:7,OR7AH_HUM:5):6,
OR1G1_PAN:10):4,OLFI8_RAT:8):2,((O56A4_HUM:3,O51A4_HUM:13):6,OPSD_BOV:14):4):3,(OR3A3_PAN:6,
OR3A1_HUM:7):6):4,((((OR6J1_HUM:8,OL287_RAT:8):5,(O11A1_HUM:10,O11H6_HUM:8):6):6,(O10K1_HUM:4,
((((OR2K1_HUM:7,O13C3_HUM:11):5,OR2F1_HUM:6):7,O2A12_HUM:5):3,O10AD_HUM:11):6):3):2,(((OR9G1_HUM:0,
OR9G5_HUM:0):9,OL502_MOU:9):9,(OLF2_CAN:7,(O1044_MOU:8,OR8D2_HUM:8):6):6):3):3):7):10);

#fgpcr_cyto.freq,b0.0625#[:cy3(,,)]{6,0.02,gpcr}
(OR4C6_HUM:4,OR4N4_HUM:7,((O12D3_HUM:2,O12D2_HUM:6):9,((((((O1078_RAT:2,OR7AH_HUM:1):4,
OR1G1_PAN:3):1,OLFI8_RAT:5):4,((O56A4_HUM:4,O51A4_HUM:5):3,OPSD_BOV:8):5):4,(OR3A3_PAN:2,
OR3A1_HUM:2):3):2,((((OR6J1_HUM:2,OL287_RAT:4):3,(O11A1_HUM:6,O11H6_HUM:3):4):1,(O10K1_HUM:3,
((((OR2K1_HUM:2,O13C3_HUM:4):1,OR2F1_HUM:4):0,O2A12_HUM:3):3,O10AD_HUM:4):3):1):2,(((OR9G1_HUM:0,
OR9G5_HUM:0):5,OL502_MOU:6):4,(OLF2_CAN:6,(O1044_MOU:1,OR8D2_HUM:7):0):0):0):0):3):5);

#fgpcr_tm.freq,b0.0526315789#[:tm6(,,)]{2,0.01,gpcr}
(OR4C6_HUM:4,OR4N4_HUM:11,((O12D3_HUM:1,O12D2_HUM:2):6,((((((O1078_RAT:0,OR7AH_HUM:1):2,
OR1G1_PAN:3):4,OLFI8_RAT:1):2,((O56A4_HUM:9,O51A4_HUM:6):2,OPSD_BOV:13):5):2,(OR3A3_PAN:3,
OR3A1_HUM:1):4):2,((((OR6J1_HUM:6,OL287_RAT:4):4,(O11A1_HUM:1,O11H6_HUM:5):5):1,(O10K1_HUM:6,
((((OR2K1_HUM:4,O13C3_HUM:3):2,OR2F1_HUM:6):1,O2A12_HUM:4):2,O10AD_HUM:5):2):2):2,(((OR9G1_HUM:0,
OR9G5_HUM:0):4,OL502_MOU:2):0,(OLF2_CAN:4,(O1044_MOU:3,OR8D2_HUM:5):2):2):2):1):2):2);

#fgpcr_extra.freq,b0.0769230769#[:ec4(,,)]{6,0.03,gpcr}
(OR4C6_HUM:4,OR4N4_HUM:4,((O12D3_HUM:2,O12D2_HUM:4):5,((((((O1078_RAT:2,OR7AH_HUM:4):4,
OR1G1_PAN:4):2,OLFI8_RAT:7):4,((O56A4_HUM:3,O51A4_HUM:4):5,OPSD_BOV:7):4):4,(OR3A3_PAN:1,
OR3A1_HUM:2):3):6,((((OR6J1_HUM:5,OL287_RAT:4):3,(O11A1_HUM:4,O11H6_HUM:6):4):4,(O10K1_HUM:7,
((((OR2K1_HUM:6,O13C3_HUM:3):2,OR2F1_HUM:6):4,O2A12_HUM:2):2,O10AD_HUM:6):2):1):1,(((OR9G1_HUM:0,
OR9G5_HUM:0):3,OL502_MOU:3):2,(OLF2_CAN:0,(O1044_MOU:4,OR8D2_HUM:6):3):2):1):5):0):2);

#fgpcr_tm.freq,b0.0588235294#[:tm7(,,)]{2,0.01,gpcr}
(OR4C6_HUM:3,OR4N4_HUM:9,((O12D3_HUM:2,O12D2_HUM:1):3,((((((O1078_RAT:1,OR7AH_HUM:1):0,
OR1G1_PAN:1):2,OLFI8_RAT:2):2,((O56A4_HUM:4,O51A4_HUM:6):8,OPSD_BOV:9):8):1,(OR3A3_PAN:2,
OR3A1_HUM:3):5):2,((((OR6J1_HUM:4,OL287_RAT:3):2,(O11A1_HUM:7,O11H6_HUM:4):1):3,(O10K1_HUM:4,
((((OR2K1_HUM:2,O13C3_HUM:2):3,OR2F1_HUM:2):0,O2A12_HUM:8):2,O10AD_HUM:4):2):4):1,(((OR9G1_HUM:0,
OR9G5_HUM:0):3,OL502_MOU:3):1,(OLF2_CAN:3,(O1044_MOU:0,OR8D2_HUM:4):3):1):1):2):1):3);

#fgpcr_cyto.freq,b0.0454545455#[:cy4(,6,r)]{8,0.04,gpcr}
(OR4C6_HUM:7,OR4N4_HUM:16,((O12D3_HUM:2,O12D2_HUM:6):4,((((((O1078_RAT:5,OR7AH_HUM:5):3,
OR1G1_PAN:10):5,OLFI8_RAT:7):6,((O56A4_HUM:7,O51A4_HUM:4):5,OPSD_BOV:6):23):3,(OR3A3_PAN:3,
OR3A1_HUM:7):7):6,((((OR6J1_HUM:10,OL287_RAT:1):8,(O11A1_HUM:7,O11H6_HUM:6):4):4,(O10K1_HUM:10,
((((OR2K1_HUM:3,O13C3_HUM:5):5,OR2F1_HUM:8):3,O2A12_HUM:5):3,O10AD_HUM:9):3):6):5,(((OR9G1_HUM:0,
OR9G5_HUM:0):1,OL502_MOU:5):2,(OLF2_CAN:4,(O1044_MOU:3,OR8D2_HUM:6):3):1):4):15):5):7);

Figure A.2

b.

```
"nterm"#b0.04545,fcoil.freq#[:nterm(,,)]{4,0.04,lipo}
(Ddis.Lip:17,Atha.OML:11,(Hsap.ApoD:11,((Dmel.NLaz:8,((Dmel.GLaz:12,Same.Laz:6):8,Msex.IcyA:9):2):4,
(Dmel.Karl:11,(Rnor.RBP:7,((((((Btau.BLB:7,Hsap.Glyc:4):5,Meug.BL:7):3,((Rnor.PGDS:6,Hsap.NGAL:5):8,
(Mmus.MUP1:2,Maur.Aphr:5):6):5):8,((Mmus.A1mg:9,Hsap.C8GC:5):7,((Sscr.VEG:5,Hsap.OBP2:4):5,
Hsap.a1G2:6):9):6):2,Ggal.QS21:6):3,Rnor.ERBP:4):6):8):11):3):5);

"beta1"#b0.125,fbeta.freq#[:beta1(,,)]{0,0}
(Ddis.Lip:3,Atha.OML:2,(Hsap.ApoD:2,((Dmel.NLaz:2,((Dmel.GLaz:5,Same.Laz:1):2,Msex.IcyA:3):1):1,
(Dmel.Karl:6,(Rnor.RBP:3,((((((Btau.BLB:0,Hsap.Glyc:3):1,Meug.BL:5):3,((Rnor.PGDS:2,Hsap.NGAL:2):0,
(Mmus.MUP1:2,Maur.Aphr:3):3):2):1,((Mmus.A1mg:2,Hsap.C8GC:2):2,((Sscr.VEG:0,Hsap.OBP2:4):3,
Hsap.a1G2:4):1):0):2,Ggal.QS21:0):3,Rnor.ERBP:1):2):2):1):1):1);

"in1"#b0.047619,fcoil.freq#[:in1(,,)]{2,0.02/0.01,lipo}
(Ddis.Lip:9,Atha.OML:13,(Hsap.ApoD:7,((Dmel.NLaz:11,((Dmel.GLaz:7,Same.Laz:12):6,Msex.IcyA:11):3):6,
(Dmel.Karl:10,(Rnor.RBP:14,((((((Btau.BLB:8,Hsap.Glyc:5):3,Meug.BL:8):2,((Rnor.PGDS:10,Hsap.NGAL:4):7,
(Mmus.MUP1:8,Maur.Aphr:6):7):7):6,((Mmus.A1mg:10,Hsap.C8GC:5):8,((Sscr.VEG:4,Hsap.OBP2:8):3,
Hsap.a1G2:13):7):4):6,Ggal.QS21:8):5,Rnor.ERBP:9):3):7):8):4):6);

"beta2"#b0.1,fbeta.freq#[:beta2(,,)]{0,0}
(Ddis.Lip:3,Atha.OML:6,(Hsap.ApoD:3,((Dmel.NLaz:4,((Dmel.GLaz:6,Same.Laz:4):4,Msex.IcyA:4):1):2,
(Dmel.Karl:10,(Rnor.RBP:5,((((((Btau.BLB:2,Hsap.Glyc:3):4,Meug.BL:6):1,((Rnor.PGDS:2,Hsap.NGAL:4):2,
(Mmus.MUP1:6,Maur.Aphr:1):5):3):3,((Mmus.A1mg:6,Hsap.C8GC:4):2,((Sscr.VEG:4,Hsap.OBP2:3):5,
Hsap.a1G2:5):2):5):3,Ggal.QS21:4):3,Rnor.ERBP:5):4):0):2):1):1);

"in2"#b0.047619,fcoil.freq#[:in2(,,)]{2,0.02/0.01,lipo}
(Ddis.Lip:9,Atha.OML:13,(Hsap.ApoD:7,((Dmel.NLaz:11,((Dmel.GLaz:7,Same.Laz:12):6,Msex.IcyA:11):3):6,
(Dmel.Karl:10,(Rnor.RBP:14,((((((Btau.BLB:8,Hsap.Glyc:5):3,Meug.BL:8):2,((Rnor.PGDS:10,Hsap.NGAL:4):7,
(Mmus.MUP1:8,Maur.Aphr:6):7):7):6,((Mmus.A1mg:10,Hsap.C8GC:5):8,((Sscr.VEG:4,Hsap.OBP2:8):3,
Hsap.a1G2:13):7):4):6,Ggal.QS21:8):5,Rnor.ERBP:9):3):7):8):4):6);

"beta3"#b0.1111,fbeta.freq#[:beta3(,,)]{0,0}
(Ddis.Lip:7,Atha.OML:5,(Hsap.ApoD:3,((Dmel.NLaz:4,((Dmel.GLaz:3,Same.Laz:6):0,Msex.IcyA:9):0):3,
(Dmel.Karl:7,(Rnor.RBP:5,((((((Btau.BLB:2,Hsap.Glyc:1):1,Meug.BL:1):5,((Rnor.PGDS:2,Hsap.NGAL:6):4,
(Mmus.MUP1:7,Maur.Aphr:3):3):0):2,((Mmus.A1mg:1,Hsap.C8GC:8):4,((Sscr.VEG:1,Hsap.OBP2:3):5,
Hsap.a1G2:6):2):2):2,Ggal.QS21:5):1,Rnor.ERBP:6):5):3):3):5):1);

"in3"#b0.047619,fcoil.freq#[:in3(,,)]{2,0.02/0.01,lipo}
(Ddis.Lip:9,Atha.OML:13,(Hsap.ApoD:7,((Dmel.NLaz:11,((Dmel.GLaz:7,Same.Laz:12):6,Msex.IcyA:11):3):6,
(Dmel.Karl:10,(Rnor.RBP:14,((((((Btau.BLB:8,Hsap.Glyc:5):3,Meug.BL:8):2,((Rnor.PGDS:10,Hsap.NGAL:4):7,
(Mmus.MUP1:8,Maur.Aphr:6):7):7):6,((Mmus.A1mg:10,Hsap.C8GC:5):8,((Sscr.VEG:4,Hsap.OBP2:8):3,
Hsap.a1G2:13):7):4):6,Ggal.QS21:8):5,Rnor.ERBP:9):3):7):8):4):6);

"beta4"#b0.1111,fbeta.freq#[:beta4(,,)]{0,0} (Ddis.Lip:6,Atha.OML:5,(Hsap.ApoD:5,((Dmel.NLaz:7,((Dmel.GLaz:5,Same.La
(Dmel.Karl:5,(Rnor.RBP:4,((((((Btau.BLB:5,Hsap.Glyc:1):0,Meug.BL:4):4,((Rnor.PGDS:3,Hsap.NGAL:6):4,
(Mmus.MUP1:5,Maur.Aphr:6):4):3):3,((Mmus.A1mg:4,Hsap.C8GC:5):2,((Sscr.VEG:5,Hsap.OBP2:4):1,
Hsap.a1G2:9):3):3):2,Ggal.QS21:7):3,Rnor.ERBP:6):1):5):4):2):6);

"beta5678"#b0.02,fbeta.freq#[:beta5678(,,)]{1,0.01/0.0,lipo}
(Ddis.Lip:13,Atha.OML:21,(Hsap.ApoD:11,((Dmel.NLaz:16,((Dmel.GLaz:16,Same.Laz:18):9,Msex.IcyA:22):8):8,
(Dmel.Karl:27,(Rnor.RBP:20,((((((Btau.BLB:9,Hsap.Glyc:18):14,Meug.BL:17):13,((Rnor.PGDS:13,
Hsap.NGAL:25):16,(Mmus.MUP1:12,Maur.Aphr:26):14):12):12,((Mmus.A1mg:20,Hsap.C8GC:15):13,((Sscr.VEG:19,
Hsap.OBP2:17):8,Hsap.a1G2:29):19):11):12,Ggal.QS21:13):12,Rnor.ERBP:13):15):9):19):8):23);
```

Figure A.2

```
"in4"#b0.047619,fcoil.freq#[:in4(,,)]{2,0.02/0.01,lipo}
(Ddis.Lip:9,Atha.OML:13,(Hsap.ApoD:7,((Dmel.NLaz:11,((Dmel.GLaz:7,Same.Laz:12):6,Msex.IcyA:11):3):6,
(Dmel.Karl:10,(Rnor.RBP:14,((((((Btau.BLB:8,Hsap.Glyc:5):3,Meug.BL:8):2,((Rnor.PGDS:10,Hsap.NGAL:4):7,
(Mmus.MUP1:8,Maur.Aphr:6):7):7):6,((Mmus.A1mg:10,Hsap.C8GC:5):8,((Sscr.VEG:4,Hsap.OBP2:8):3,
Hsap.a1G2:13):7):4):6,Ggal.QS21:8):5,Rnor.ERBP:9):3):7):8):4):6);

"alpha1"#b0.076923,falpha.freq#[:alpha1(,,)]{0,0}
(Ddis.Lip:7,Atha.OML:6,(Hsap.ApoD:5,((Dmel.NLaz:5,((Dmel.GLaz:6,Same.Laz:3):2,Msex.IcyA:5):7):2,
(Dmel.Karl:7,(Rnor.RBP:9,((((((Btau.BLB:4,Hsap.Glyc:5):1,Meug.BL:7):2,((Rnor.PGDS:2,Hsap.NGAL:2):5,
(Mmus.MUP1:5,Maur.Aphr:6):2):5):1,((Mmus.A1mg:2,Hsap.C8GC:7):4,((Sscr.VEG:3,Hsap.OBP2:9):3,
Hsap.a1G2:9):2):4):2,Ggal.QS21:5):1,Rnor.ERBP:6):3):2):3):3):4);

"cterm"#b0.037037,fcoil.freq#[:cterm(,,)]{4,0.04,lipo}
(Ddis.Lip:8,Atha.OML:20,(Hsap.ApoD:8,((Dmel.NLaz:12,((Dmel.GLaz:5,Same.Laz:7):7,Msex.IcyA:12):13):14,
(Dmel.Karl:16,(Rnor.RBP:14,((((((Btau.BLB:4,Hsap.Glyc:7):7,Meug.BL:7):9,((Rnor.PGDS:6,Hsap.NGAL:5):3,
(Mmus.MUP1:4,Maur.Aphr:10):9):3):8,((Mmus.A1mg:13,Hsap.C8GC:11):7,((Sscr.VEG:9,Hsap.OBP2:7):7,
Hsap.a1G2:10):11):5):3,Ggal.QS21:10):8,Rnor.ERBP:11):6):11):13):3):5);
```

Figure A.2: The input file that gives the specification of each subsequence of the (a) vertebrate olfactory-receptor and (b) lipocalin template multiple alignments. For a detailed description of each of the options.

# Appendix B

# Gap Profiling Supplementary Material

This appendix holds all relevant figures and tables for comparisons made in Chapter 4.

## B.1 Scoring Methods Colored by Benchmark Parameter Values

This section compares scoring metric return values versus four of the statistics from Table 4.3: Contiguous block length, contiguous indel length, average normalized hamming distance (ANHD), and percent gappiness. The first figure compares each metric/statistic pair with five plots, where each plot colors the data points according to the parameter value used to create the benchmark datasets corresponding to the data point. The second figure continues to use the above statistics, but compares two scoring metrics to one another by plotting the values of both methods in the same plot.

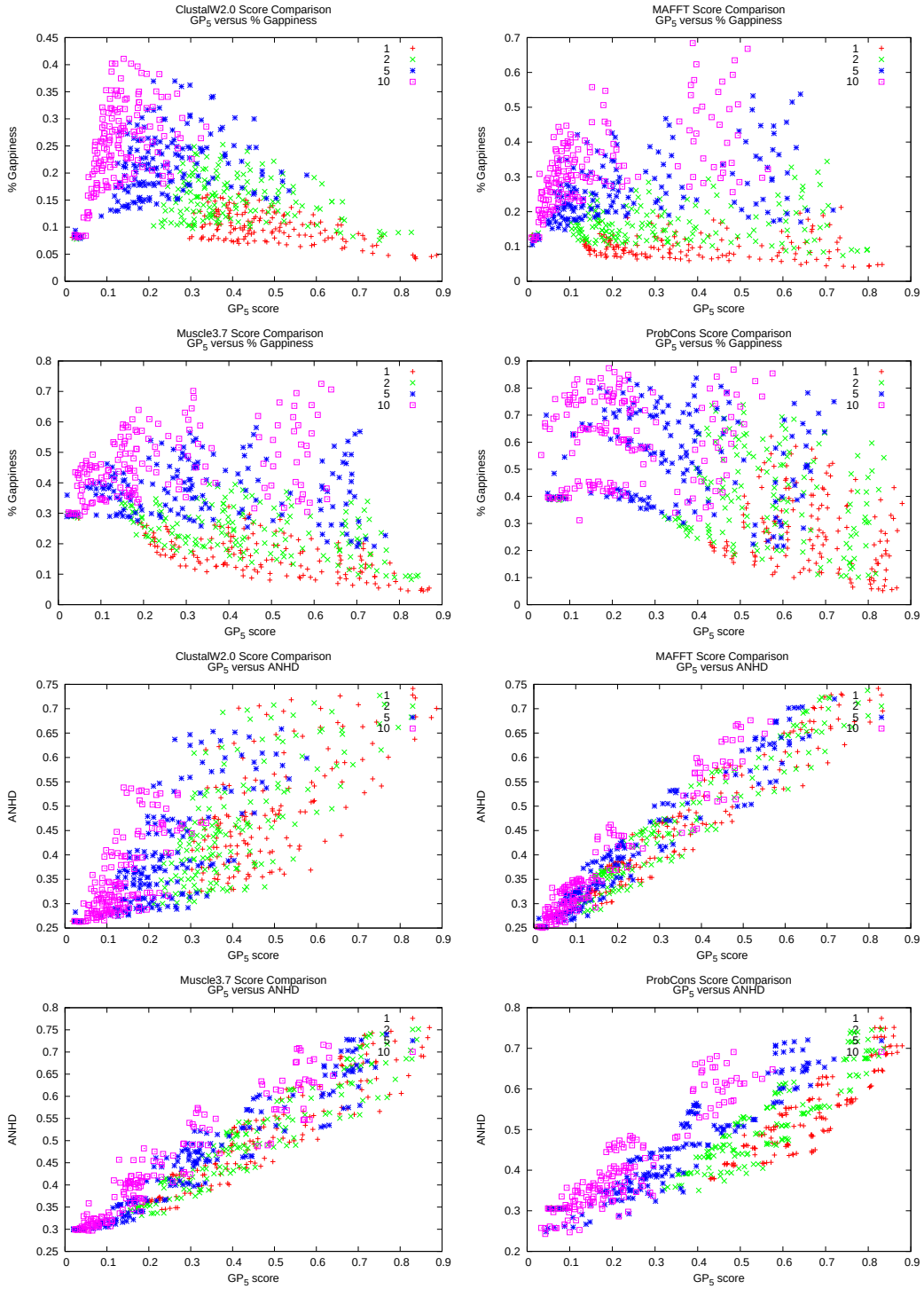

ClustalW2.0 Score Comparison
GP5 versus % Gappiness
MAFFT Score Comparison
GP5 versus % Gappiness
Muscle3.7 Score Comparison
GP5 versus % Gappiness
ProbCons Score Comparison
GP5 versus % Gappiness
ClustalW2.0 Score Comparison
GP5 versus ANHD
MAFFT Score Comparison
GP5 versus ANHD
Muscle3.7 Score Comparison
GP5 versus ANHD
ProbCons Score Comparison
GP5 versus ANHD
% Gappiness
ANHD
GP5 score
1
2
5
10

Figure B.1

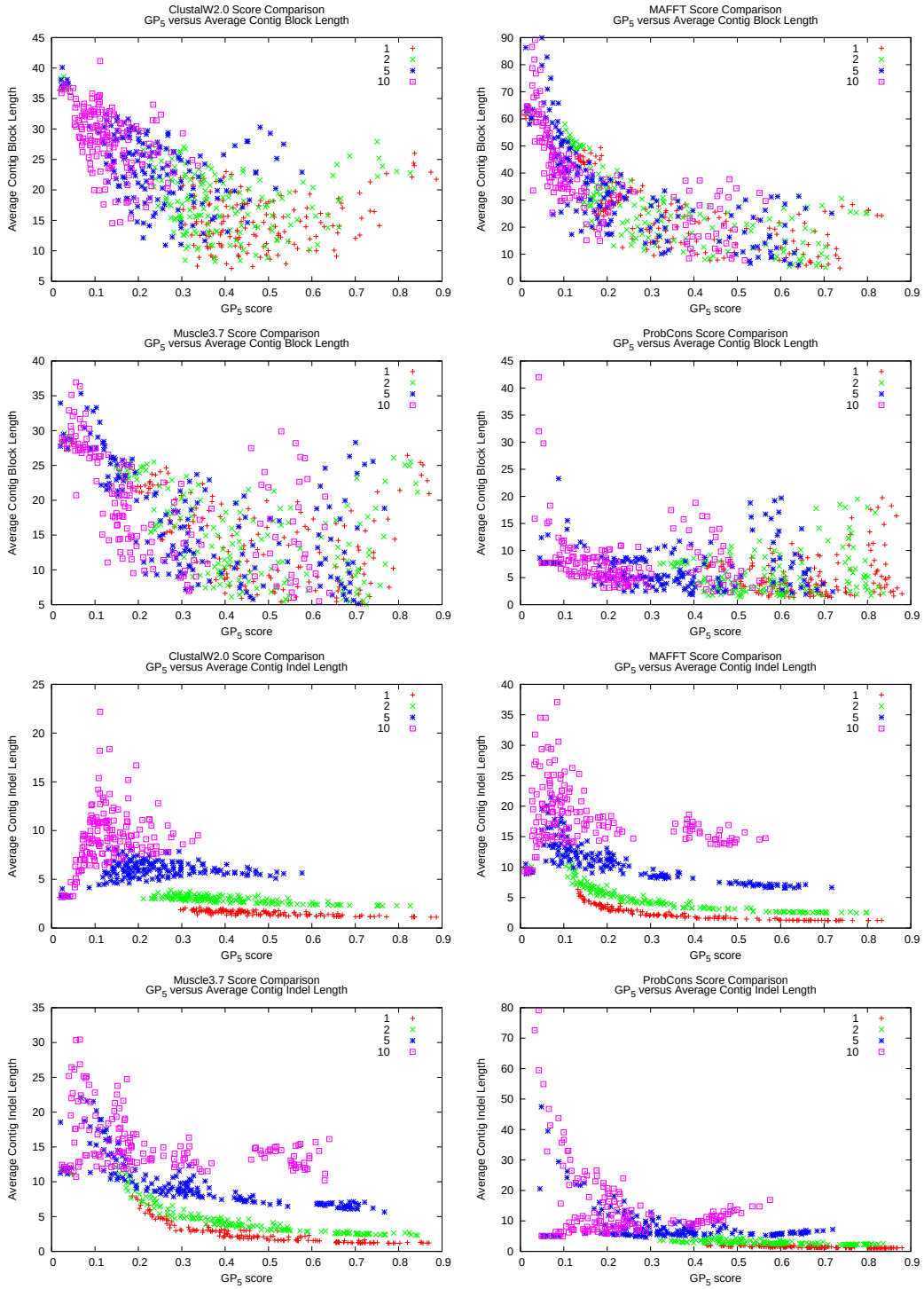

ClustalW2.0 Score Comparison
GP5 versus Average Contig Block Length
MAFFT Score Comparison
GP5 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Block Length
ProbCons Score Comparison
GP5 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP5 versus Average Contig Indel Length
MAFFT Score Comparison
GP5 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Indel Length
ProbCons Score Comparison
GP5 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP5 score
1
2
5
10

Figure B.1

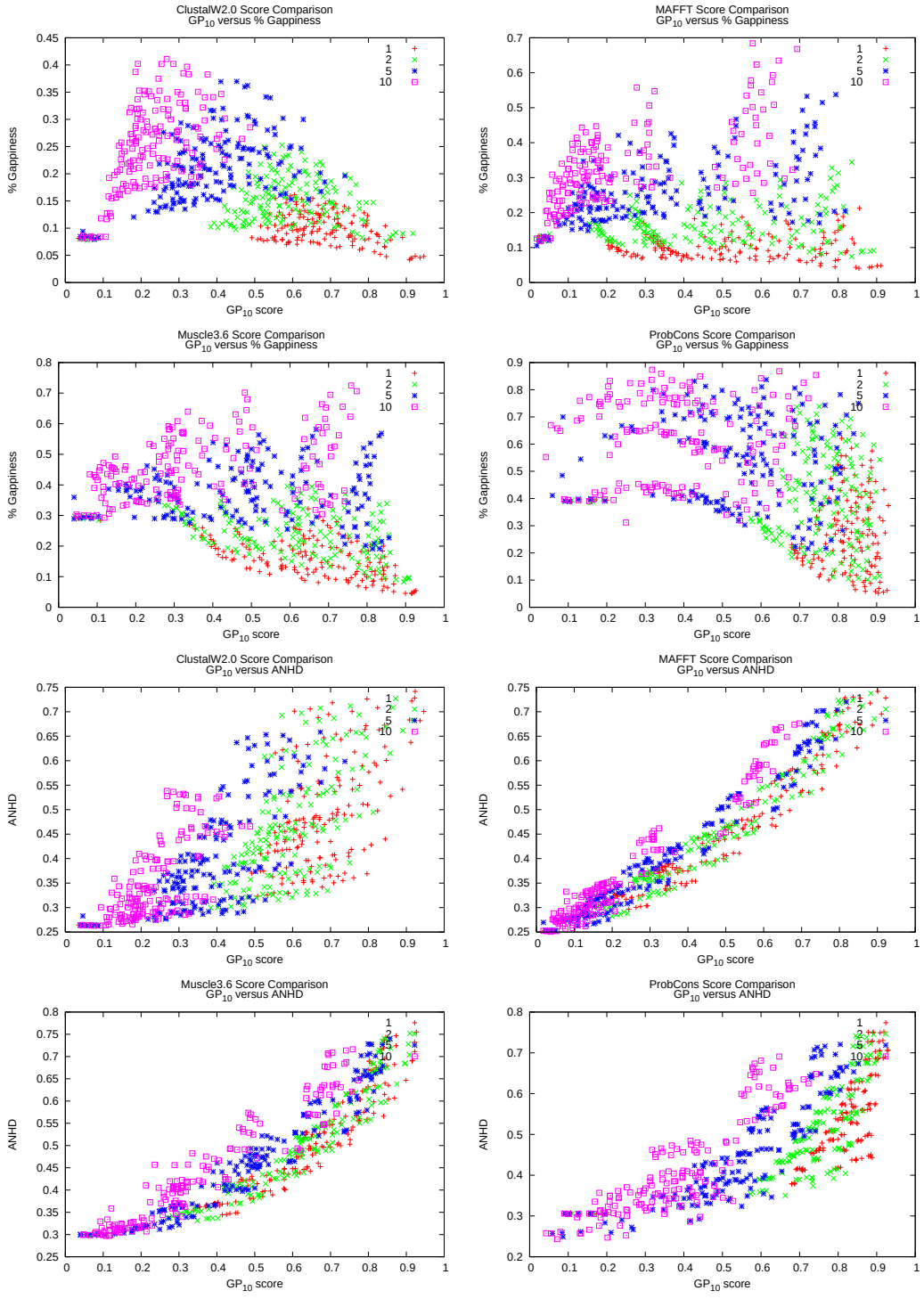

ClustalW2.0 Score Comparison
GP10 versus % Gappiness
MAFFT Score Comparison
GP10 versus % Gappiness
Muscle3.6 Score Comparison
GP10 versus % Gappiness
ProbCons Score Comparison
GP10 versus % Gappiness
ClustalW2.0 Score Comparison
GP10 versus ANHD
MAFFT Score Comparison
GP10 versus ANHD
Muscle3.6 Score Comparison
GP10 versus ANHD
ProbCons Score Comparison
GP10 versus ANHD
% Gappiness
ANHD
GP10 score
1
2
5
10

Figure B.1

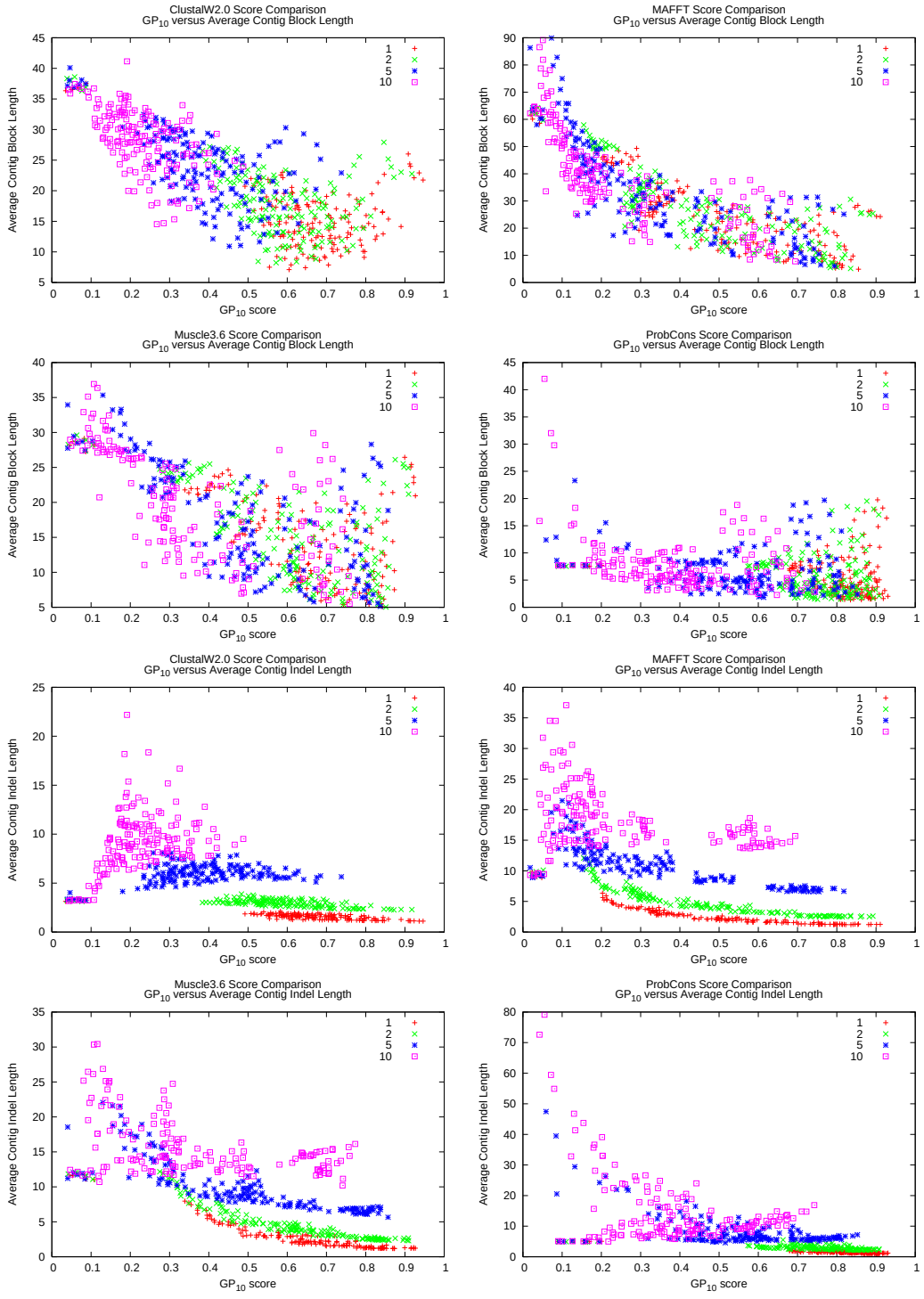

ClustalW2.0 Score Comparison
GP10 versus Average Contig Block Length
MAFFT Score Comparison
GP10 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Block Length
ProbCons Score Comparison
GP10 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP10 versus Average Contig Indel Length
MAFFT Score Comparison
GP10 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Indel Length
ProbCons Score Comparison
GP10 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP10 score
1
2
5
10

Figure B.1

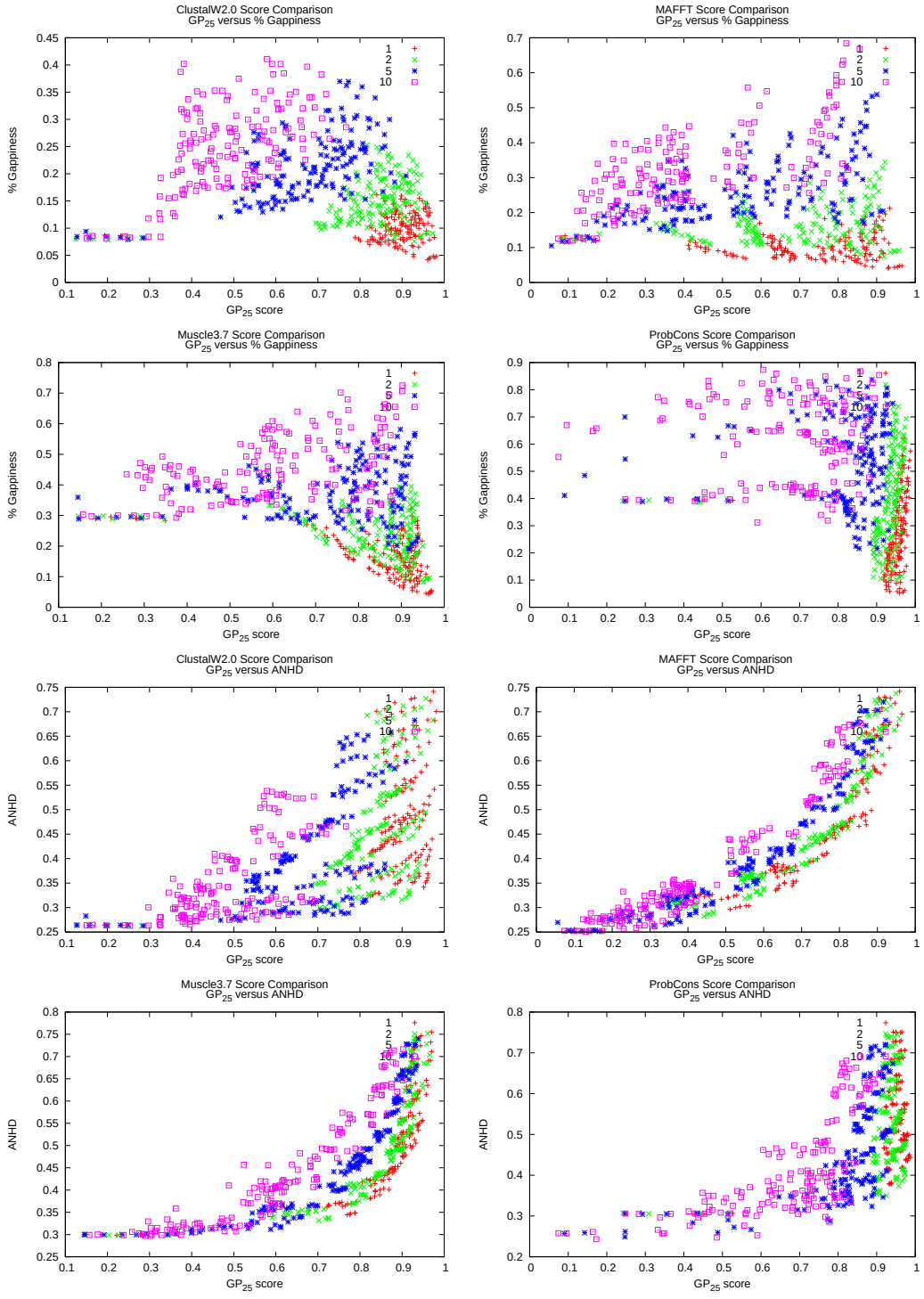

ClustalW2.0 Score Comparison
GP25 versus % Gappiness
MAFFT Score Comparison
GP25 versus % Gappiness
Muscle3.7 Score Comparison
GP25 versus % Gappiness
ProbCons Score Comparison
GP25 versus % Gappiness
ClustalW2.0 Score Comparison
GP25 versus ANHD
MAFFT Score Comparison
GP25 versus ANHD
Muscle3.7 Score Comparison
GP25 versus ANHD
ProbCons Score Comparison
GP25 versus ANHD
% Gappiness
ANHD
GP25 score
1
2
5
10

Figure B.1

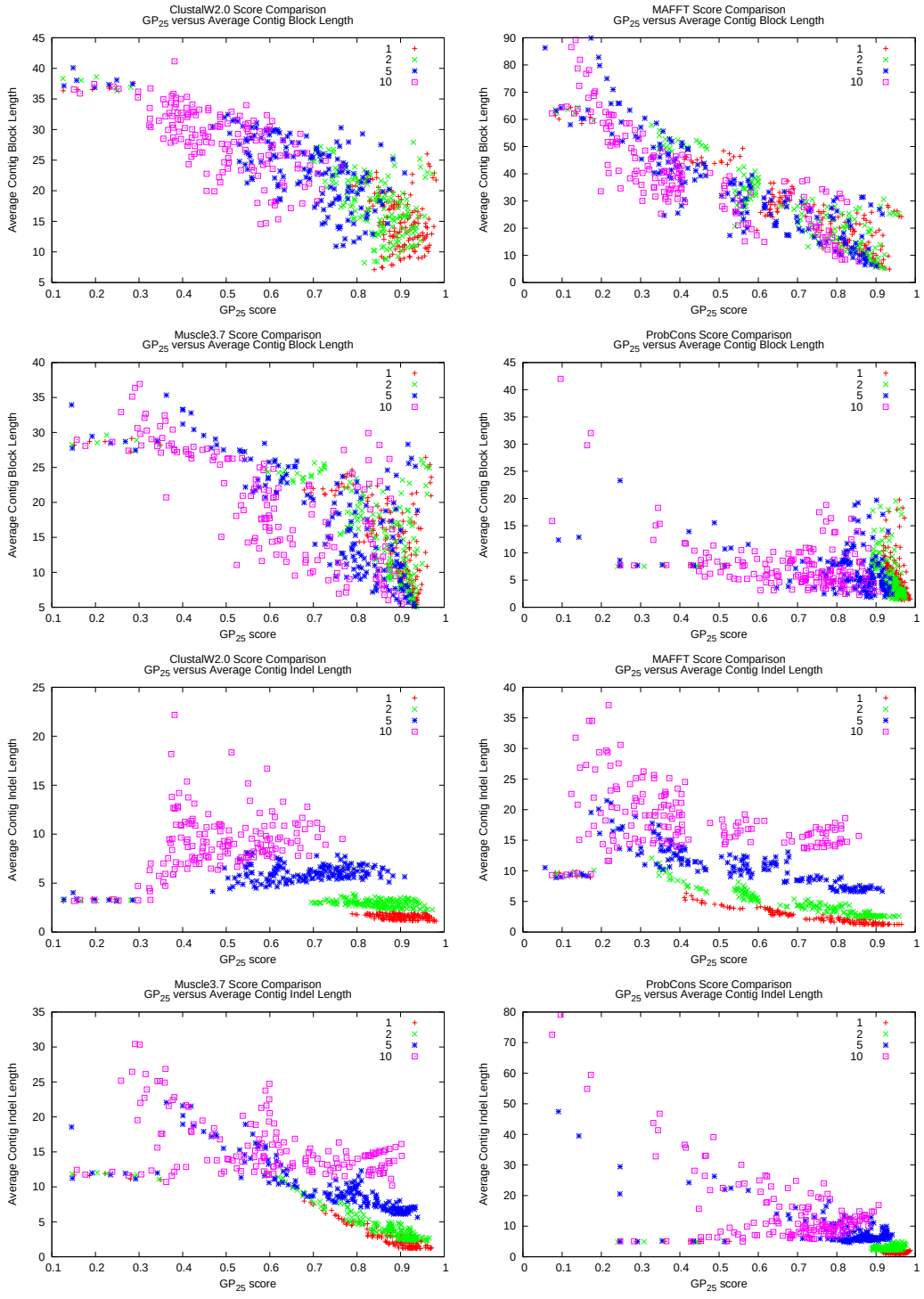

ClustalW2.0 Score Comparison
GP25 versus Average Contig Block Length
MAFFT Score Comparison
GP25 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Block Length
ProbCons Score Comparison
GP25 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP25 versus Average Contig Indel Length
MAFFT Score Comparison
GP25 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Indel Length
ProbCons Score Comparison
GP25 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP25 score
1
2
5
10

Figure B.1

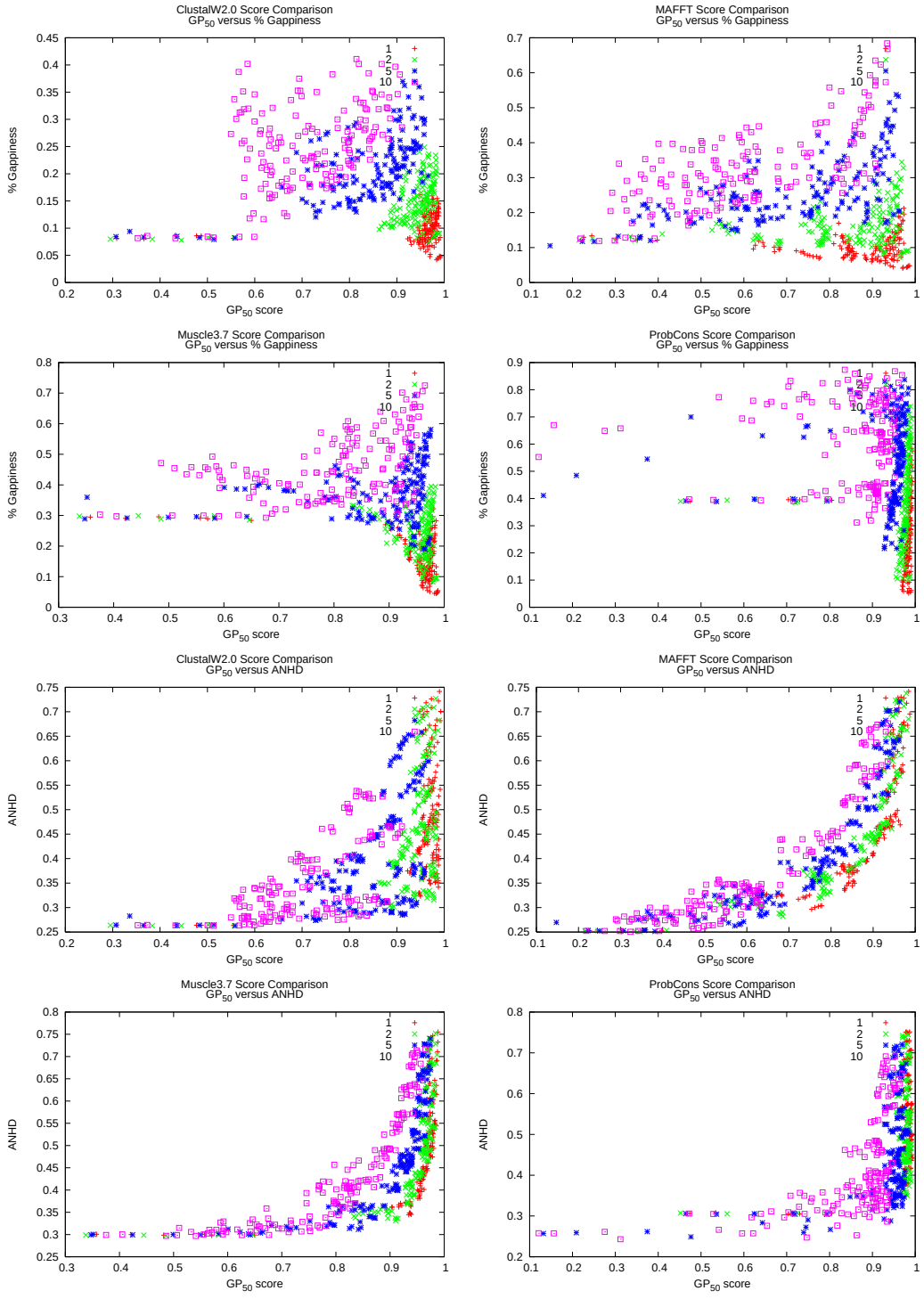

ClustalW2.0 Score Comparison
GP50 versus % Gappiness
MAFFT Score Comparison
GP50 versus % Gappiness
Muscle3.7 Score Comparison
GP50 versus % Gappiness
ProbCons Score Comparison
GP50 versus % Gappiness
ClustalW2.0 Score Comparison
GP50 versus ANHD
MAFFT Score Comparison
GP50 versus ANHD
Muscle3.7 Score Comparison
GP50 versus ANHD
ProbCons Score Comparison
GP50 versus ANHD
% Gappiness
ANHD
GP50 score
1
2
5
10

Figure B.1

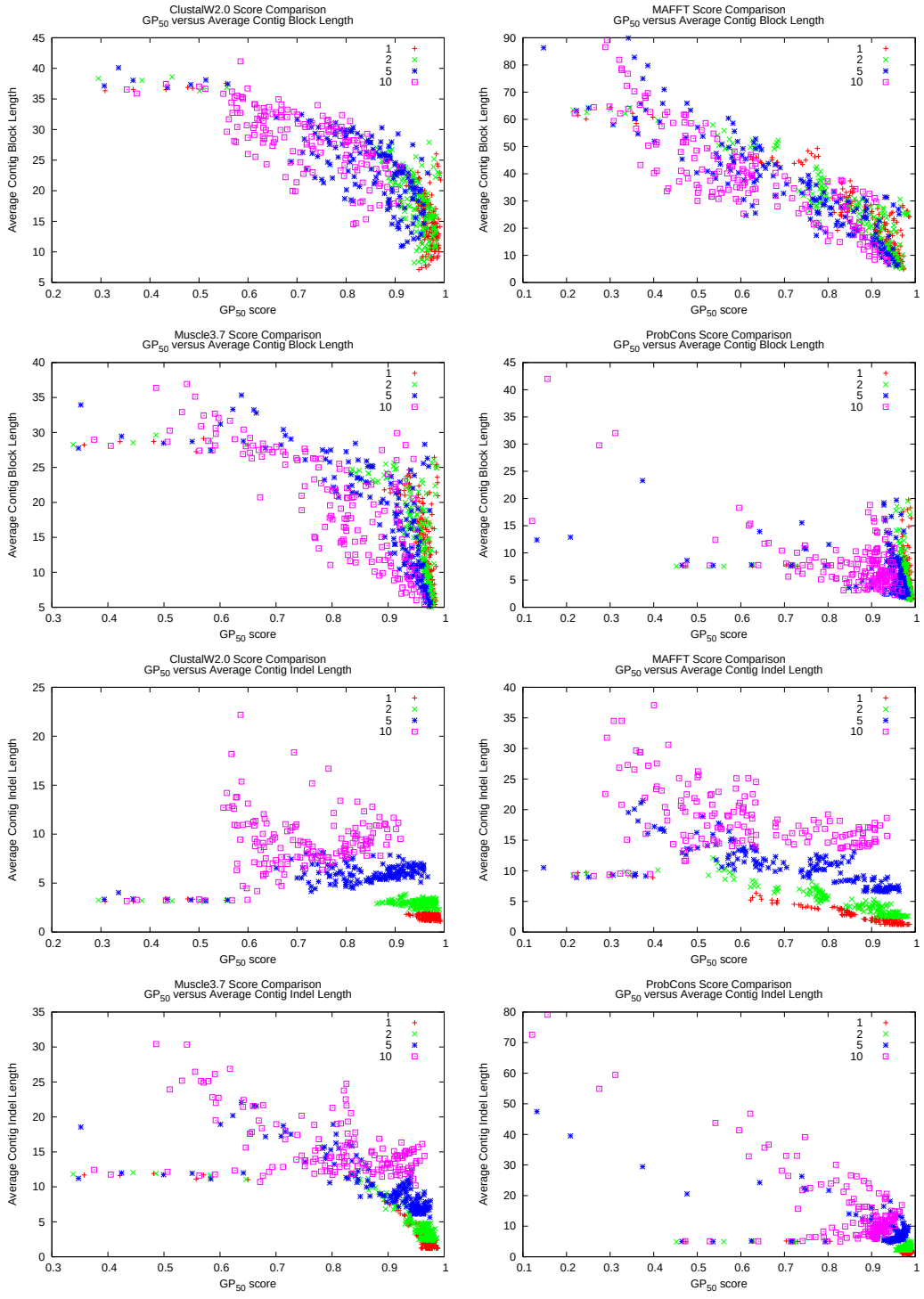

ClustalW2.0 Score Comparison
GP50 versus Average Contig Block Length
MAFFT Score Comparison
GP50 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Block Length
ProbCons Score Comparison
GP50 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP50 versus Average Contig Indel Length
MAFFT Score Comparison
GP50 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Indel Length
ProbCons Score Comparison
GP50 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP50 score
1
2
5
10

Figure B.1

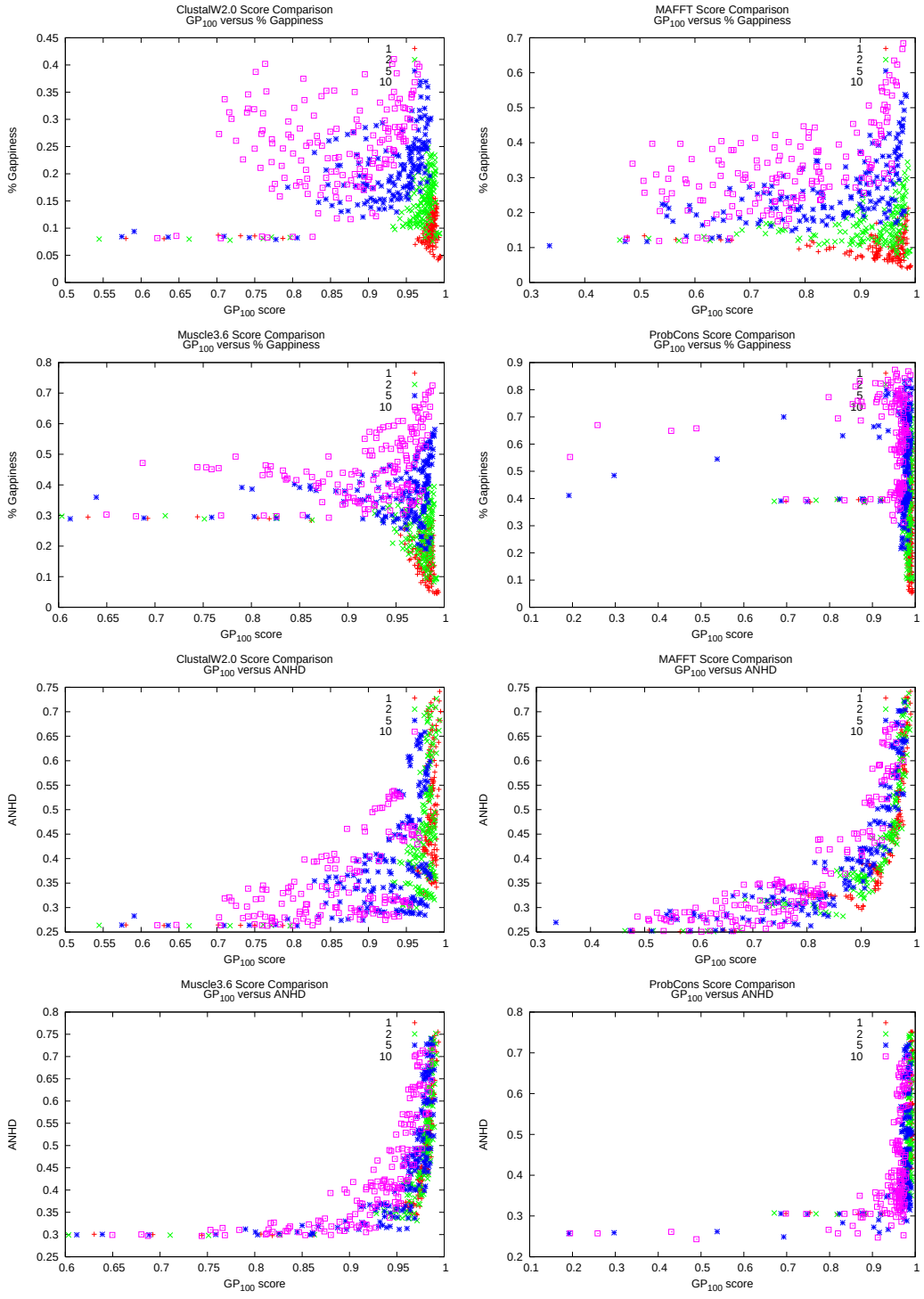

ClustalW2.0 Score Comparison
GP100 versus % Gappiness
% Gappiness
GP100 score
MAFFT Score Comparison
GP100 versus % Gappiness
% Gappiness
GP100 score
Muscle3.6 Score Comparison
GP100 versus % Gappiness
% Gappiness
GP100 score
ProbCons Score Comparison
GP100 versus % Gappiness
% Gappiness
GP100 score
ClustalW2.0 Score Comparison
GP100 versus ANHD
ANHD
GP100 score
MAFFT Score Comparison
GP100 versus ANHD
ANHD
GP100 score
Muscle3.6 Score Comparison
GP100 versus ANHD
ANHD
GP100 score
ProbCons Score Comparison
GP100 versus ANHD
ANHD
GP100 score
1
2
5
10

Figure B.1

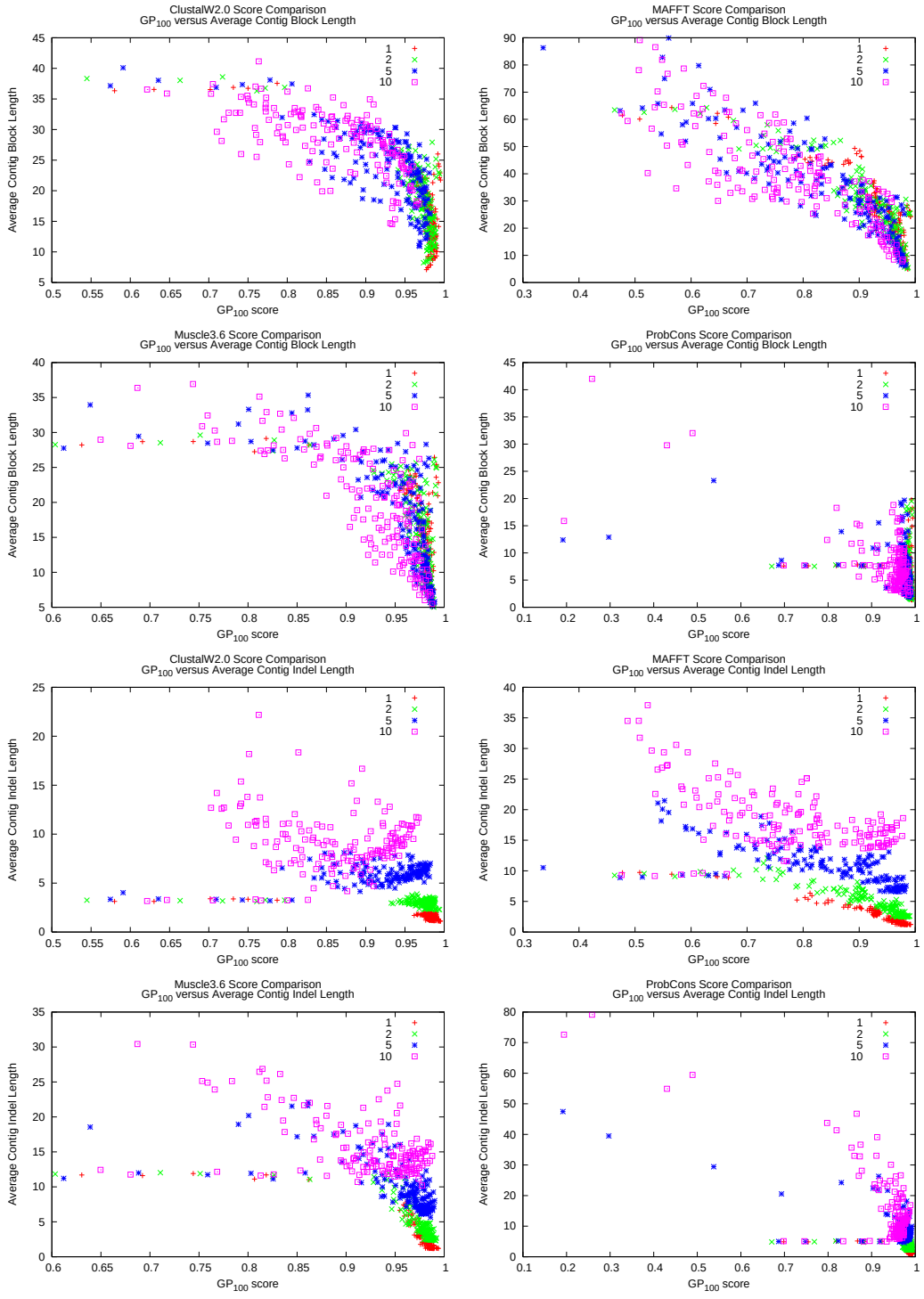

ClustalW2.0 Score Comparison
GP100 versus Average Contig Block Length
MAFFT Score Comparison
GP100 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Block Length
ProbCons Score Comparison
GP100 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP100 versus Average Contig Indel Length
MAFFT Score Comparison
GP100 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Indel Length
ProbCons Score Comparison
GP100 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP100 score
1
2
5
10

Figure B.1

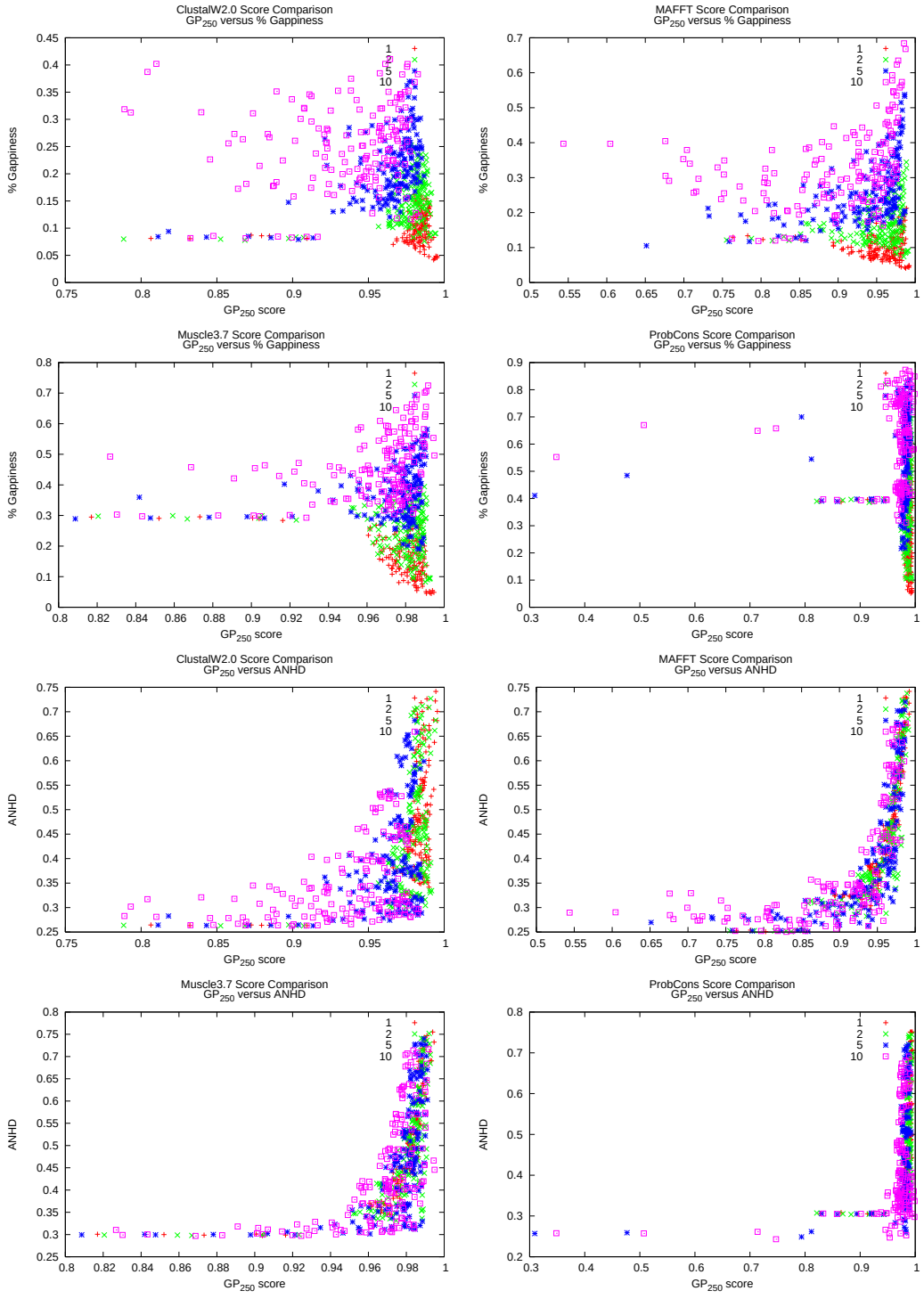

ClustalW2.0 Score Comparison
GP250 versus % Gappiness
MAFFT Score Comparison
GP250 versus % Gappiness
Muscle3.7 Score Comparison
GP250 versus % Gappiness
ProbCons Score Comparison
GP250 versus % Gappiness
ClustalW2.0 Score Comparison
GP250 versus ANHD
MAFFT Score Comparison
GP250 versus ANHD
Muscle3.7 Score Comparison
GP250 versus ANHD
ProbCons Score Comparison
GP250 versus ANHD
% Gappiness
ANHD
GP250 score
1
2
5
10

Figure B.1

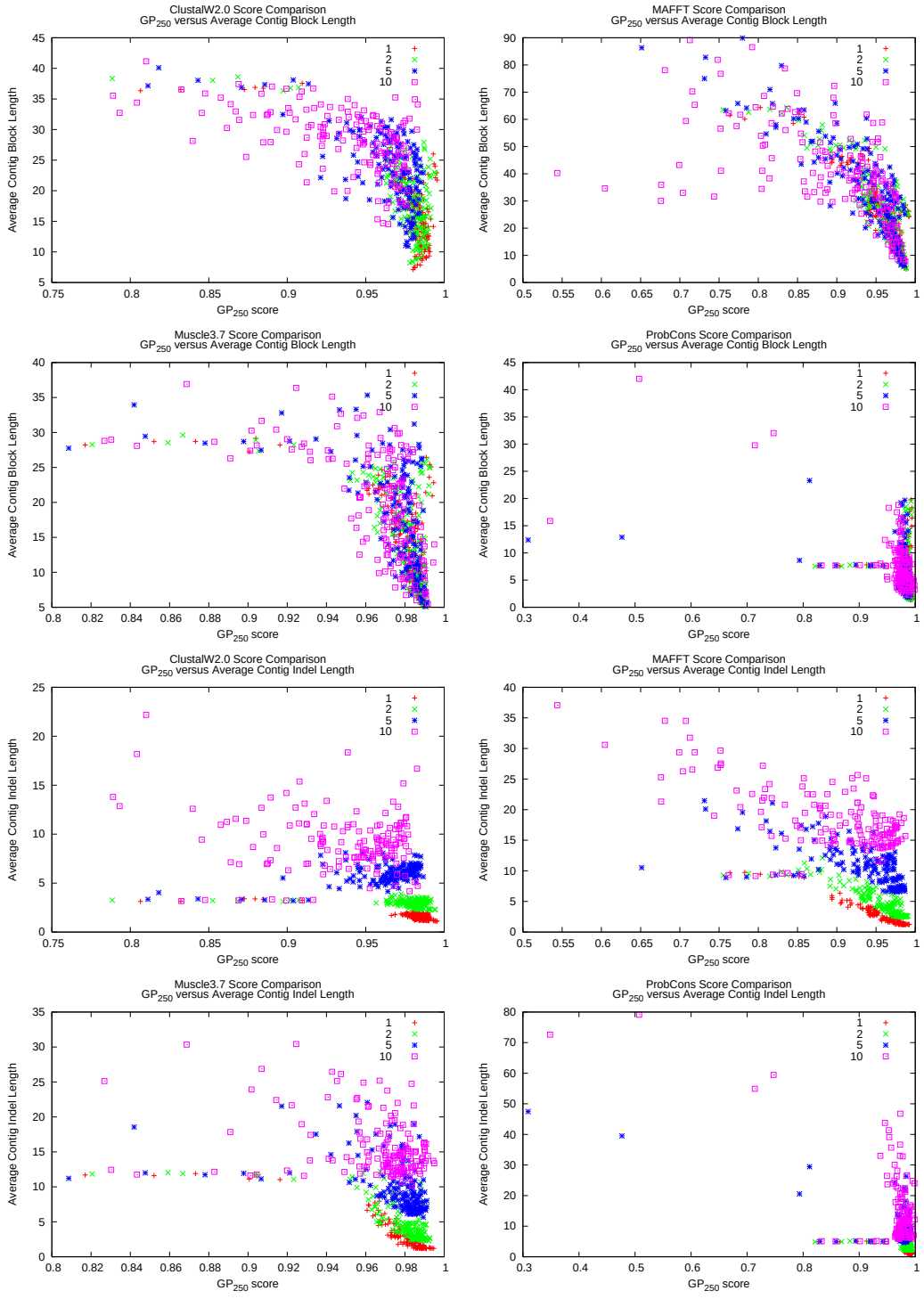

ClustalW2.0 Score Comparison
GP250 versus Average Contig Block Length
MAFFT Score Comparison
GP250 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Block Length
ProbCons Score Comparison
GP250 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP250 versus Average Contig Indel Length
MAFFT Score Comparison
GP250 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Indel Length
ProbCons Score Comparison
GP250 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP250 score
1
2
5
10

Figure B.1

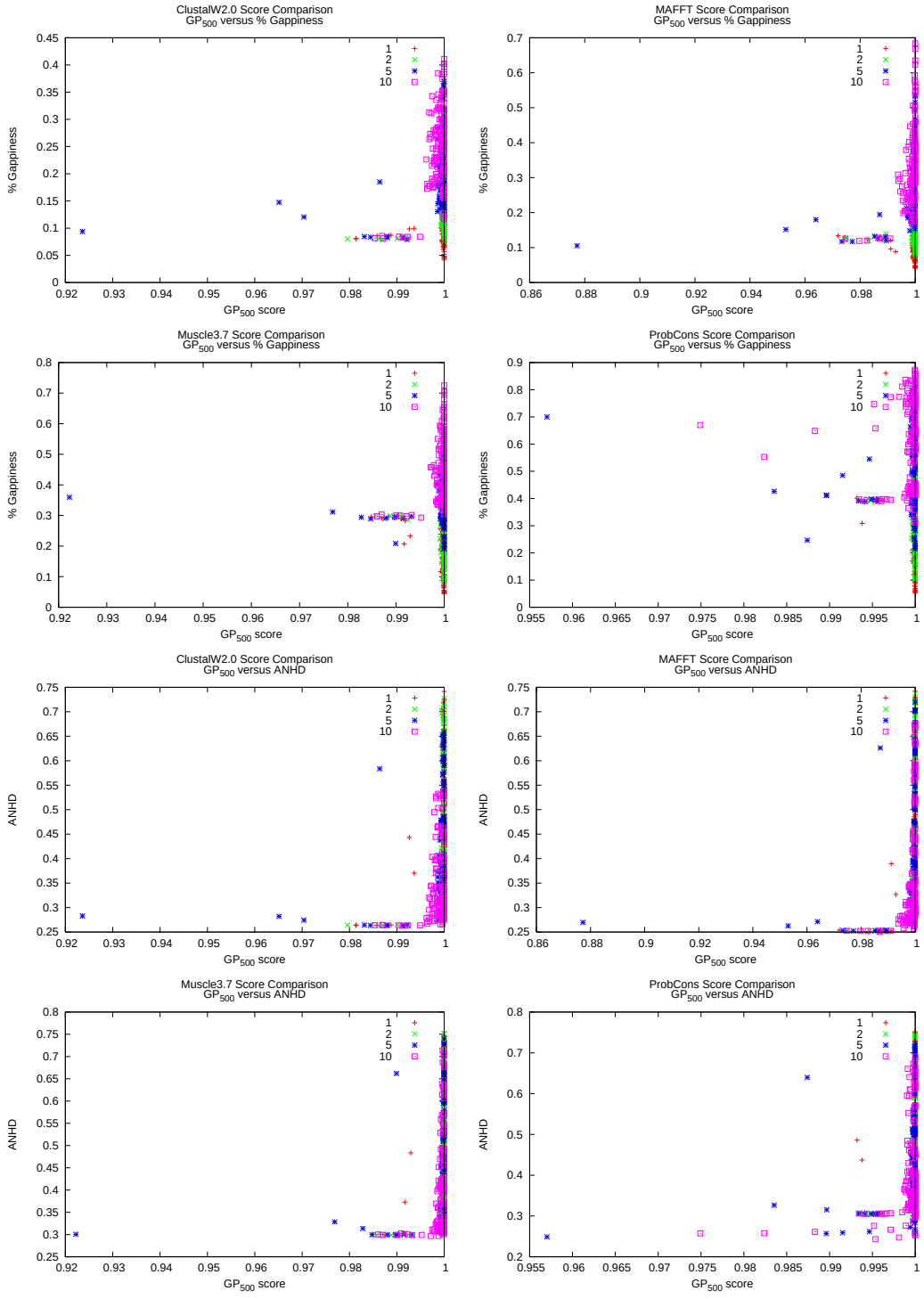

ClustalW2.0 Score Comparison
GP500 versus % Gappiness
MAFFT Score Comparison
GP500 versus % Gappiness
Muscle3.7 Score Comparison
GP500 versus % Gappiness
ProbCons Score Comparison
GP500 versus % Gappiness
ClustalW2.0 Score Comparison
GP500 versus ANHD
MAFFT Score Comparison
GP500 versus ANHD
Muscle3.7 Score Comparison
GP500 versus ANHD
ProbCons Score Comparison
GP500 versus ANHD
% Gappiness
ANHD
GP500 score
1
2
5
10

Figure B.1

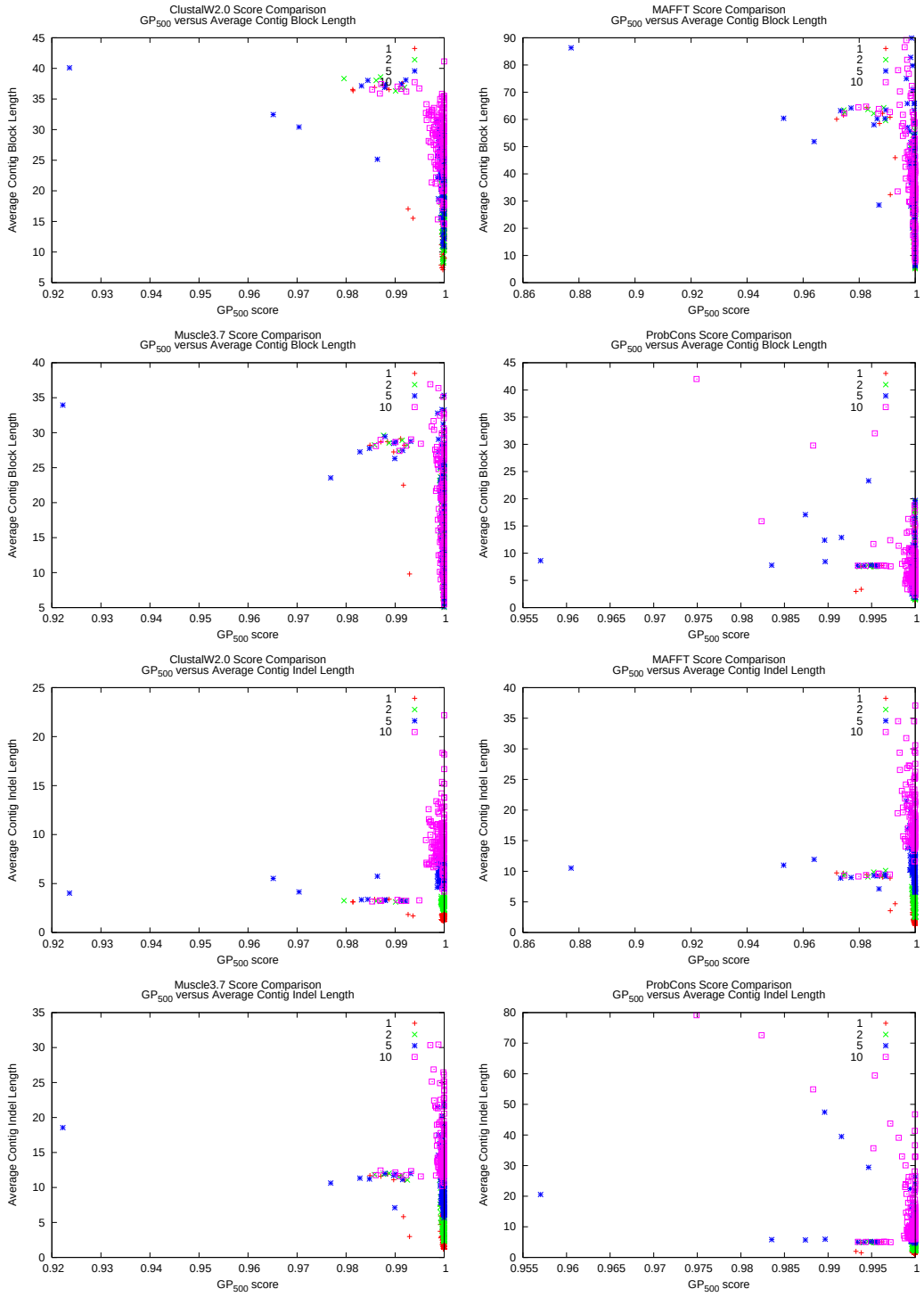

ClustalW2.0 Score Comparison
GP500 versus Average Contig Block Length
MAFFT Score Comparison
GP500 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Block Length
ProbCons Score Comparison
GP500 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP500 versus Average Contig Indel Length
MAFFT Score Comparison
GP500 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Indel Length
ProbCons Score Comparison
GP500 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP500 score
1
2
5
10

Figure B.1

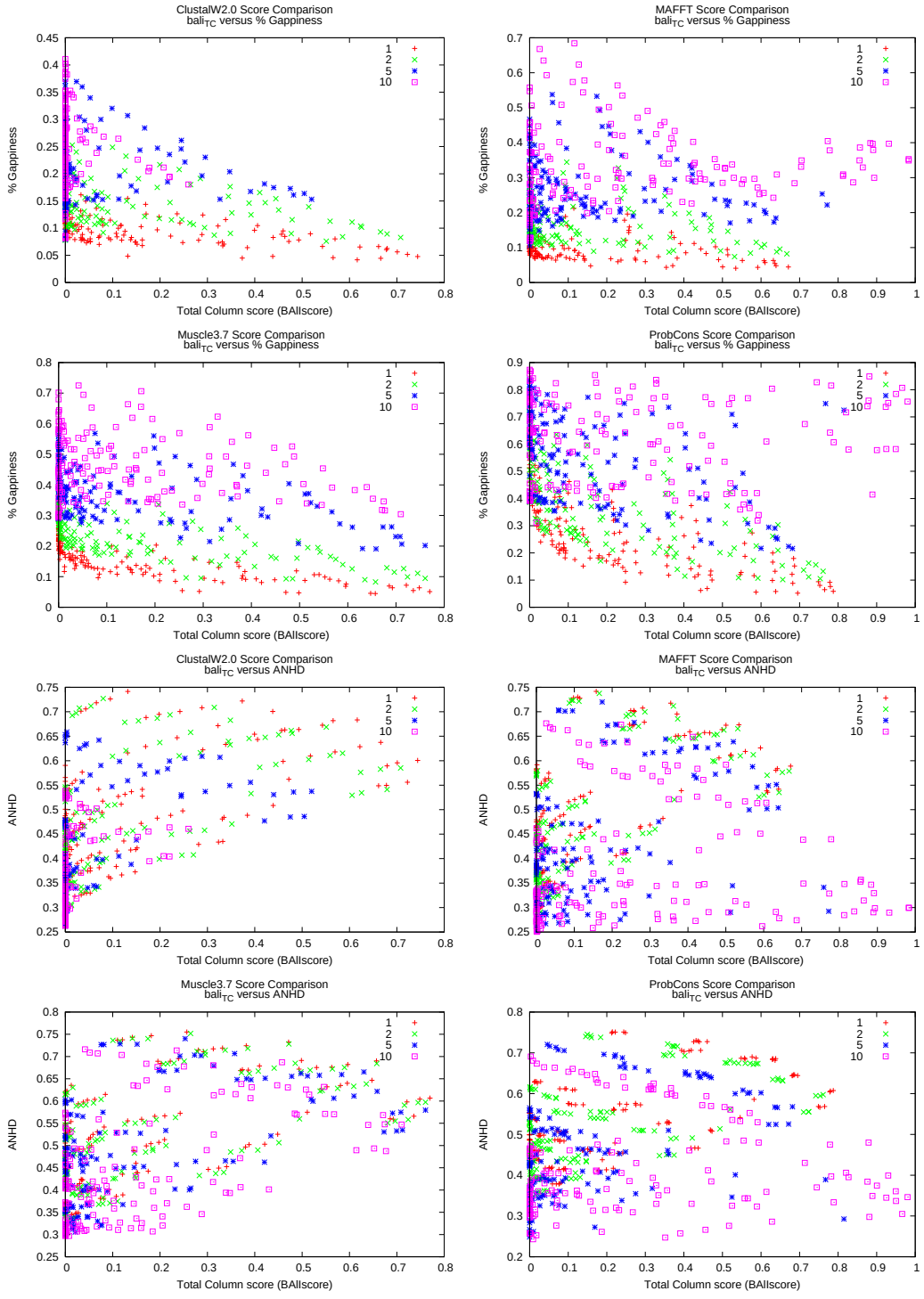

ClustalW2.0 Score Comparison
bali_TC versus % Gappiness
MAFFT Score Comparison
bali_TC versus % Gappiness
Muscle3.7 Score Comparison
bali_TC versus % Gappiness
ProbCons Score Comparison
bali_TC versus % Gappiness
ClustalW2.0 Score Comparison
bali_TC versus ANHD
MAFFT Score Comparison
bali_TC versus ANHD
Muscle3.7 Score Comparison
bali_TC versus ANHD
ProbCons Score Comparison
bali_TC versus ANHD
% Gappiness
ANHD
Total Column score (BAIIscore)
1
2
5
10

Figure B.1

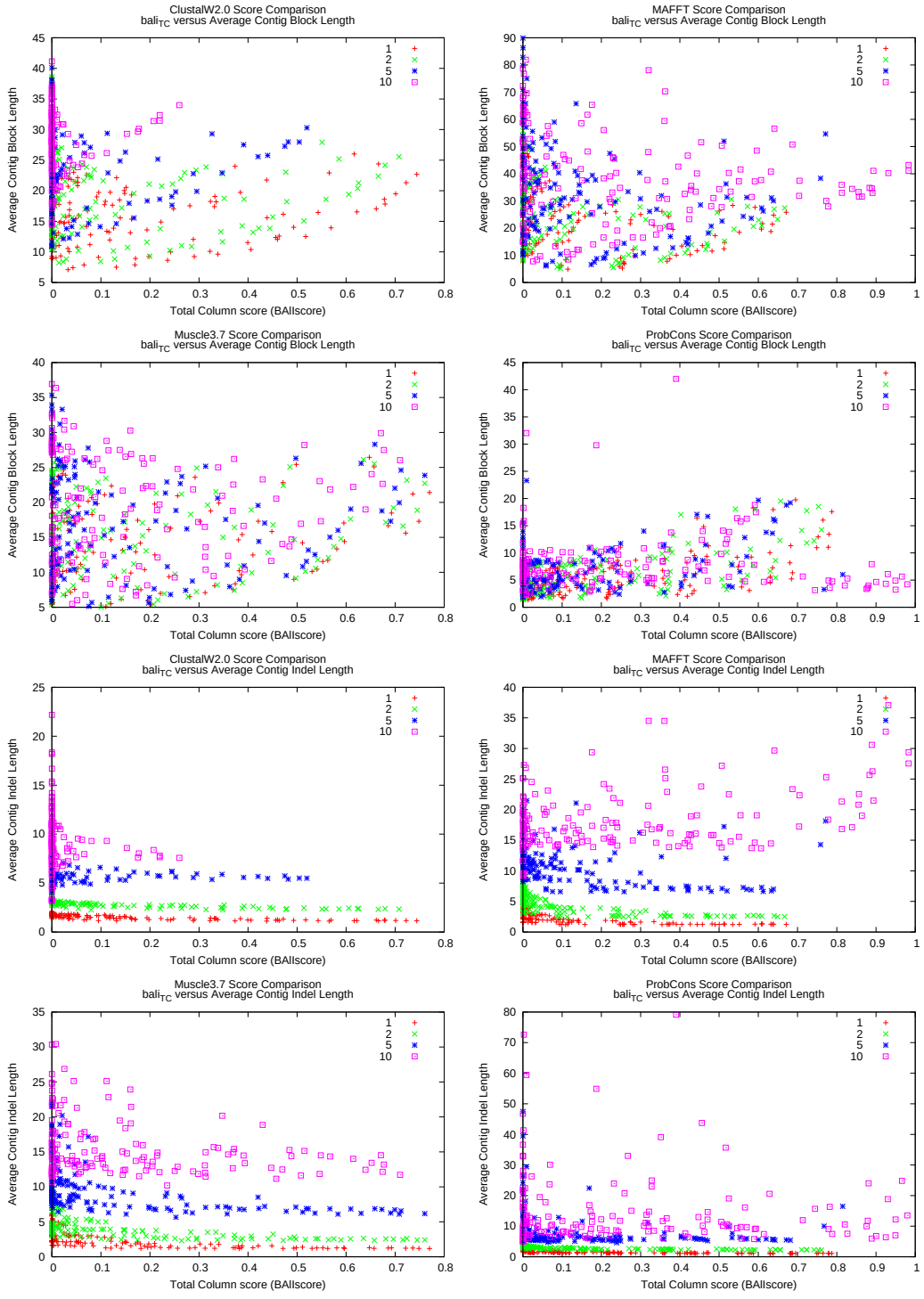

ClustalW2.0 Score Comparison
bali_TC versus Average Contig Block Length
MAFFT Score Comparison
bali_TC versus Average Contig Block Length
Muscle3.7 Score Comparison
bali_TC versus Average Contig Block Length
ProbCons Score Comparison
bali_TC versus Average Contig Block Length
ClustalW2.0 Score Comparison
bali_TC versus Average Contig Indel Length
MAFFT Score Comparison
bali_TC versus Average Contig Indel Length
Muscle3.7 Score Comparison
bali_TC versus Average Contig Indel Length
ProbCons Score Comparison
bali_TC versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Total Column score (BAIIscore)
1
2
5
10

Figure B.1

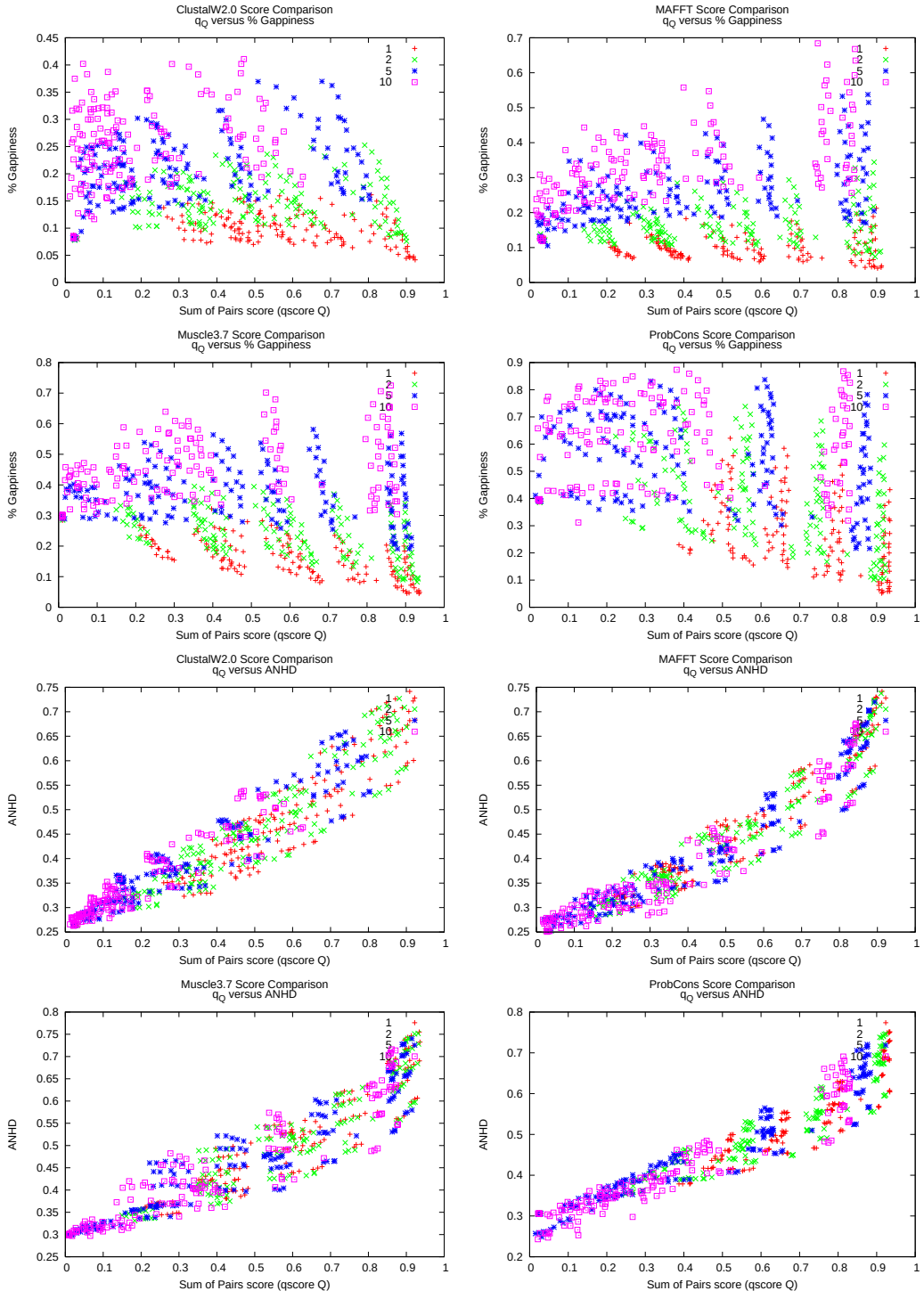

ClustalW2.0 Score Comparison
$q_Q$ versus % Gappiness
MAFFT Score Comparison
$q_Q$ versus % Gappiness
Muscle3.7 Score Comparison
$q_Q$ versus % Gappiness
ProbCons Score Comparison
$q_Q$ versus % Gappiness
ClustalW2.0 Score Comparison
$q_Q$ versus ANHD
MAFFT Score Comparison
$q_Q$ versus ANHD
Muscle3.7 Score Comparison
$q_Q$ versus ANHD
ProbCons Score Comparison
$q_Q$ versus ANHD
% Gappiness
ANHD
Sum of Pairs score (qscore Q)
1
2
5
10

Figure B.1

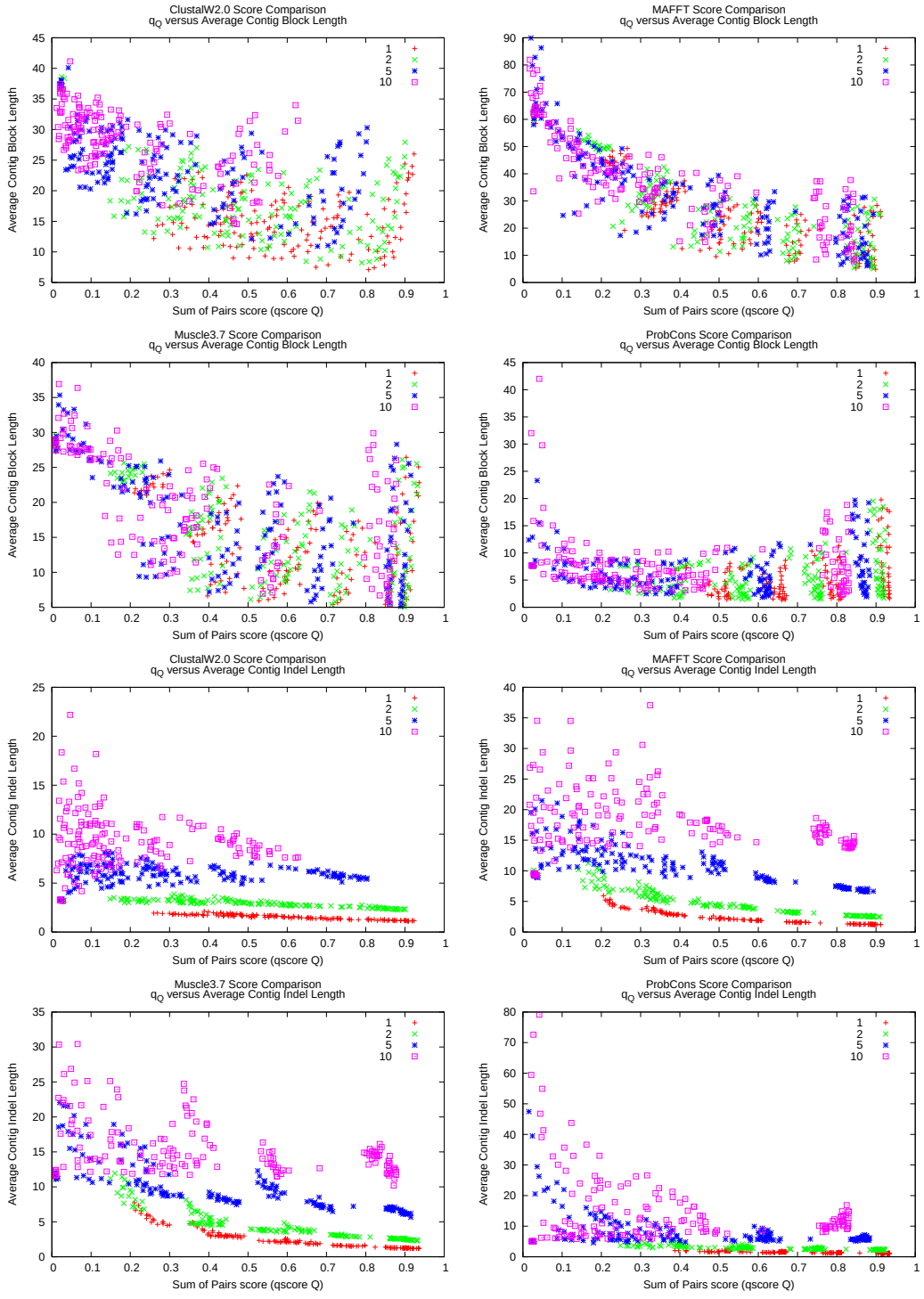

ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Block Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Block Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Indel Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Indel Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Sum of Pairs score (qscore Q)
1
2
5
10

Figure B.1

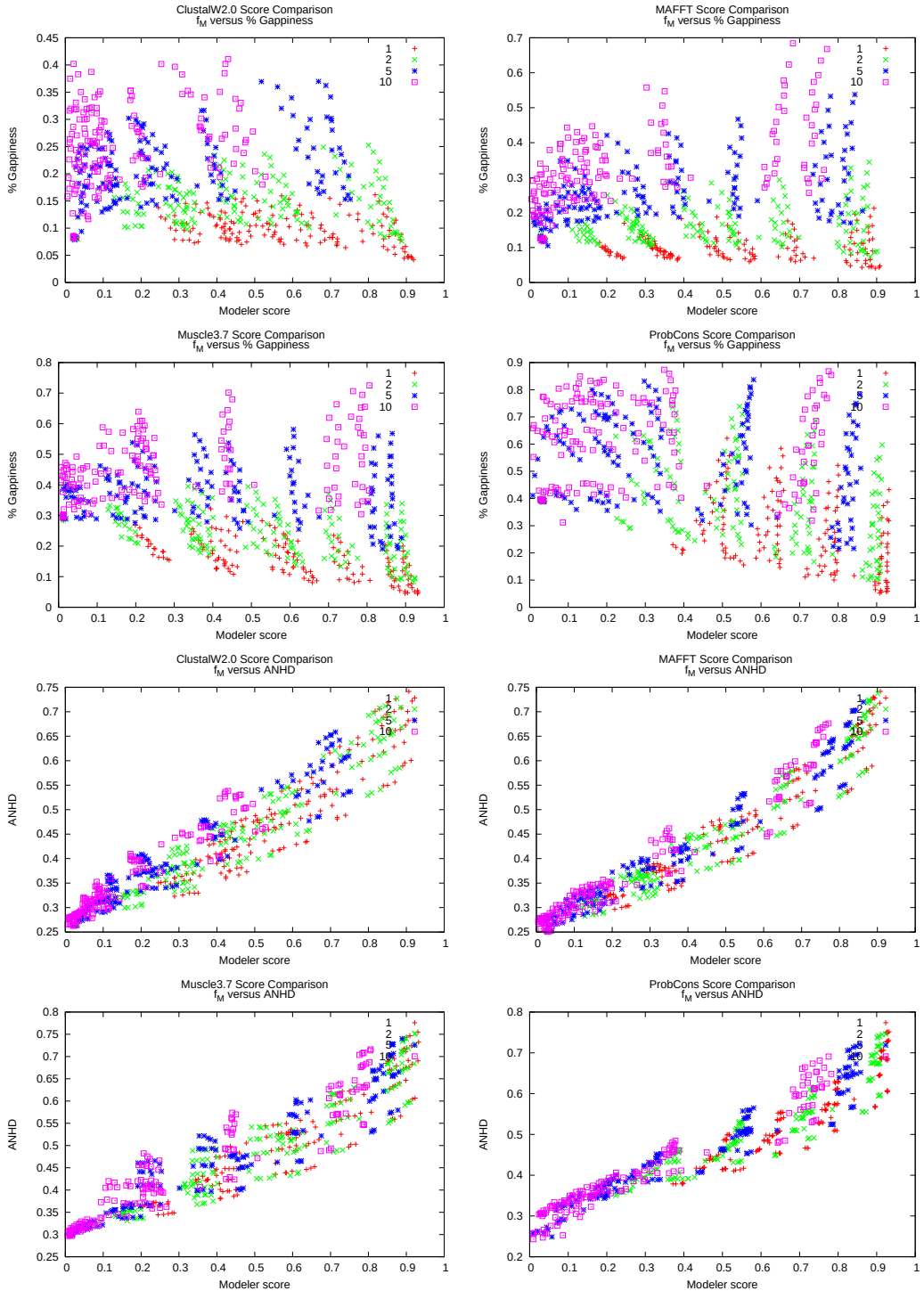

ClustalW2.0 Score Comparison
$f_M$ versus % Gappiness
MAFFT Score Comparison
$f_M$ versus % Gappiness
Muscle3.7 Score Comparison
$f_M$ versus % Gappiness
ProbCons Score Comparison
$f_M$ versus % Gappiness
ClustalW2.0 Score Comparison
$f_M$ versus ANHD
MAFFT Score Comparison
$f_M$ versus ANHD
Muscle3.7 Score Comparison
$f_M$ versus ANHD
ProbCons Score Comparison
$f_M$ versus ANHD
% Gappiness
ANHD
Modeler score
1
2
5
10

Figure B.1

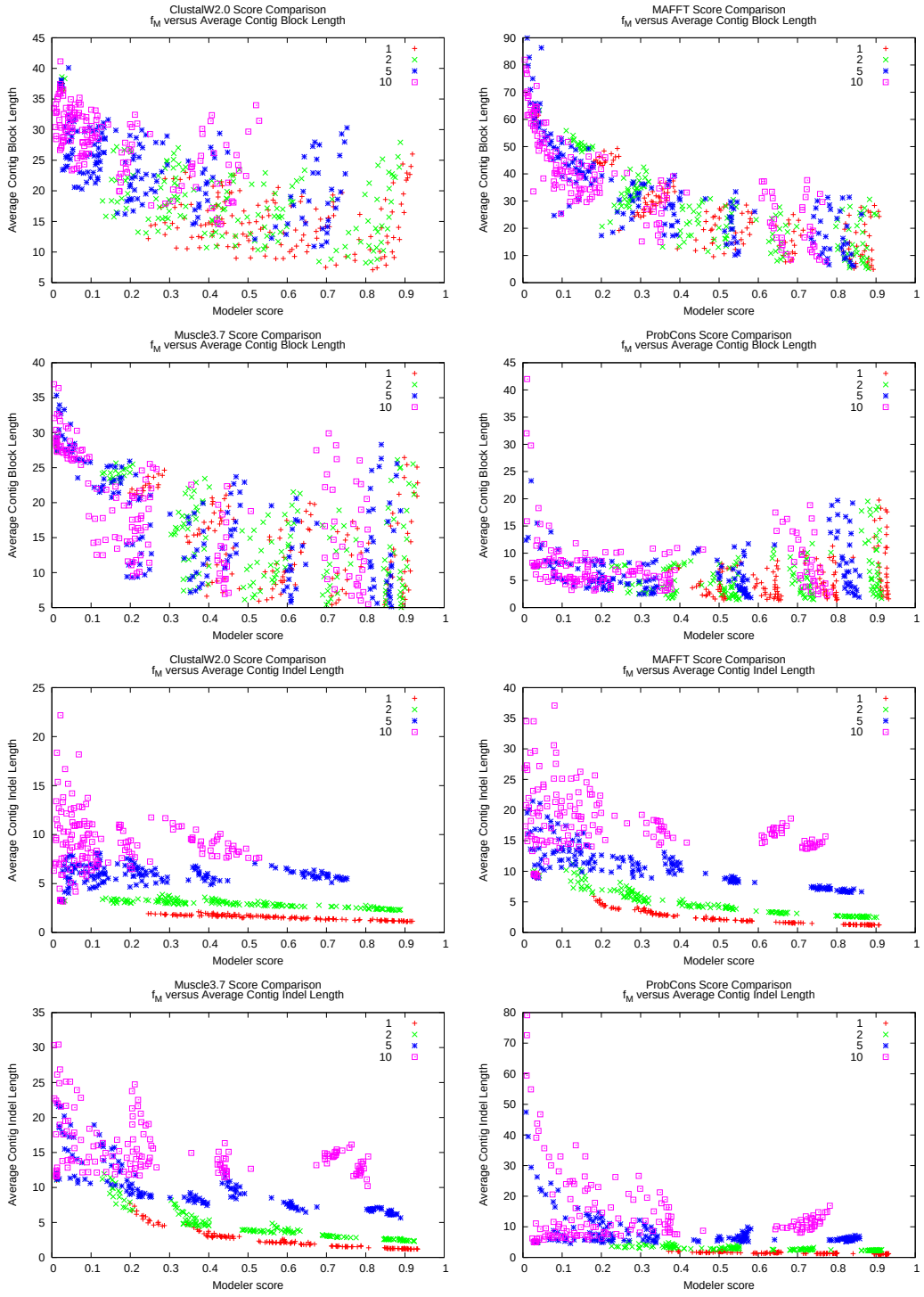

ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Block Length
MAFFT Score Comparison
$f_M$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Block Length
ProbCons Score Comparison
$f_M$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Indel Length
MAFFT Score Comparison
$f_M$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Indel Length
ProbCons Score Comparison
$f_M$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Modeler score
1
2
5
10

Figure B.1

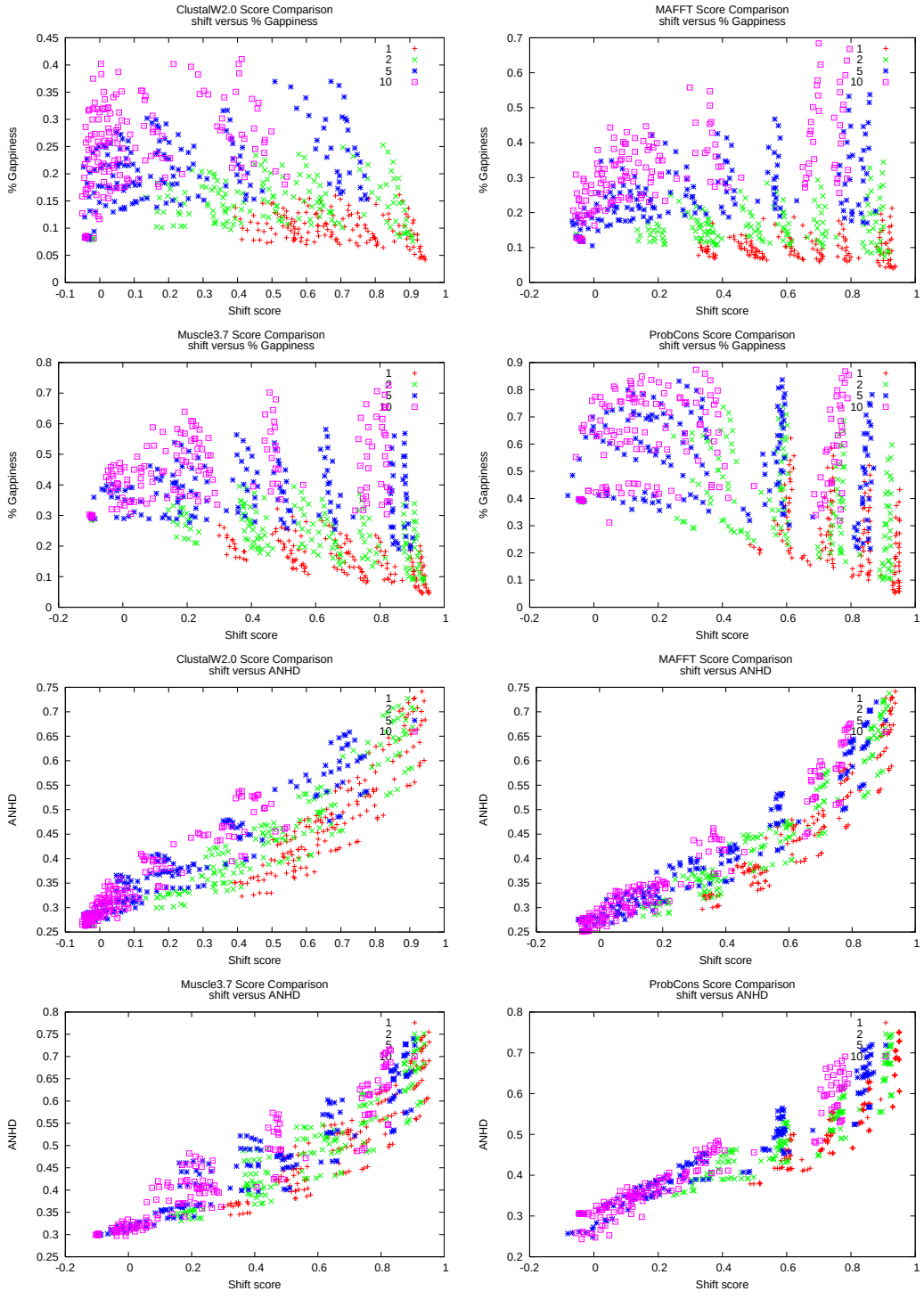

ClustalW2.0 Score Comparison
shift versus % Gappiness
MAFFT Score Comparison
shift versus % Gappiness
Muscle3.7 Score Comparison
shift versus % Gappiness
ProbCons Score Comparison
shift versus % Gappiness
ClustalW2.0 Score Comparison
shift versus ANHD
MAFFT Score Comparison
shift versus ANHD
Muscle3.7 Score Comparison
shift versus ANHD
ProbCons Score Comparison
shift versus ANHD
% Gappiness
ANHD
Shift score
1
2
5
10

Figure B.1

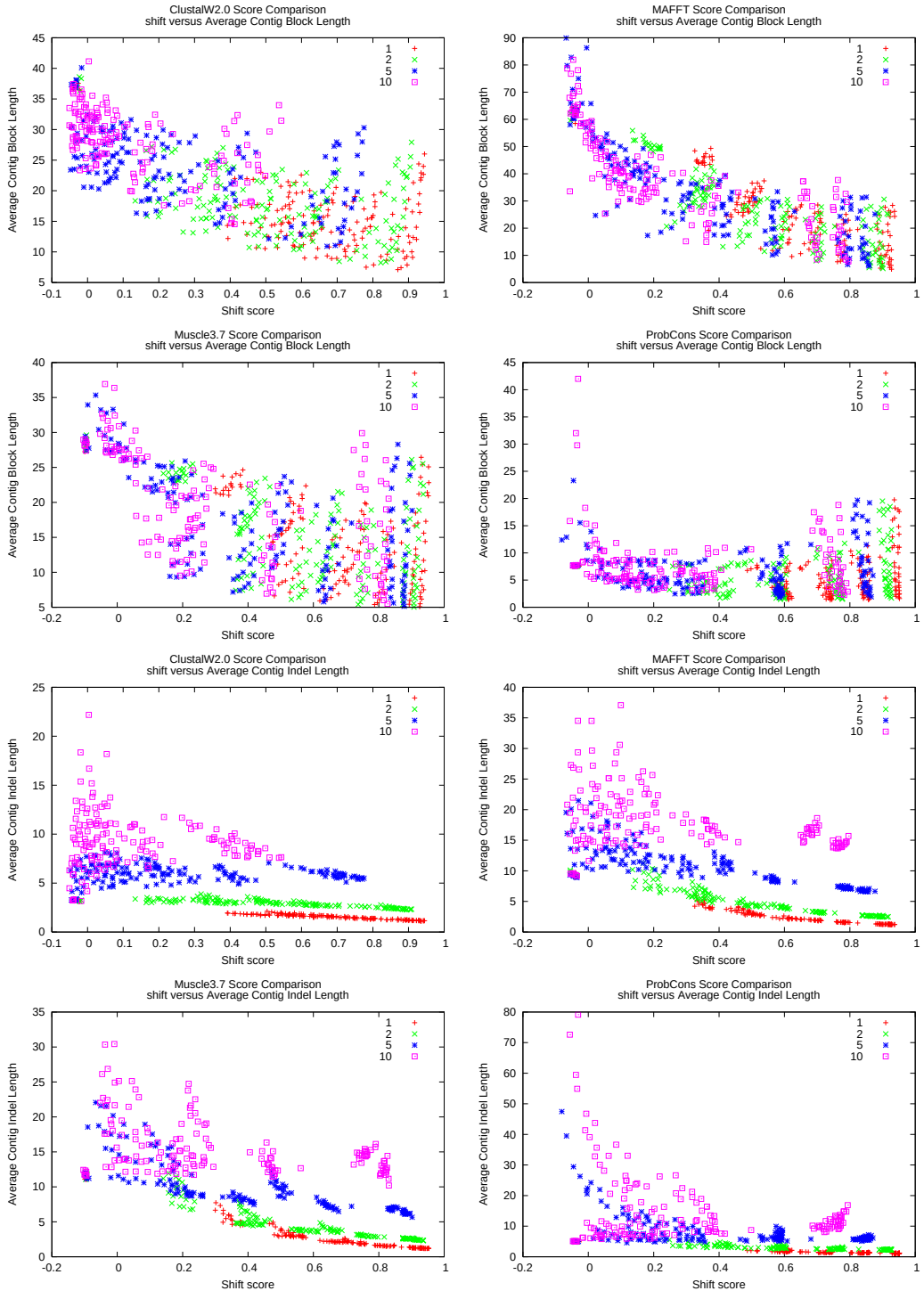

ClustalW2.0 Score Comparison
shift versus Average Contig Block Length
MAFFT Score Comparison
shift versus Average Contig Block Length
Muscle3.7 Score Comparison
shift versus Average Contig Block Length
ProbCons Score Comparison
shift versus Average Contig Block Length
ClustalW2.0 Score Comparison
shift versus Average Contig Indel Length
MAFFT Score Comparison
shift versus Average Contig Indel Length
Muscle3.7 Score Comparison
shift versus Average Contig Indel Length
ProbCons Score Comparison
shift versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Shift score
1
2
5
10

Figure B.1

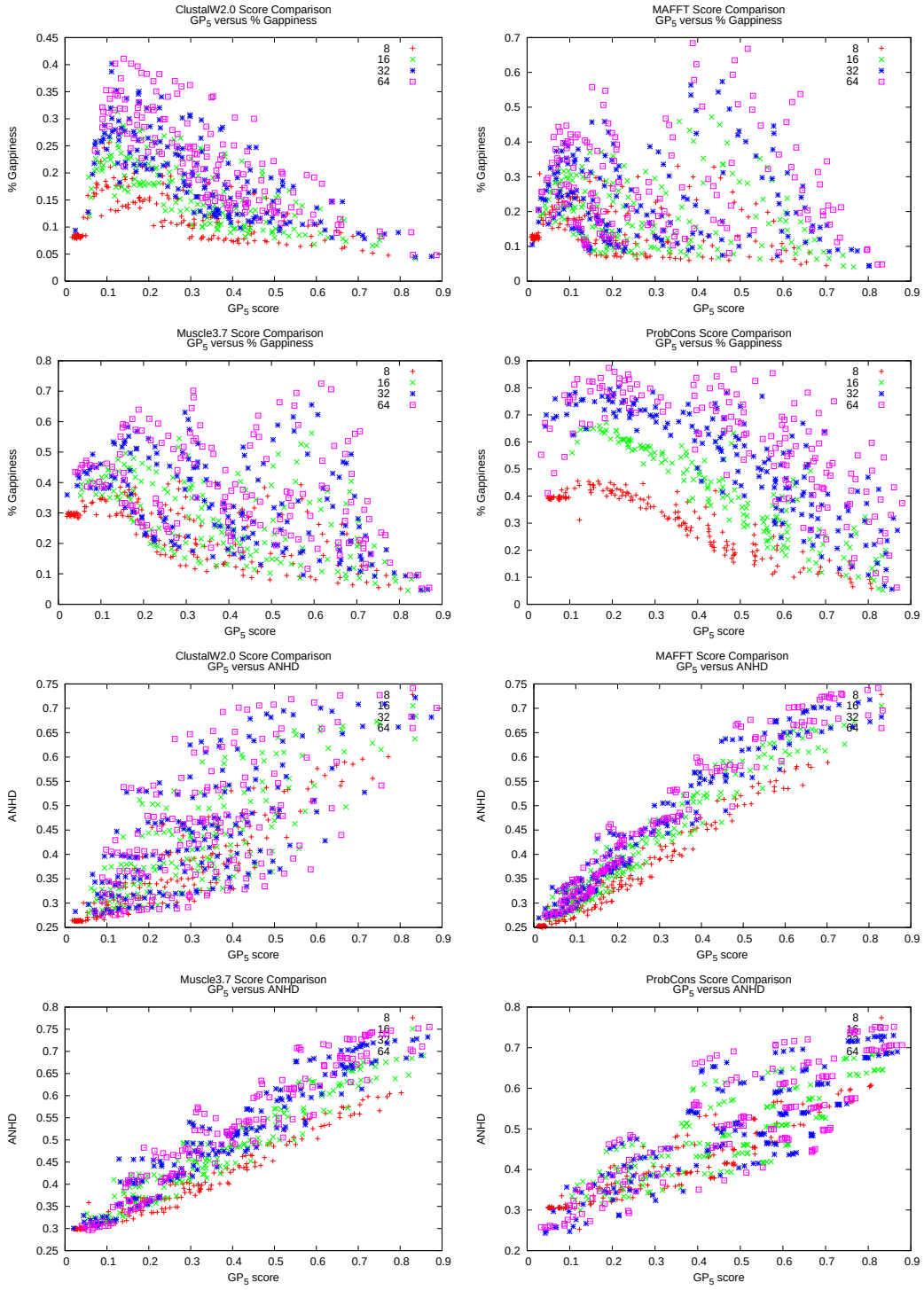

ClustalW2.0 Score Comparison
GP5 versus % Gappiness
MAFFT Score Comparison
GP5 versus % Gappiness
Muscle3.7 Score Comparison
GP5 versus % Gappiness
ProbCons Score Comparison
GP5 versus % Gappiness
ClustalW2.0 Score Comparison
GP5 versus ANHD
MAFFT Score Comparison
GP5 versus ANHD
Muscle3.7 Score Comparison
GP5 versus ANHD
ProbCons Score Comparison
GP5 versus ANHD
% Gappiness
ANHD
GP5 score
8
16
32
64

Figure B.1

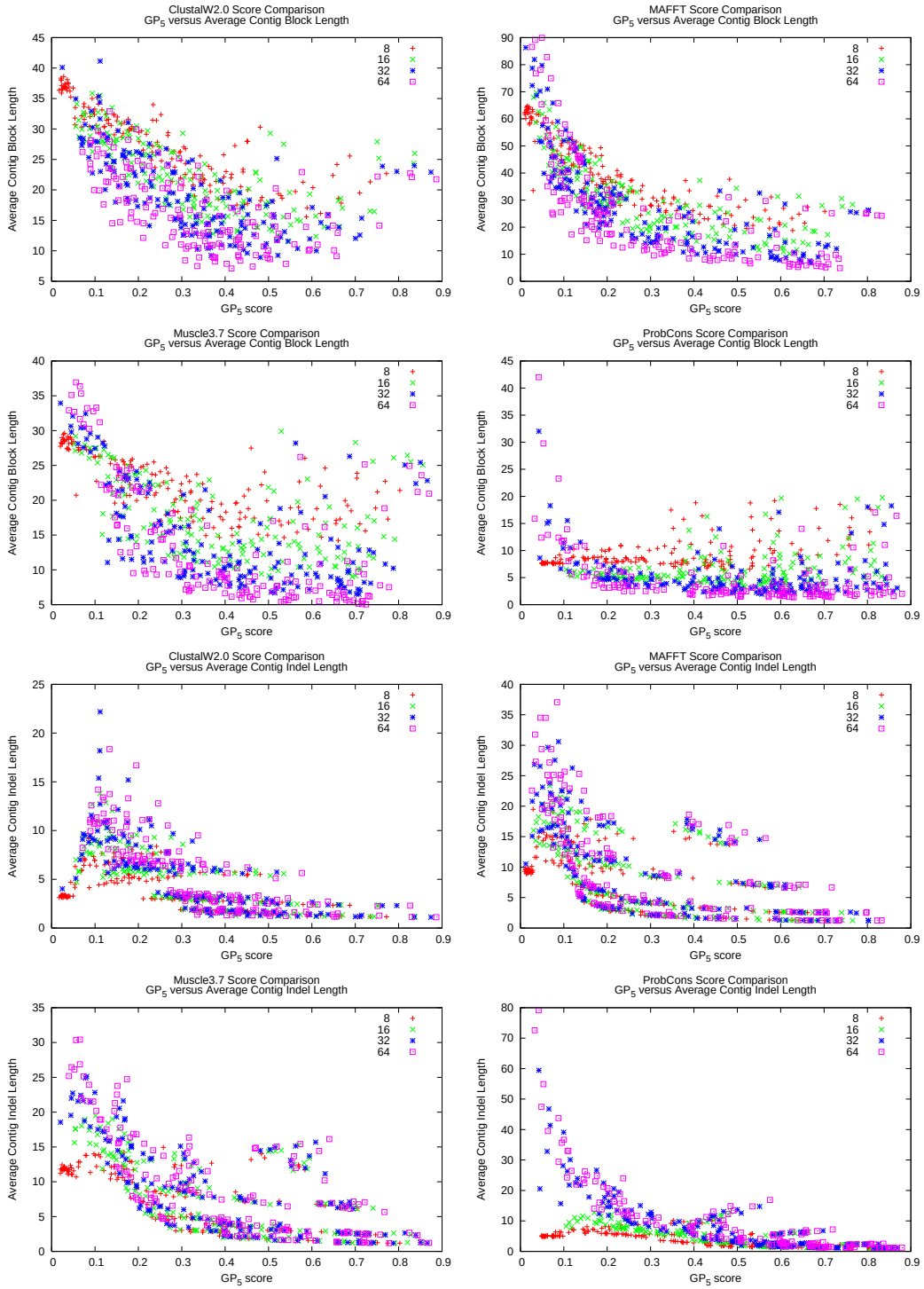

ClustalW2.0 Score Comparison
GP5 versus Average Contig Block Length
MAFFT Score Comparison
GP5 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Block Length
ProbCons Score Comparison
GP5 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP5 versus Average Contig Indel Length
MAFFT Score Comparison
GP5 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Indel Length
ProbCons Score Comparison
GP5 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP5 score
8
16
32
64

Figure B.1

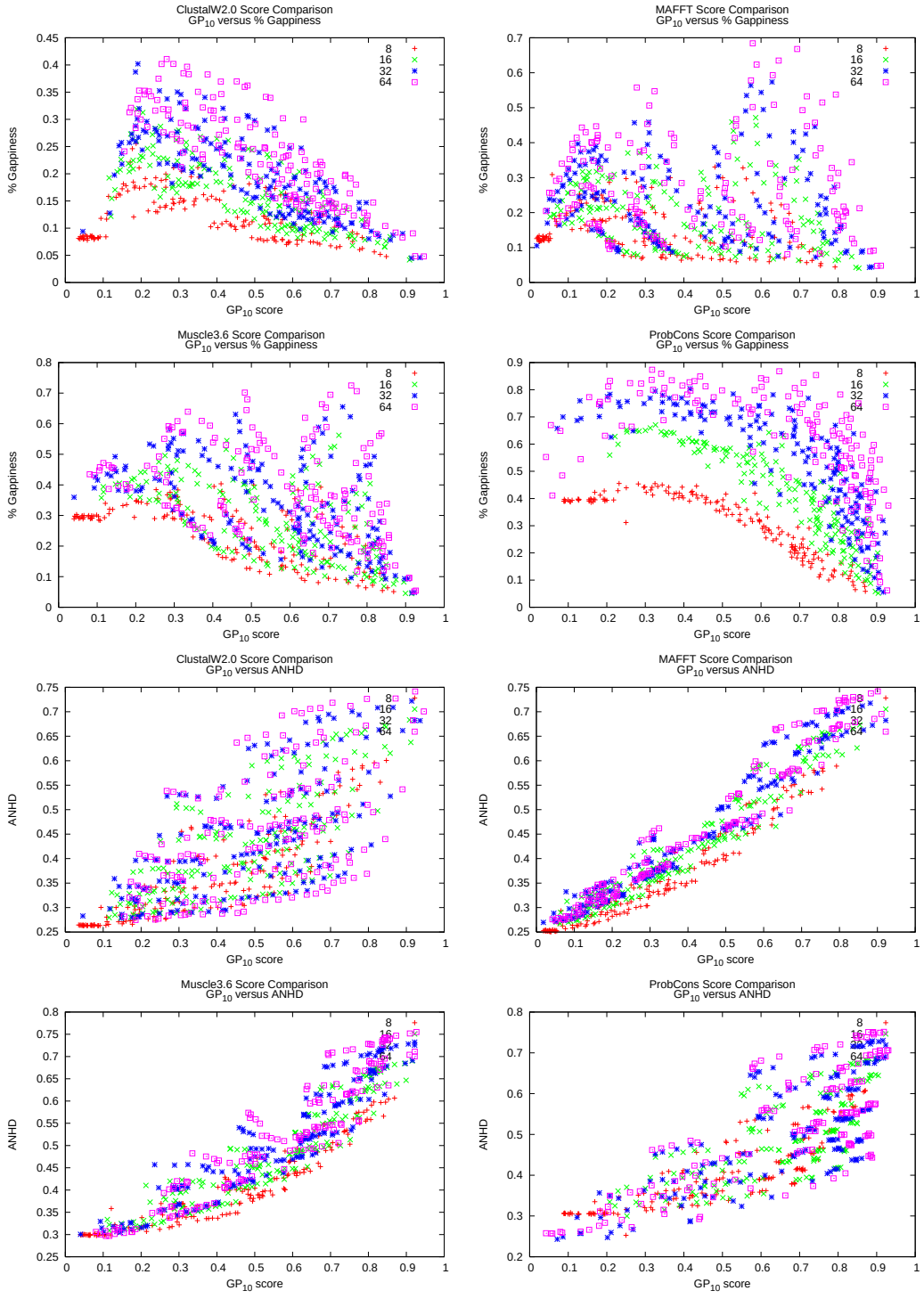

ClustalW2.0 Score Comparison
GP10 versus % Gappiness
MAFFT Score Comparison
GP10 versus % Gappiness
Muscle3.6 Score Comparison
GP10 versus % Gappiness
ProbCons Score Comparison
GP10 versus % Gappiness
ClustalW2.0 Score Comparison
GP10 versus ANHD
MAFFT Score Comparison
GP10 versus ANHD
Muscle3.6 Score Comparison
GP10 versus ANHD
ProbCons Score Comparison
GP10 versus ANHD
% Gappiness
ANHD
GP10 score
8
16
32
64

Figure B.1

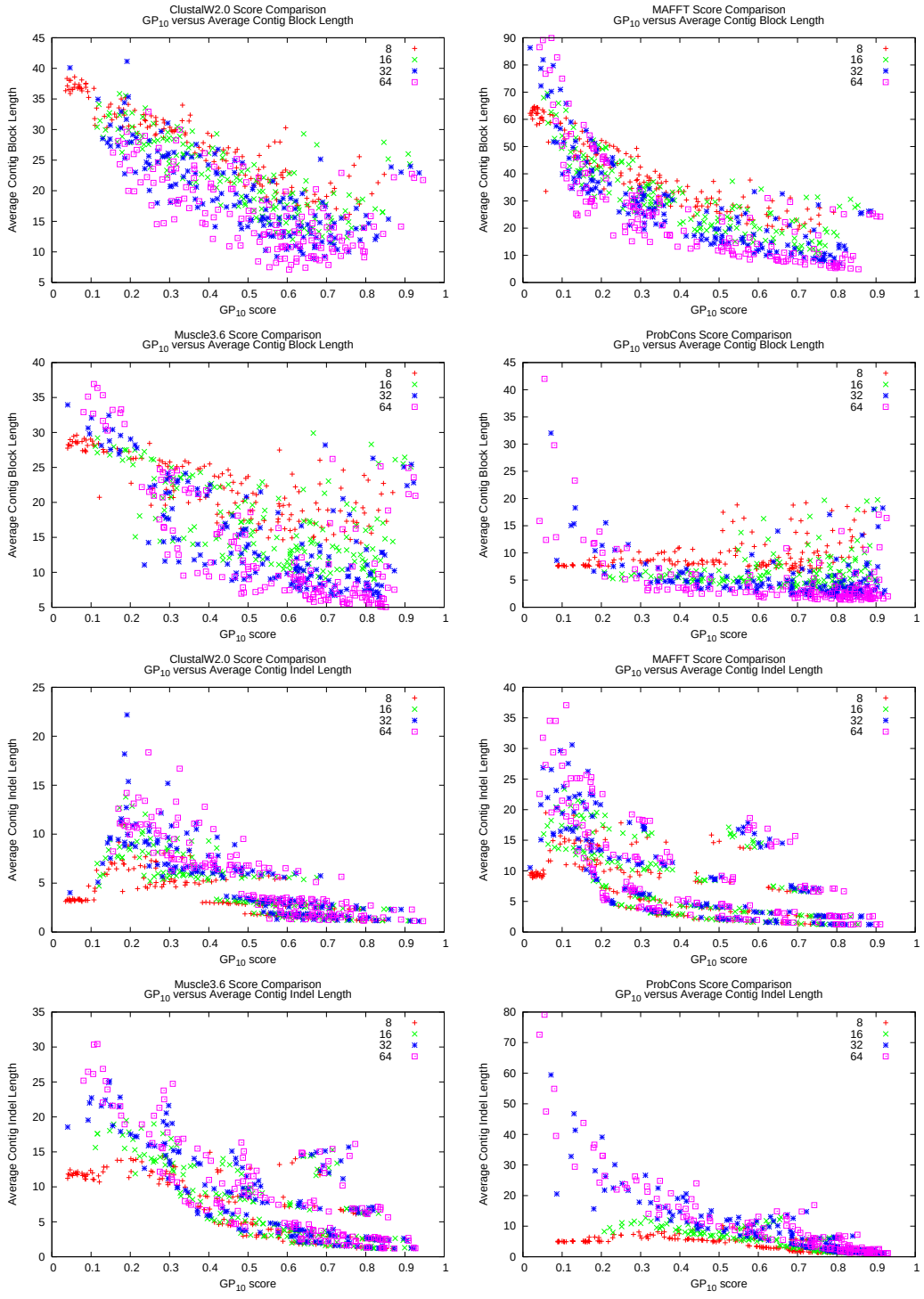

ClustalW2.0 Score Comparison
GP10 versus Average Contig Block Length
MAFFT Score Comparison
GP10 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Block Length
ProbCons Score Comparison
GP10 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP10 versus Average Contig Indel Length
MAFFT Score Comparison
GP10 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Indel Length
ProbCons Score Comparison
GP10 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP10 score
8
16
32
64

Figure B.1

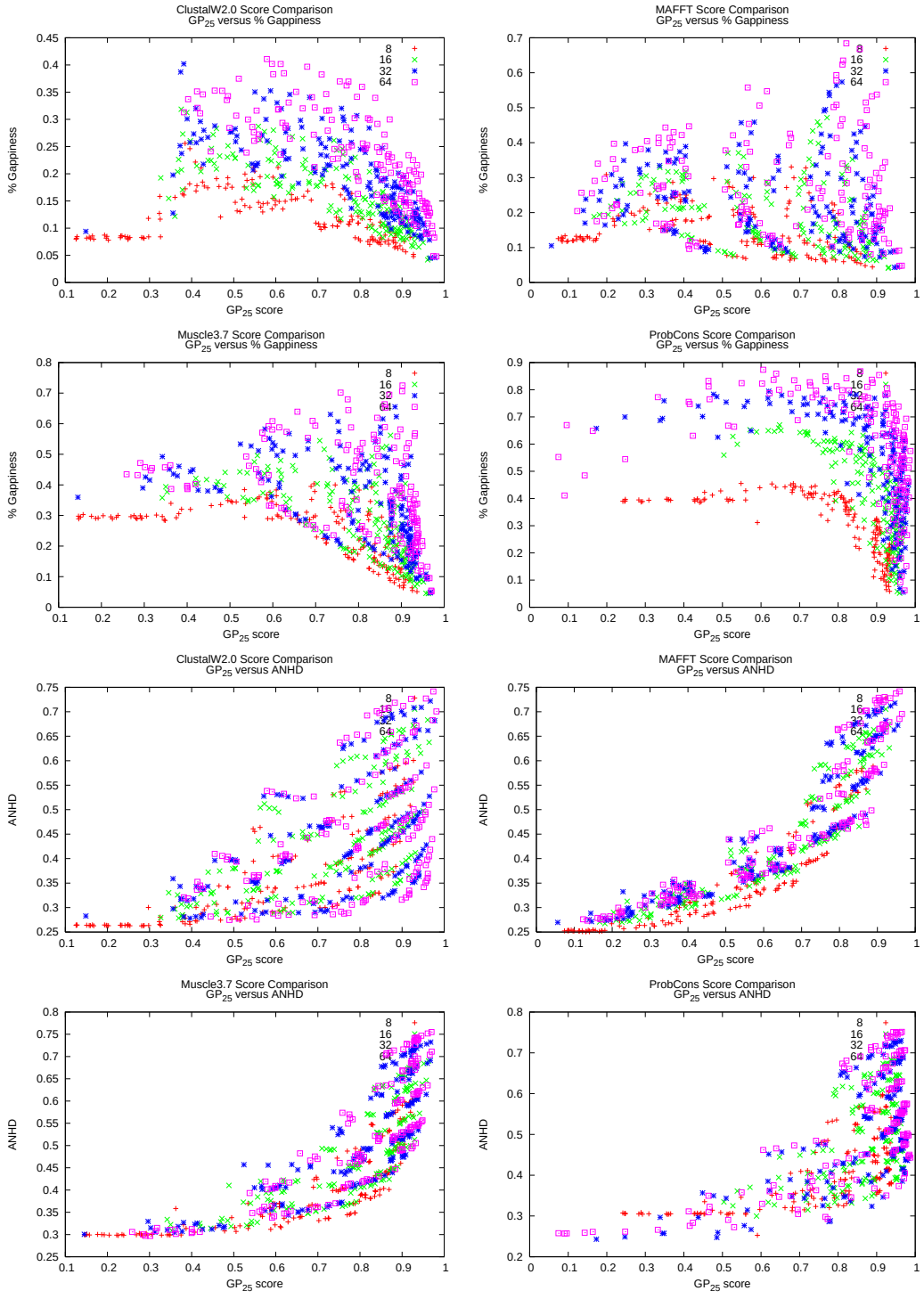

ClustalW2.0 Score Comparison
GP25 versus % Gappiness
MAFFT Score Comparison
GP25 versus % Gappiness
Muscle3.7 Score Comparison
GP25 versus % Gappiness
ProbCons Score Comparison
GP25 versus % Gappiness
ClustalW2.0 Score Comparison
GP25 versus ANHD
MAFFT Score Comparison
GP25 versus ANHD
Muscle3.7 Score Comparison
GP25 versus ANHD
ProbCons Score Comparison
GP25 versus ANHD
% Gappiness
ANHD
GP25 score
8
16
32
64

Figure B.1

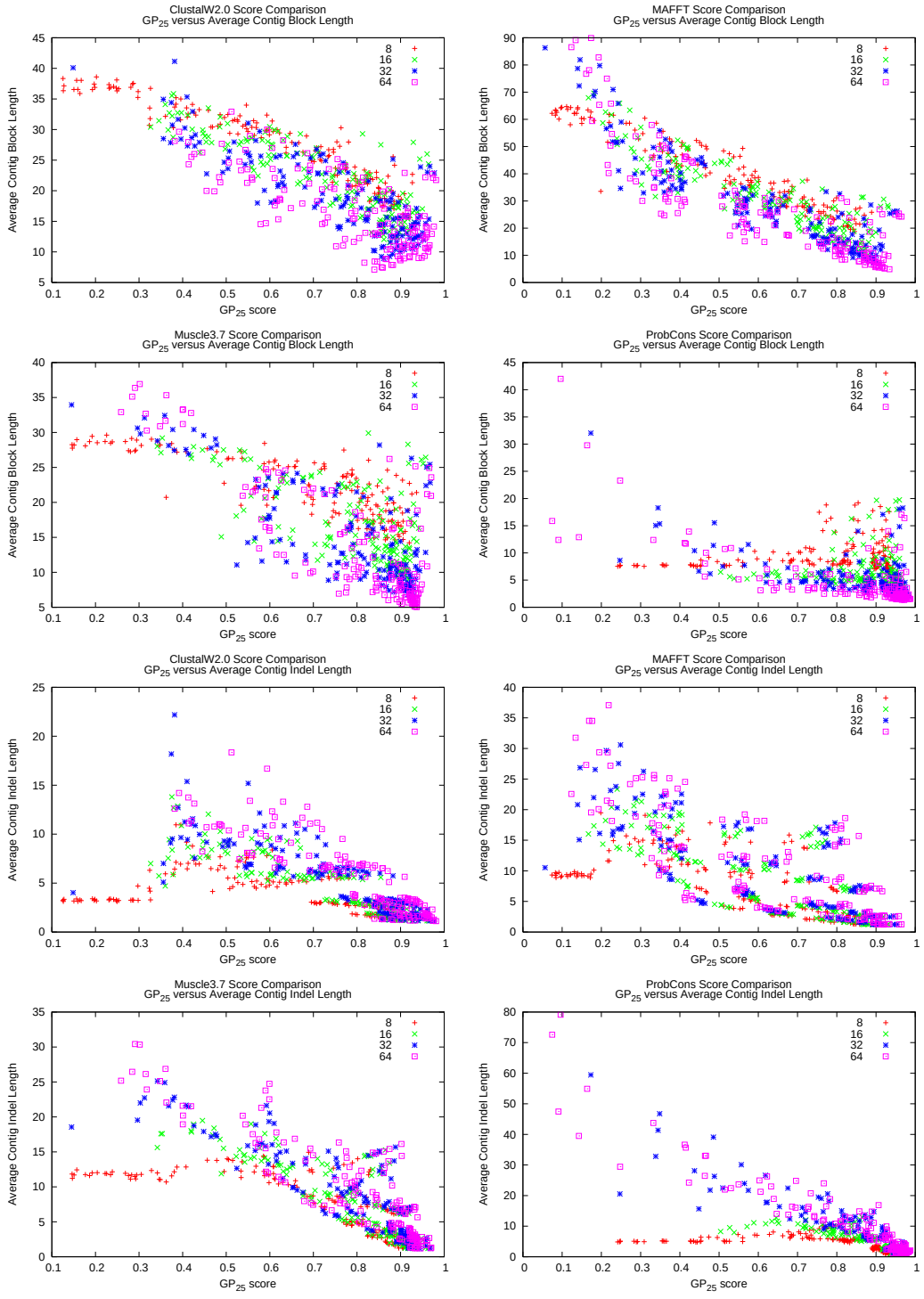

ClustalW2.0 Score Comparison
GP25 versus Average Contig Block Length
MAFFT Score Comparison
GP25 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Block Length
ProbCons Score Comparison
GP25 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP25 versus Average Contig Indel Length
MAFFT Score Comparison
GP25 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Indel Length
ProbCons Score Comparison
GP25 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP25 score
8
16
32
64

Figure B.1

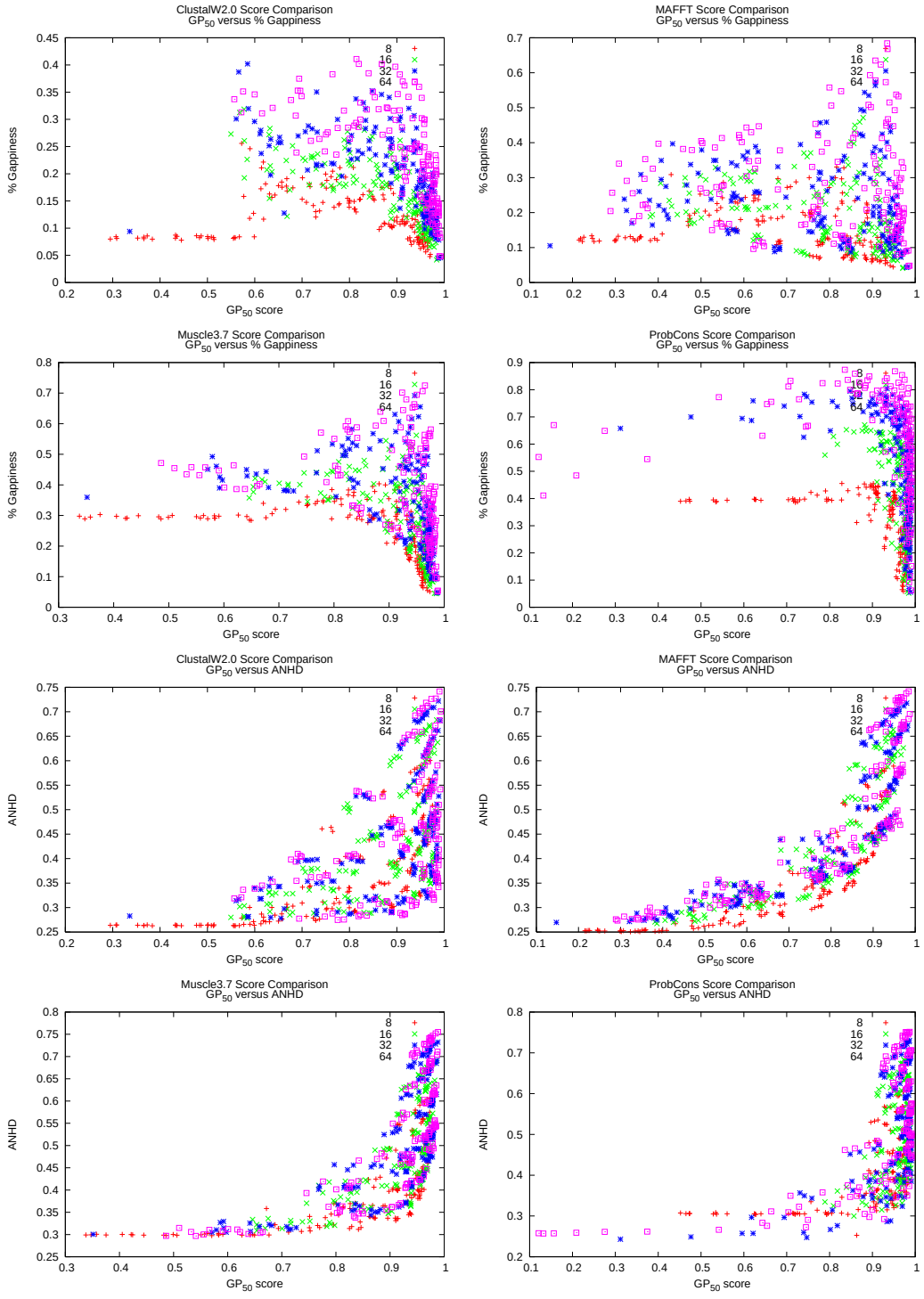

ClustalW2.0 Score Comparison
GP50 versus % Gappiness
MAFFT Score Comparison
GP50 versus % Gappiness
Muscle3.7 Score Comparison
GP50 versus % Gappiness
ProbCons Score Comparison
GP50 versus % Gappiness
ClustalW2.0 Score Comparison
GP50 versus ANHD
MAFFT Score Comparison
GP50 versus ANHD
Muscle3.7 Score Comparison
GP50 versus ANHD
ProbCons Score Comparison
GP50 versus ANHD
% Gappiness
ANHD
GP50 score
8
16
32
64

Figure B.1

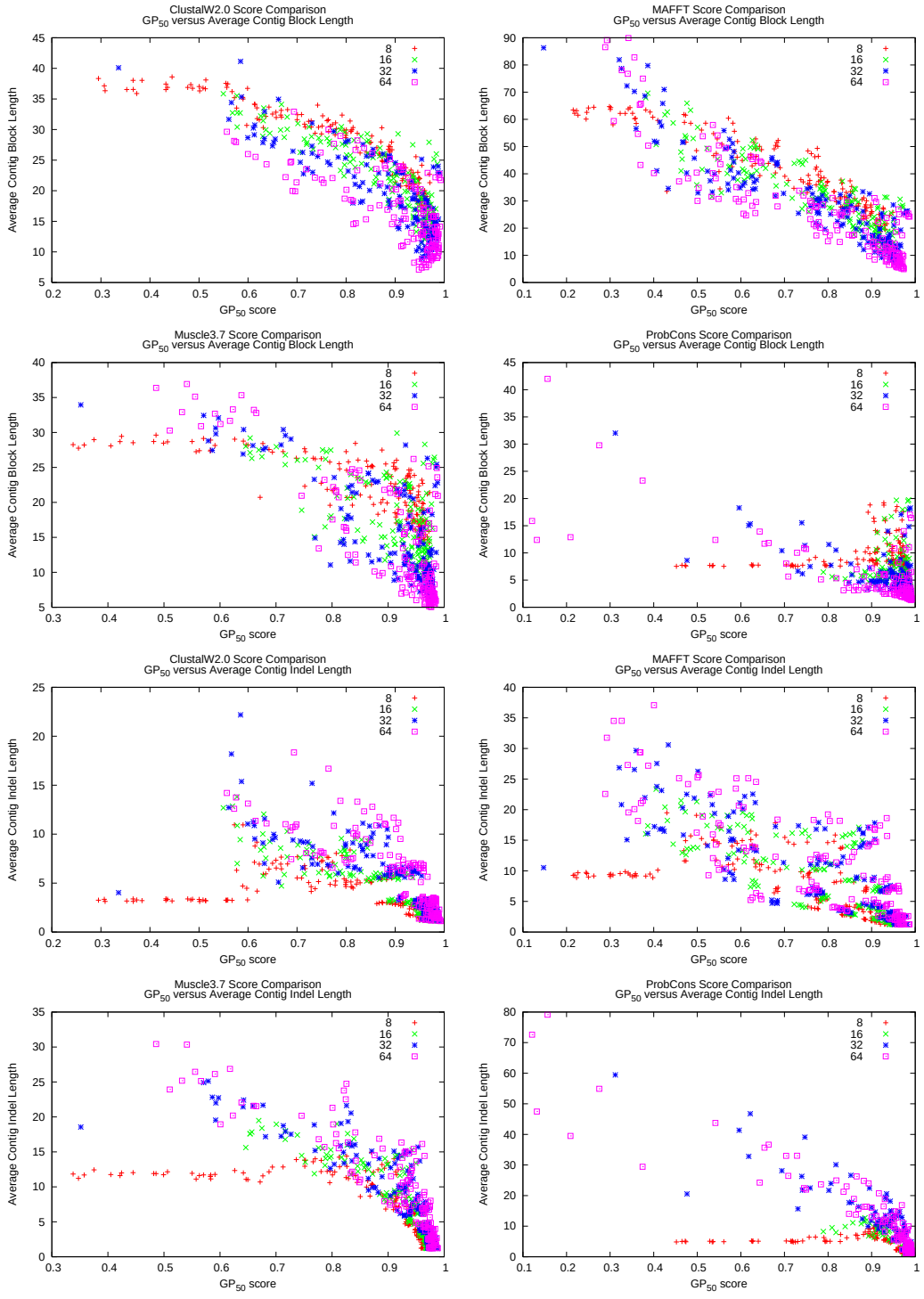

ClustalW2.0 Score Comparison
GP50 versus Average Contig Block Length
MAFFT Score Comparison
GP50 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Block Length
ProbCons Score Comparison
GP50 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP50 versus Average Contig Indel Length
MAFFT Score Comparison
GP50 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Indel Length
ProbCons Score Comparison
GP50 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP50 score
8
16
32
64

Figure B.1

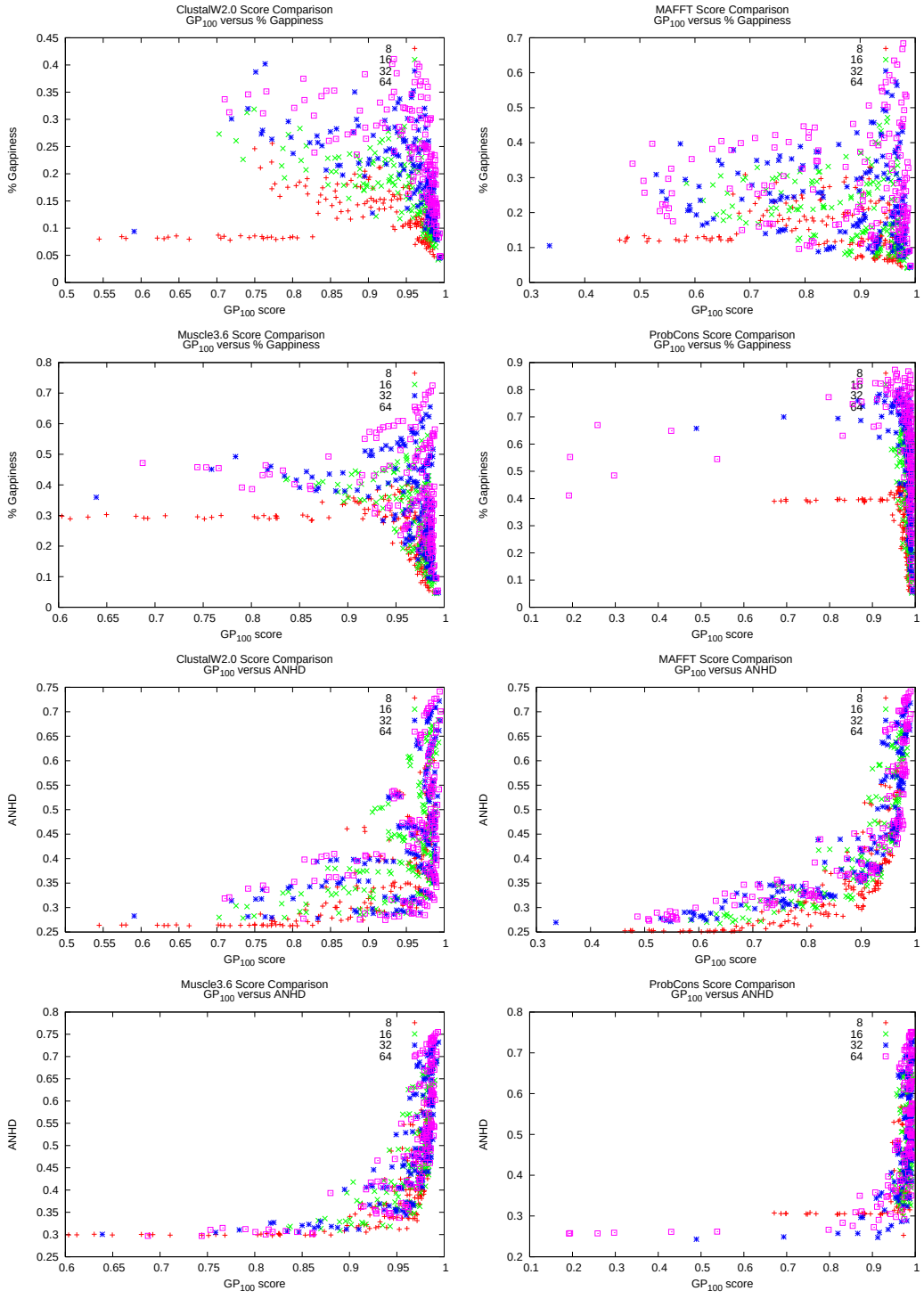

ClustalW2.0 Score Comparison
GP100 versus % Gappiness
MAFFT Score Comparison
GP100 versus % Gappiness
Muscle3.6 Score Comparison
GP100 versus % Gappiness
ProbCons Score Comparison
GP100 versus % Gappiness
ClustalW2.0 Score Comparison
GP100 versus ANHD
MAFFT Score Comparison
GP100 versus ANHD
Muscle3.6 Score Comparison
GP100 versus ANHD
ProbCons Score Comparison
GP100 versus ANHD
% Gappiness
ANHD
GP100 score
8
16
32
64

Figure B.1

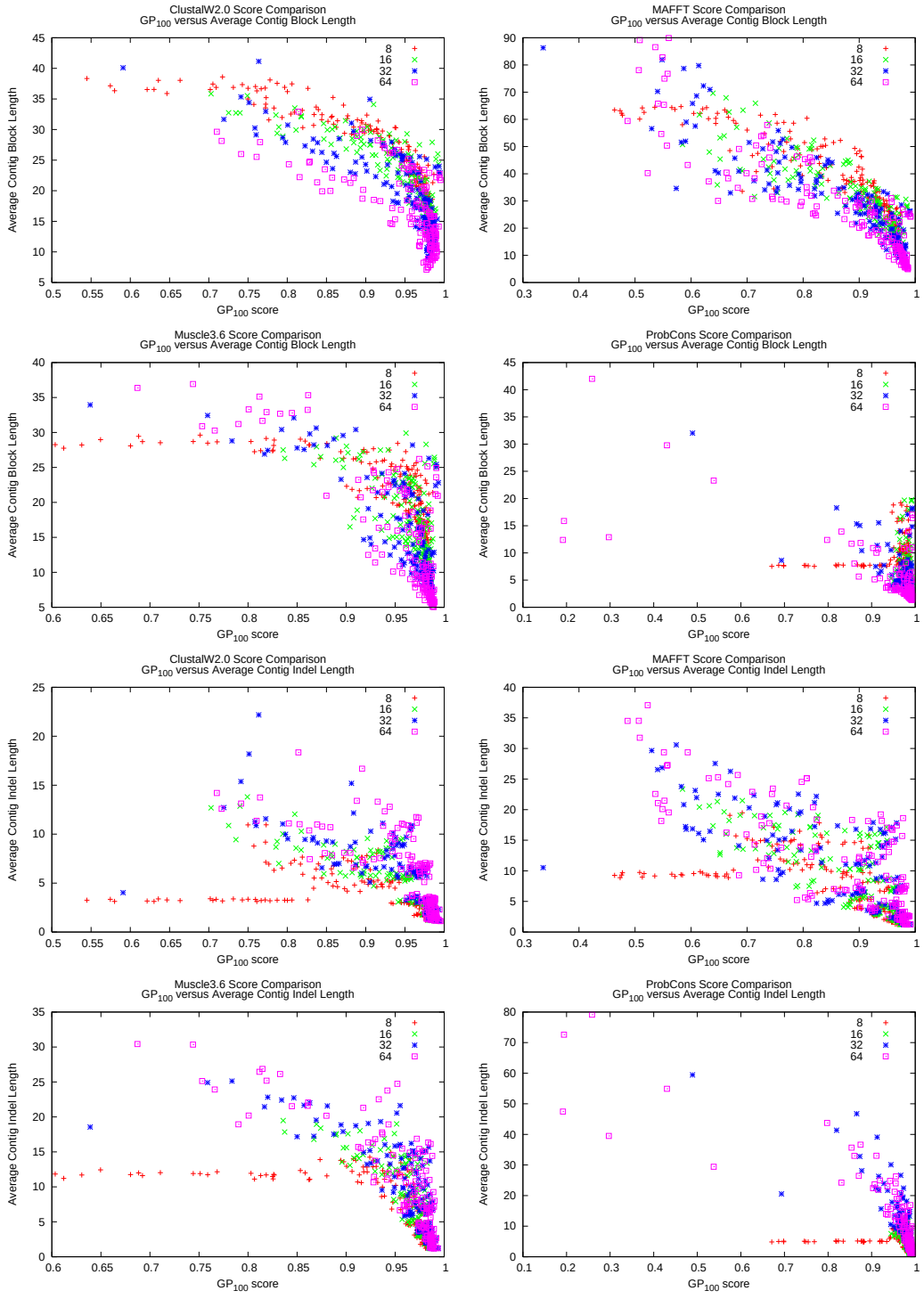

ClustalW2.0 Score Comparison
GP100 versus Average Contig Block Length
MAFFT Score Comparison
GP100 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Block Length
ProbCons Score Comparison
GP100 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP100 versus Average Contig Indel Length
MAFFT Score Comparison
GP100 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Indel Length
ProbCons Score Comparison
GP100 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP100 score
8
16
32
64

Figure B.1

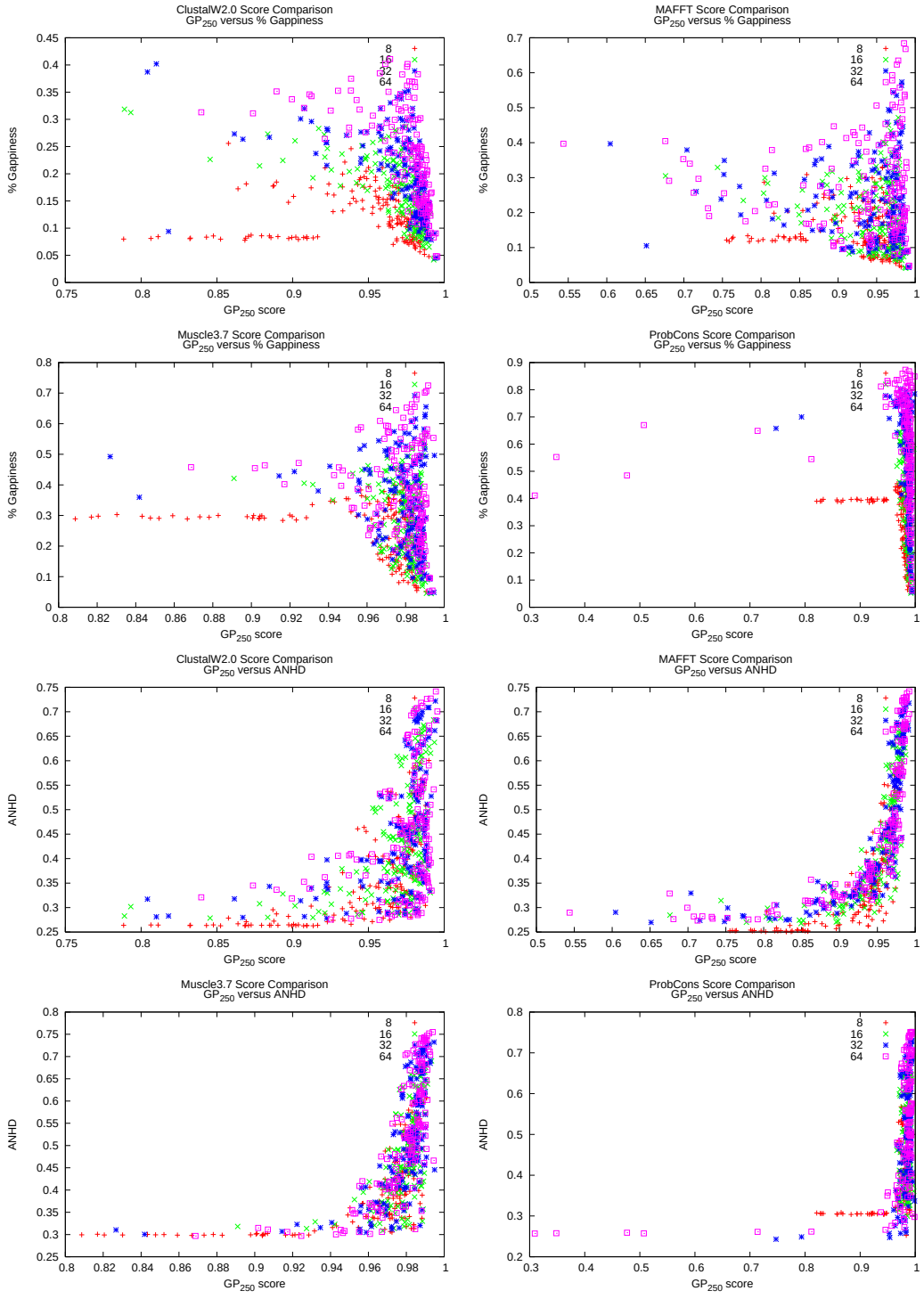

ClustalW2.0 Score Comparison
GP250 versus % Gappiness
MAFFT Score Comparison
GP250 versus % Gappiness
Muscle3.7 Score Comparison
GP250 versus % Gappiness
ProbCons Score Comparison
GP250 versus % Gappiness
ClustalW2.0 Score Comparison
GP250 versus ANHD
MAFFT Score Comparison
GP250 versus ANHD
Muscle3.7 Score Comparison
GP250 versus ANHD
ProbCons Score Comparison
GP250 versus ANHD
% Gappiness
ANHD
GP250 score
8
16
32
64

Figure B.1

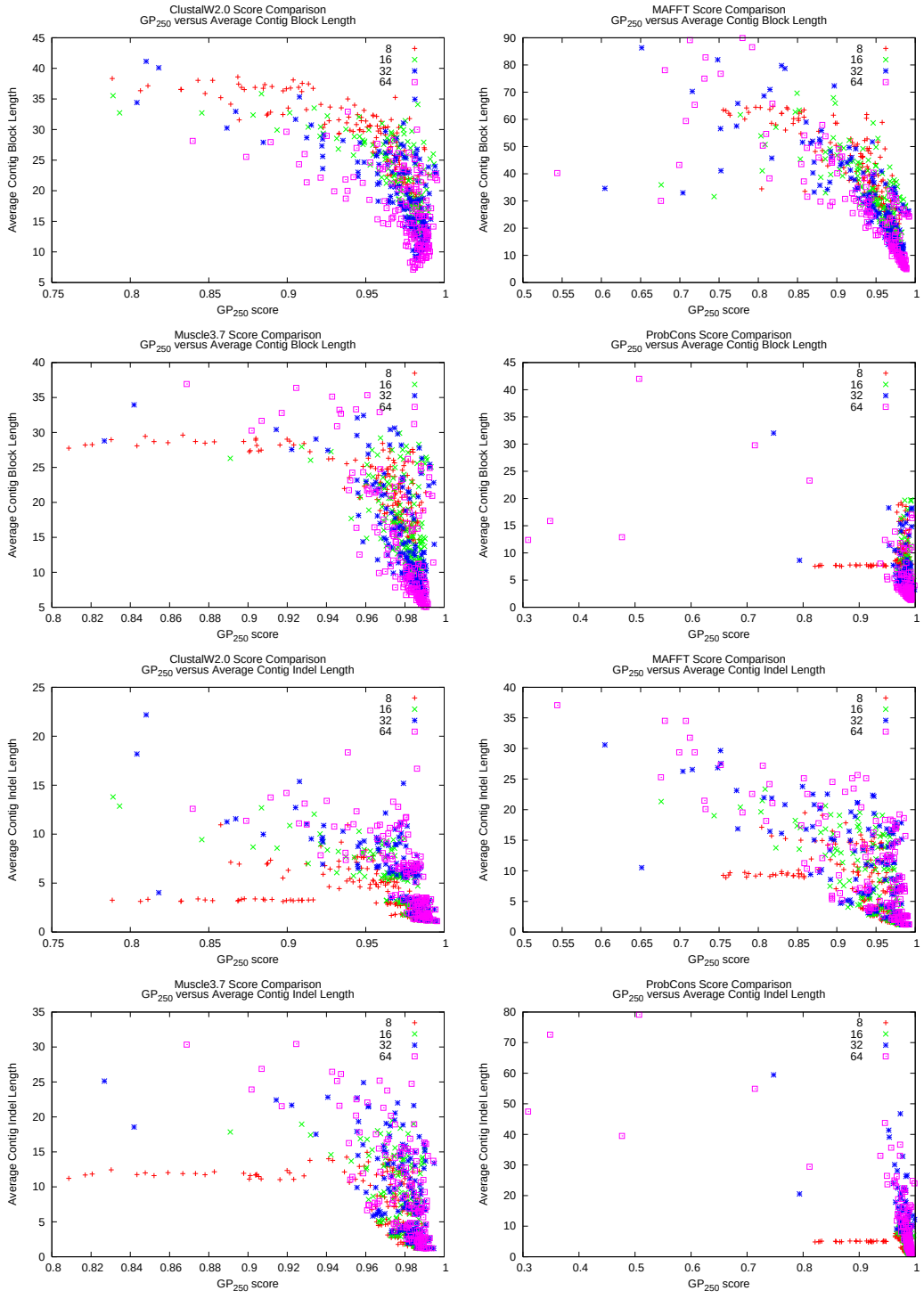

ClustalW2.0 Score Comparison
GP250 versus Average Contig Block Length
MAFFT Score Comparison
GP250 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Block Length
ProbCons Score Comparison
GP250 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP250 versus Average Contig Indel Length
MAFFT Score Comparison
GP250 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Indel Length
ProbCons Score Comparison
GP250 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP250 score
8
16
32
64

Figure B.1

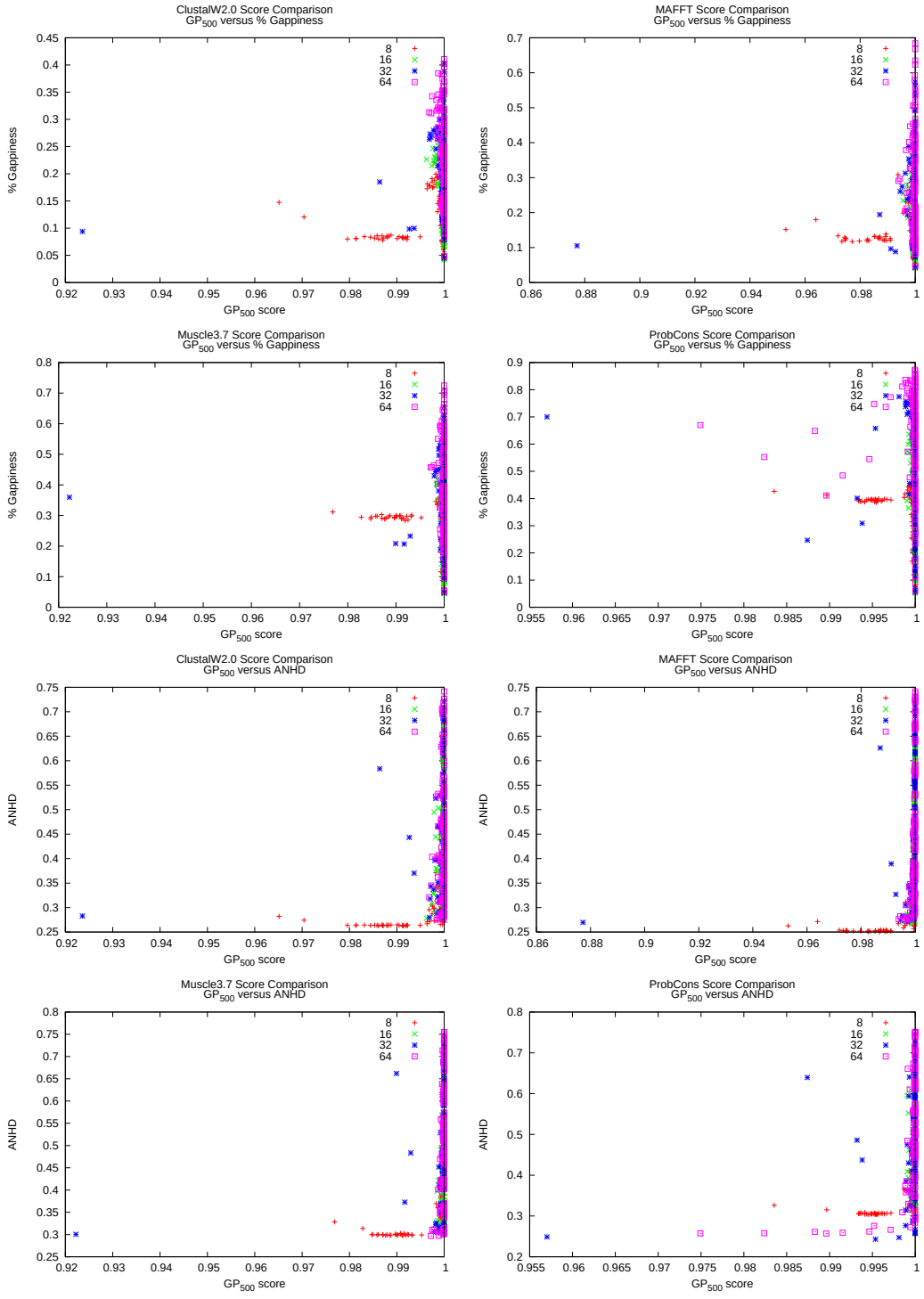

ClustalW2.0 Score Comparison
GP500 versus % Gappiness
MAFFT Score Comparison
GP500 versus % Gappiness
Muscle3.7 Score Comparison
GP500 versus % Gappiness
ProbCons Score Comparison
GP500 versus % Gappiness
ClustalW2.0 Score Comparison
GP500 versus ANHD
MAFFT Score Comparison
GP500 versus ANHD
Muscle3.7 Score Comparison
GP500 versus ANHD
ProbCons Score Comparison
GP500 versus ANHD
% Gappiness
ANHD
GP500 score
8
16
32
64

Figure B.1

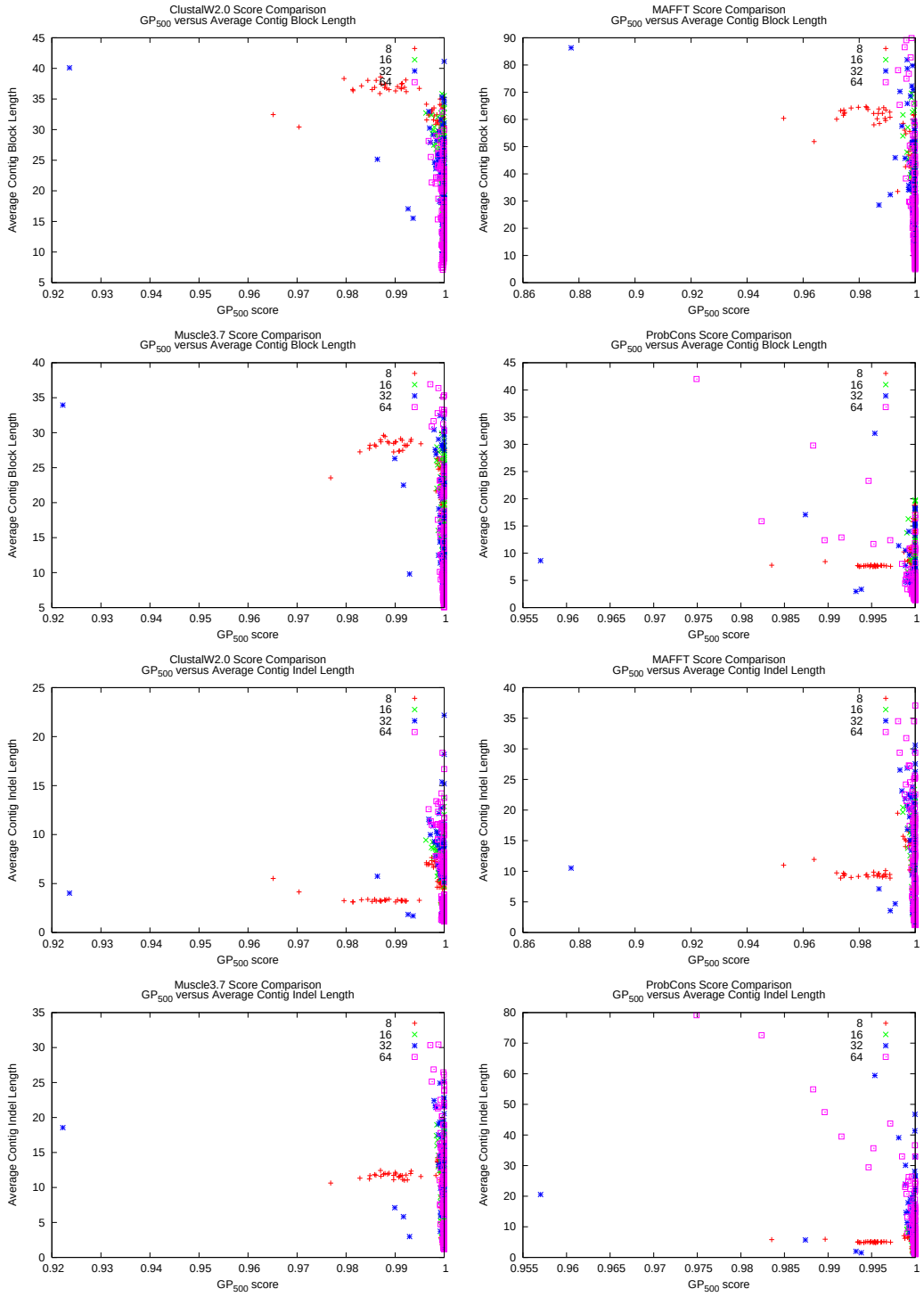

ClustalW2.0 Score Comparison
GP500 versus Average Contig Block Length
MAFFT Score Comparison
GP500 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Block Length
ProbCons Score Comparison
GP500 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP500 versus Average Contig Indel Length
MAFFT Score Comparison
GP500 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Indel Length
ProbCons Score Comparison
GP500 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP500 score
8
16
32
64

Figure B.1

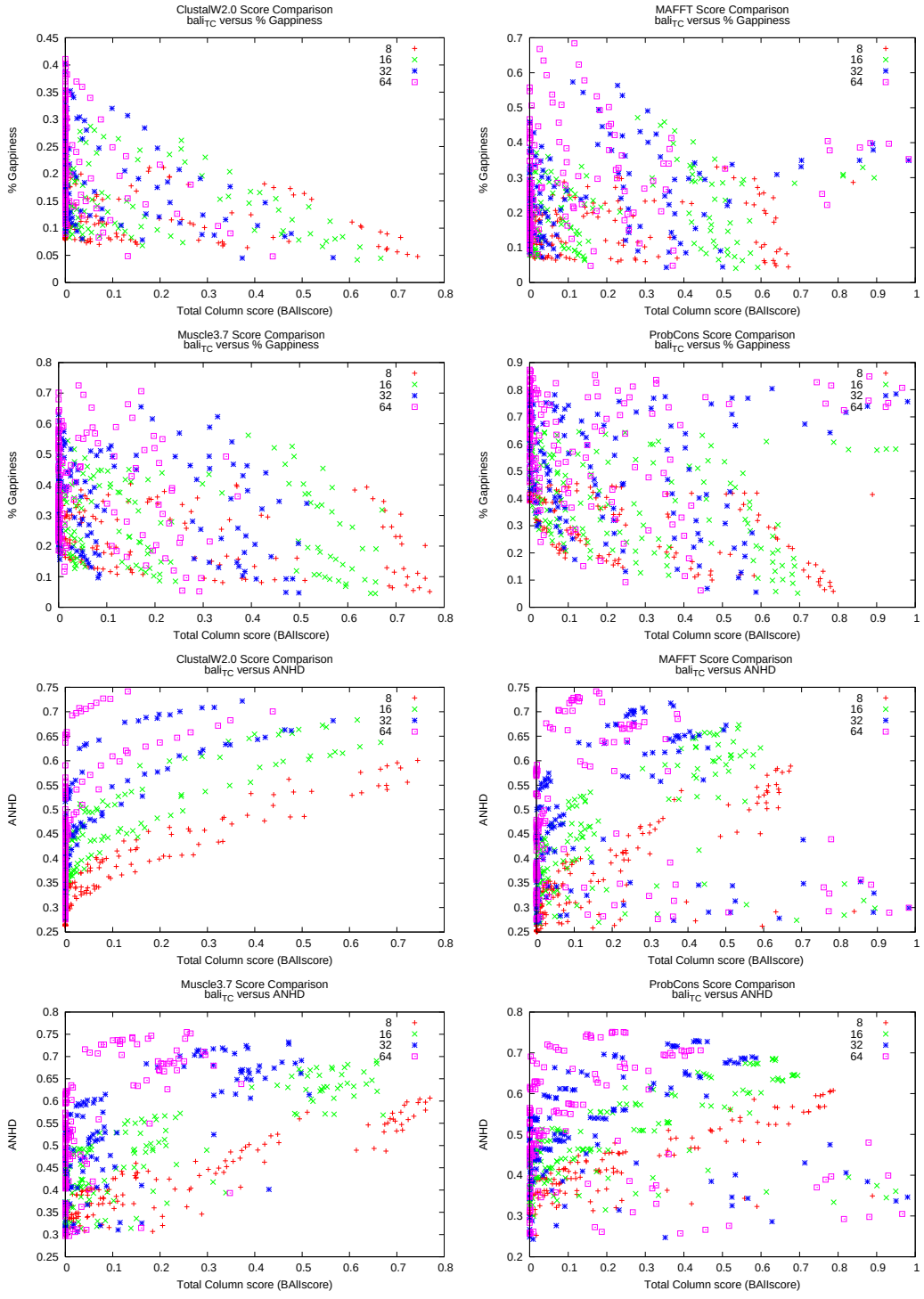

ClustalW2.0 Score Comparison
bali_TC versus % Gappiness
MAFFT Score Comparison
bali_TC versus % Gappiness
Muscle3.7 Score Comparison
bali_TC versus % Gappiness
ProbCons Score Comparison
bali_TC versus % Gappiness
ClustalW2.0 Score Comparison
bali_TC versus ANHD
MAFFT Score Comparison
bali_TC versus ANHD
Muscle3.7 Score Comparison
bali_TC versus ANHD
ProbCons Score Comparison
bali_TC versus ANHD
% Gappiness
ANHD
Total Column score (BAIIscore)
8
16
32
64

Figure B.1

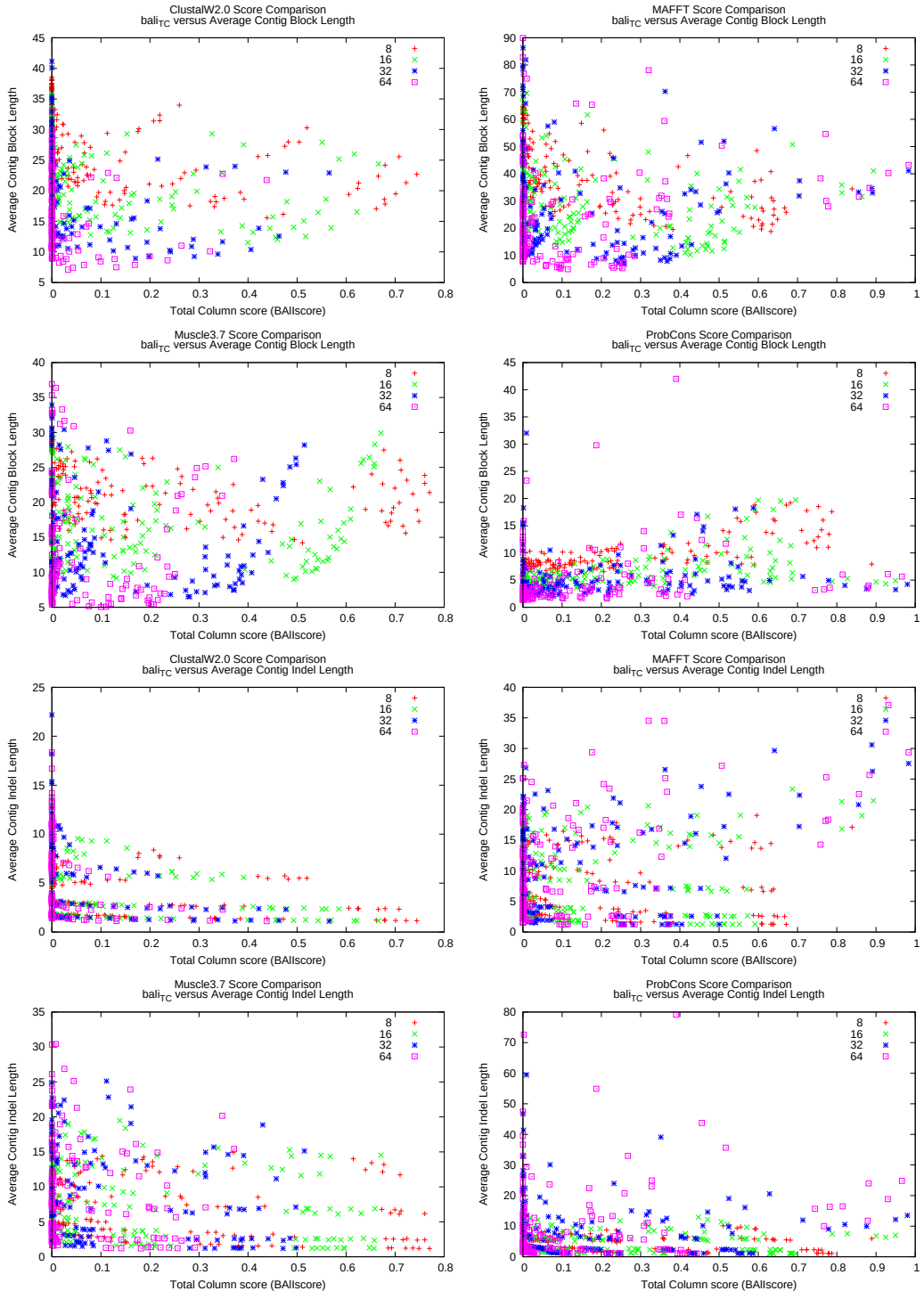

ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Block Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Total Column score (BAIIscore)
8
16
32
64

Figure B.1

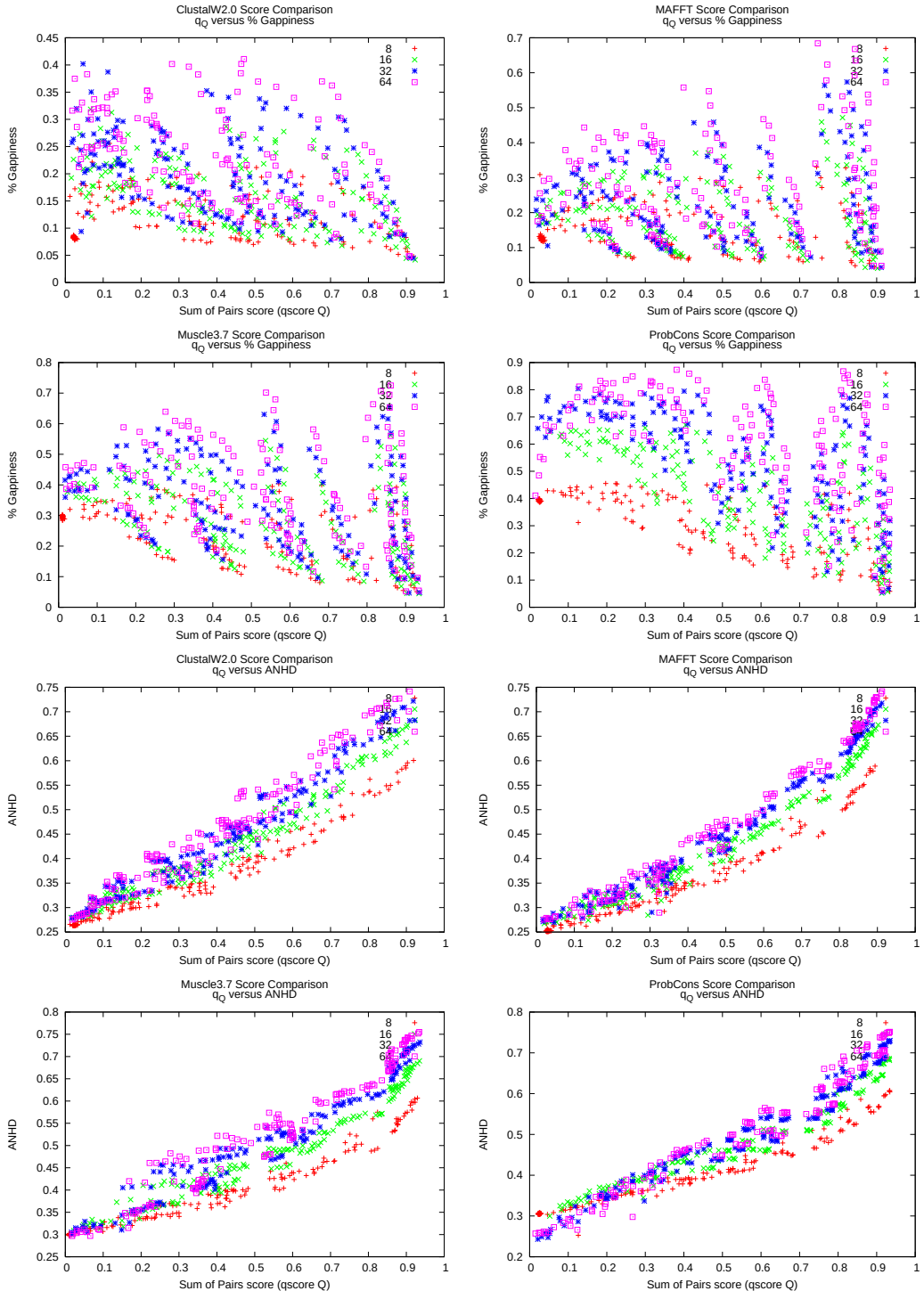

ClustalW2.0 Score Comparison
$q_Q$ versus % Gappiness
MAFFT Score Comparison
$q_Q$ versus % Gappiness
Muscle3.7 Score Comparison
$q_Q$ versus % Gappiness
ProbCons Score Comparison
$q_Q$ versus % Gappiness
ClustalW2.0 Score Comparison
$q_Q$ versus ANHD
MAFFT Score Comparison
$q_Q$ versus ANHD
Muscle3.7 Score Comparison
$q_Q$ versus ANHD
ProbCons Score Comparison
$q_Q$ versus ANHD
% Gappiness
ANHD
Sum of Pairs score (qscore Q)
8
16
32
64

Figure B.1

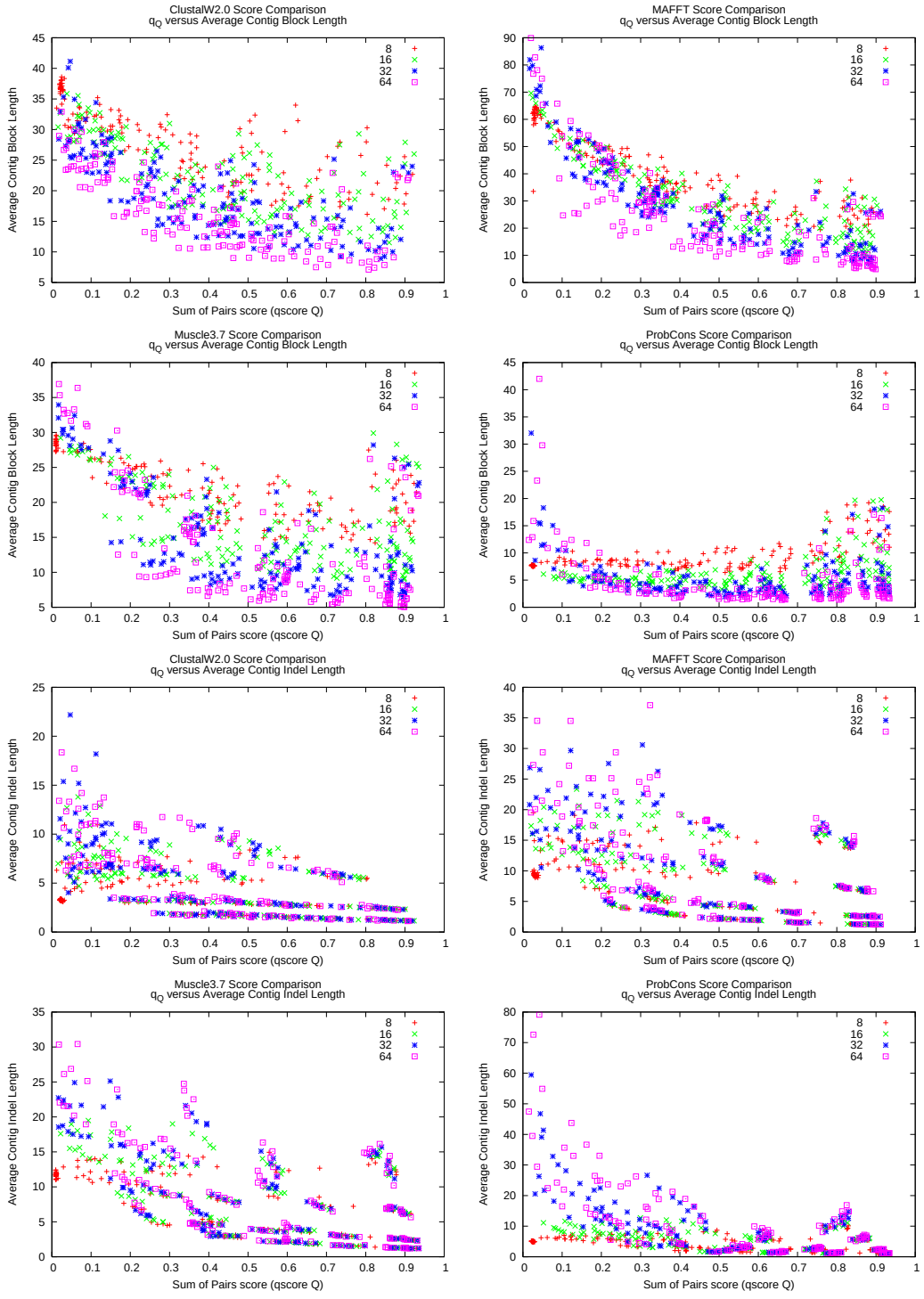

ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Block Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Block Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Indel Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Indel Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Sum of Pairs score (qscore Q)
8
16
32
64

Figure B.1

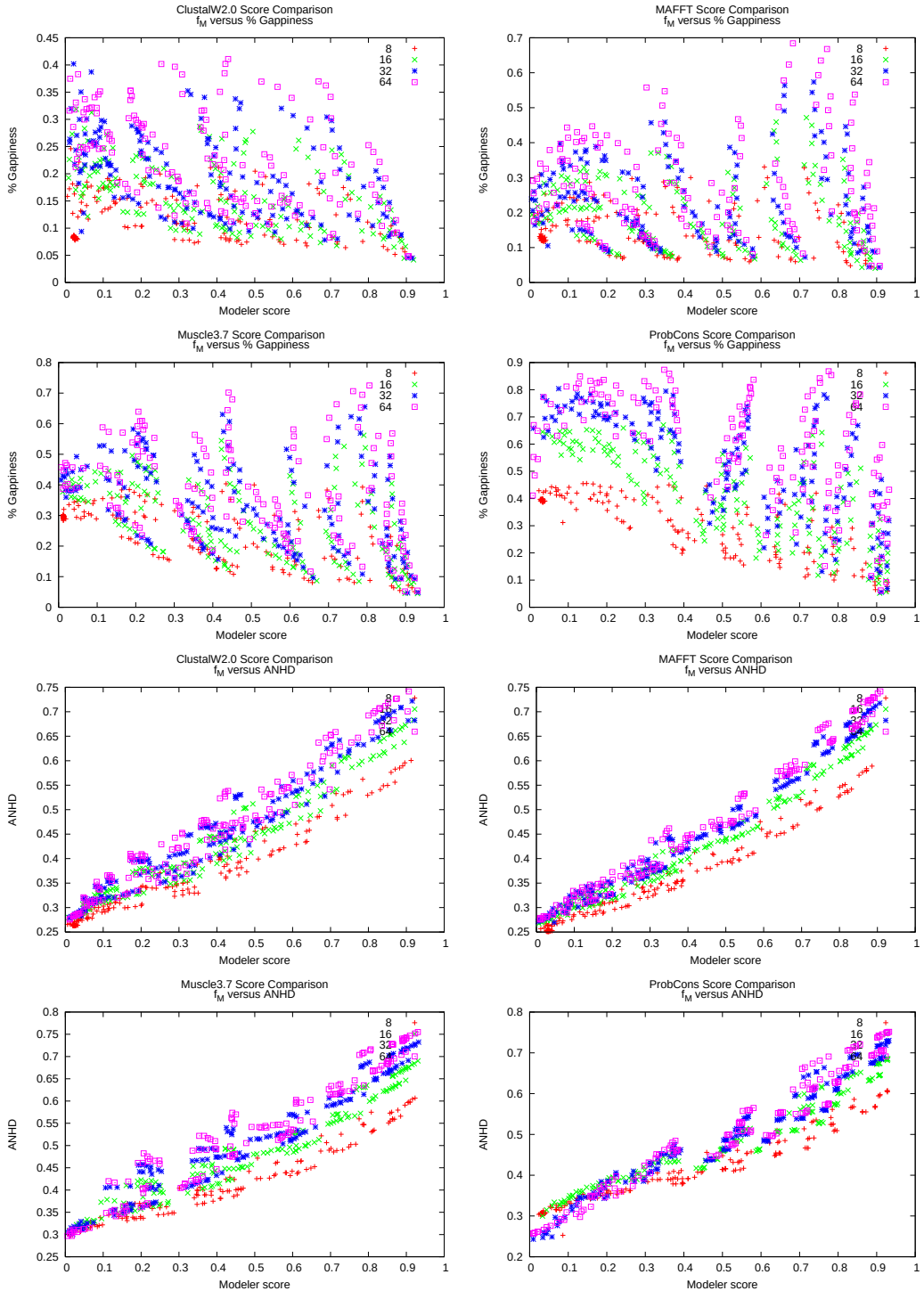

ClustalW2.0 Score Comparison
$f_M$ versus % Gappiness
MAFFT Score Comparison
$f_M$ versus % Gappiness
Muscle3.7 Score Comparison
$f_M$ versus % Gappiness
ProbCons Score Comparison
$f_M$ versus % Gappiness
ClustalW2.0 Score Comparison
$f_M$ versus ANHD
MAFFT Score Comparison
$f_M$ versus ANHD
Muscle3.7 Score Comparison
$f_M$ versus ANHD
ProbCons Score Comparison
$f_M$ versus ANHD
% Gappiness
ANHD
Modeler score
8
16
32
64

Figure B.1

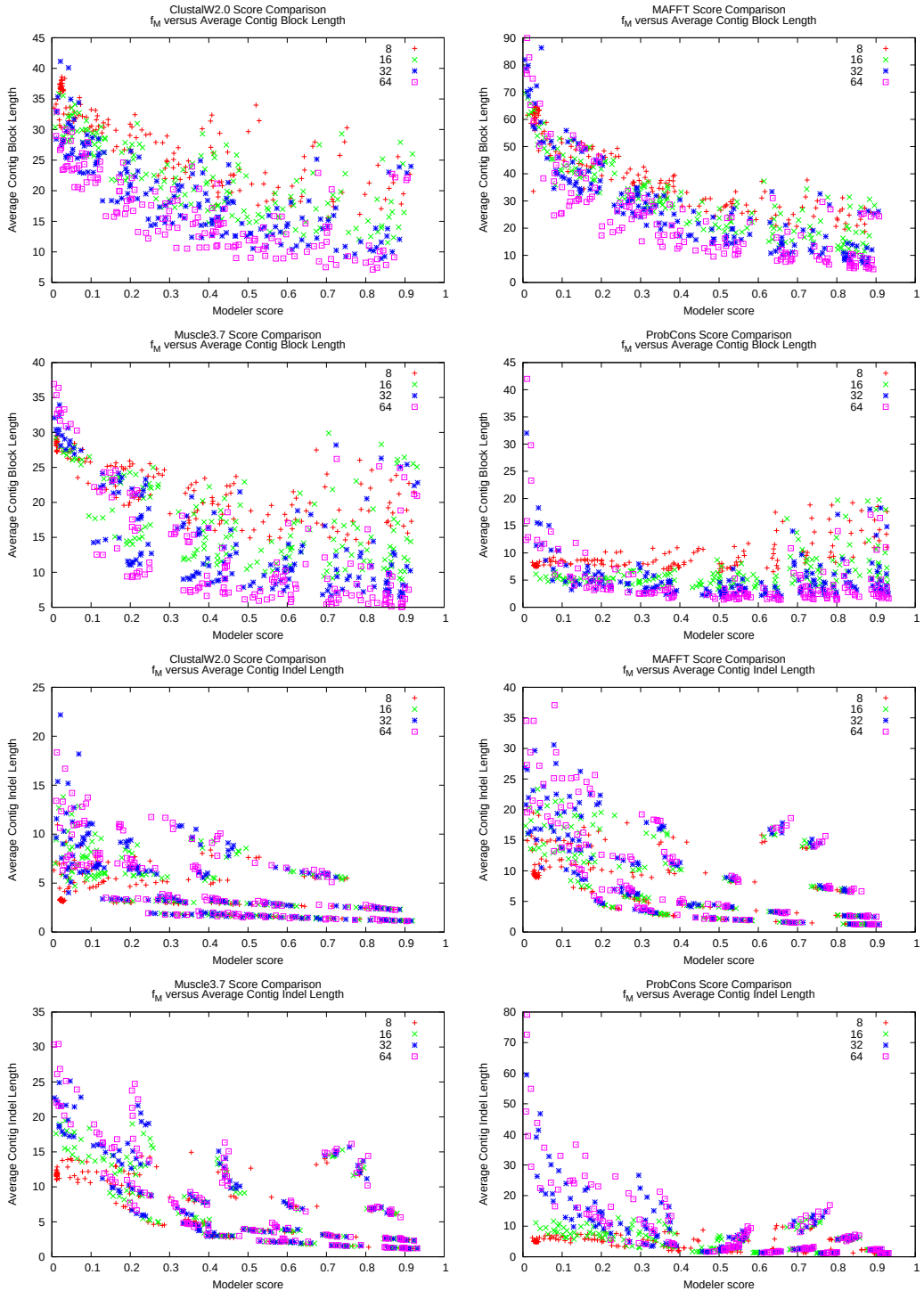

ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Block Length
MAFFT Score Comparison
$f_M$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Block Length
ProbCons Score Comparison
$f_M$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Indel Length
MAFFT Score Comparison
$f_M$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Indel Length
ProbCons Score Comparison
$f_M$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Modeler score
8
16
32
64

Figure B.1

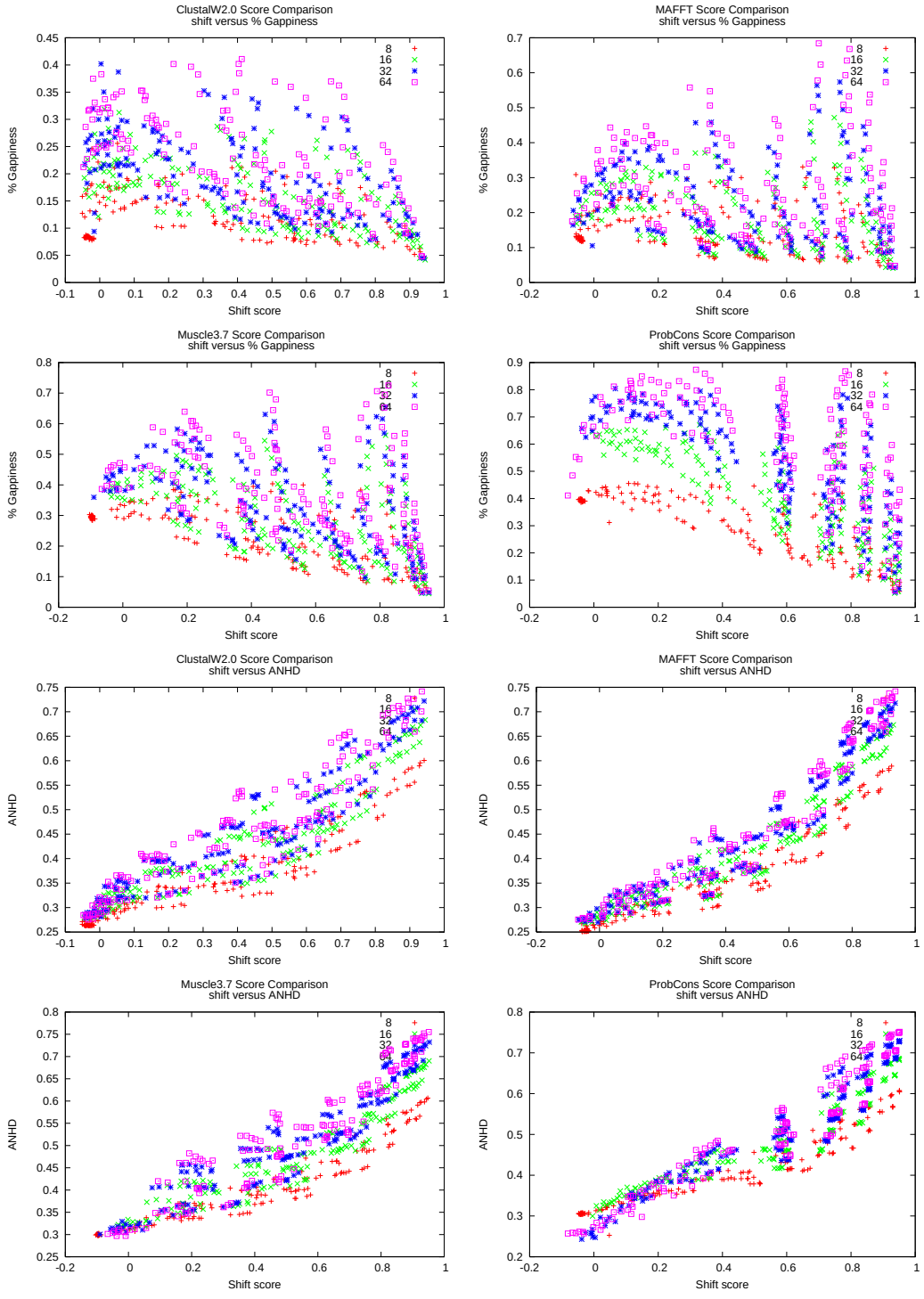

ClustalW2.0 Score Comparison
shift versus % Gappiness
MAFFT Score Comparison
shift versus % Gappiness
Muscle3.7 Score Comparison
shift versus % Gappiness
ProbCons Score Comparison
shift versus % Gappiness
ClustalW2.0 Score Comparison
shift versus ANHD
MAFFT Score Comparison
shift versus ANHD
Muscle3.7 Score Comparison
shift versus ANHD
ProbCons Score Comparison
shift versus ANHD
% Gappiness
ANHD
Shift score
8
16
32
64

Figure B.1

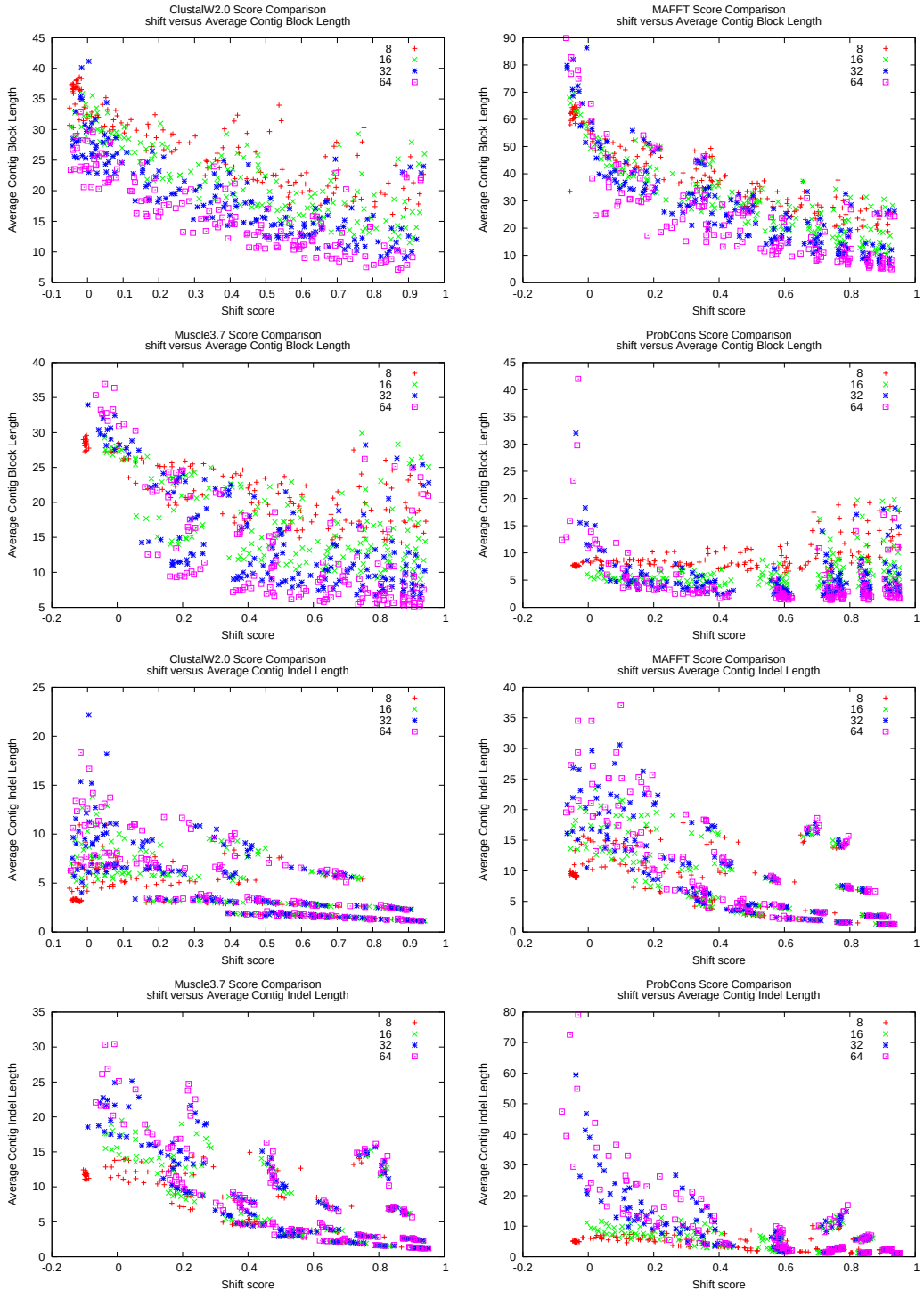

ClustalW2.0 Score Comparison
shift versus Average Contig Block Length
MAFFT Score Comparison
shift versus Average Contig Block Length
Muscle3.7 Score Comparison
shift versus Average Contig Block Length
ProbCons Score Comparison
shift versus Average Contig Block Length
ClustalW2.0 Score Comparison
shift versus Average Contig Indel Length
MAFFT Score Comparison
shift versus Average Contig Indel Length
Muscle3.7 Score Comparison
shift versus Average Contig Indel Length
ProbCons Score Comparison
shift versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Shift score
8
16
32
64

Figure B.1

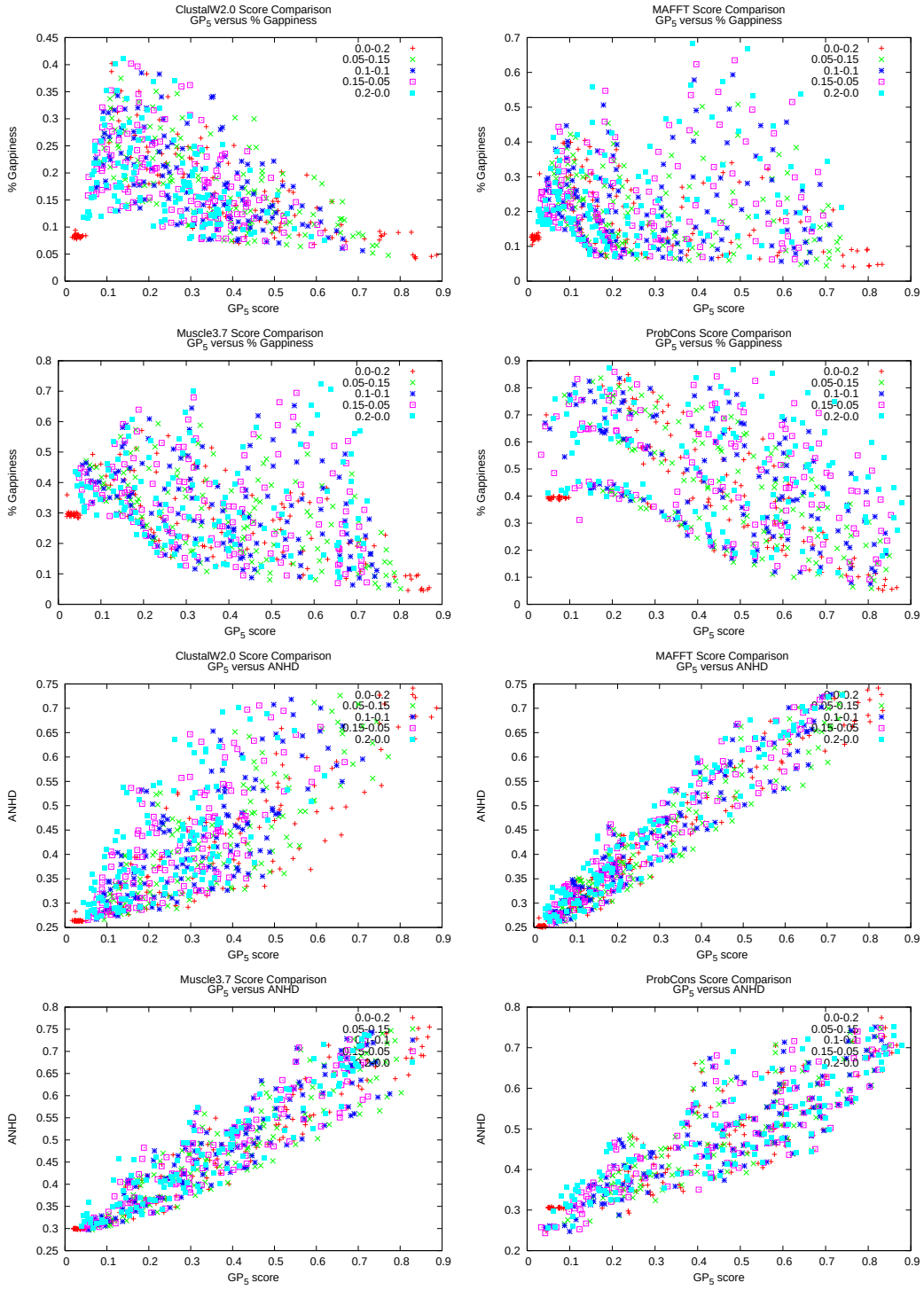

ClustalW2.0 Score Comparison
GP5 versus % Gappiness
MAFFT Score Comparison
GP5 versus % Gappiness
Muscle3.7 Score Comparison
GP5 versus % Gappiness
ProbCons Score Comparison
GP5 versus % Gappiness
ClustalW2.0 Score Comparison
GP5 versus ANHD
MAFFT Score Comparison
GP5 versus ANHD
Muscle3.7 Score Comparison
GP5 versus ANHD
ProbCons Score Comparison
GP5 versus ANHD
% Gappiness
ANHD
GP5 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

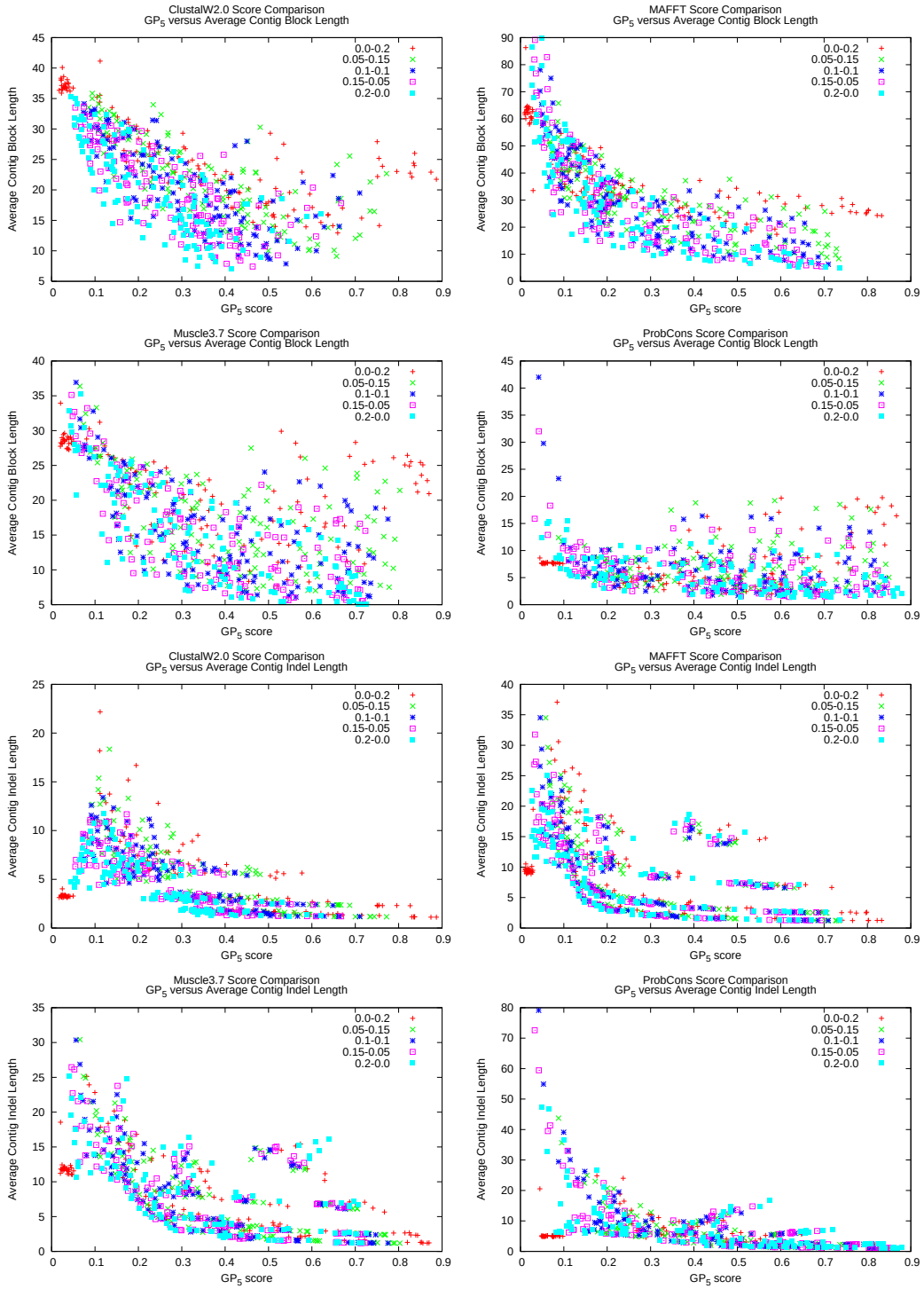

ClustalW2.0 Score Comparison
GP5 versus Average Contig Block Length
MAFFT Score Comparison
GP5 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Block Length
ProbCons Score Comparison
GP5 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP5 versus Average Contig Indel Length
MAFFT Score Comparison
GP5 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Indel Length
ProbCons Score Comparison
GP5 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP5 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

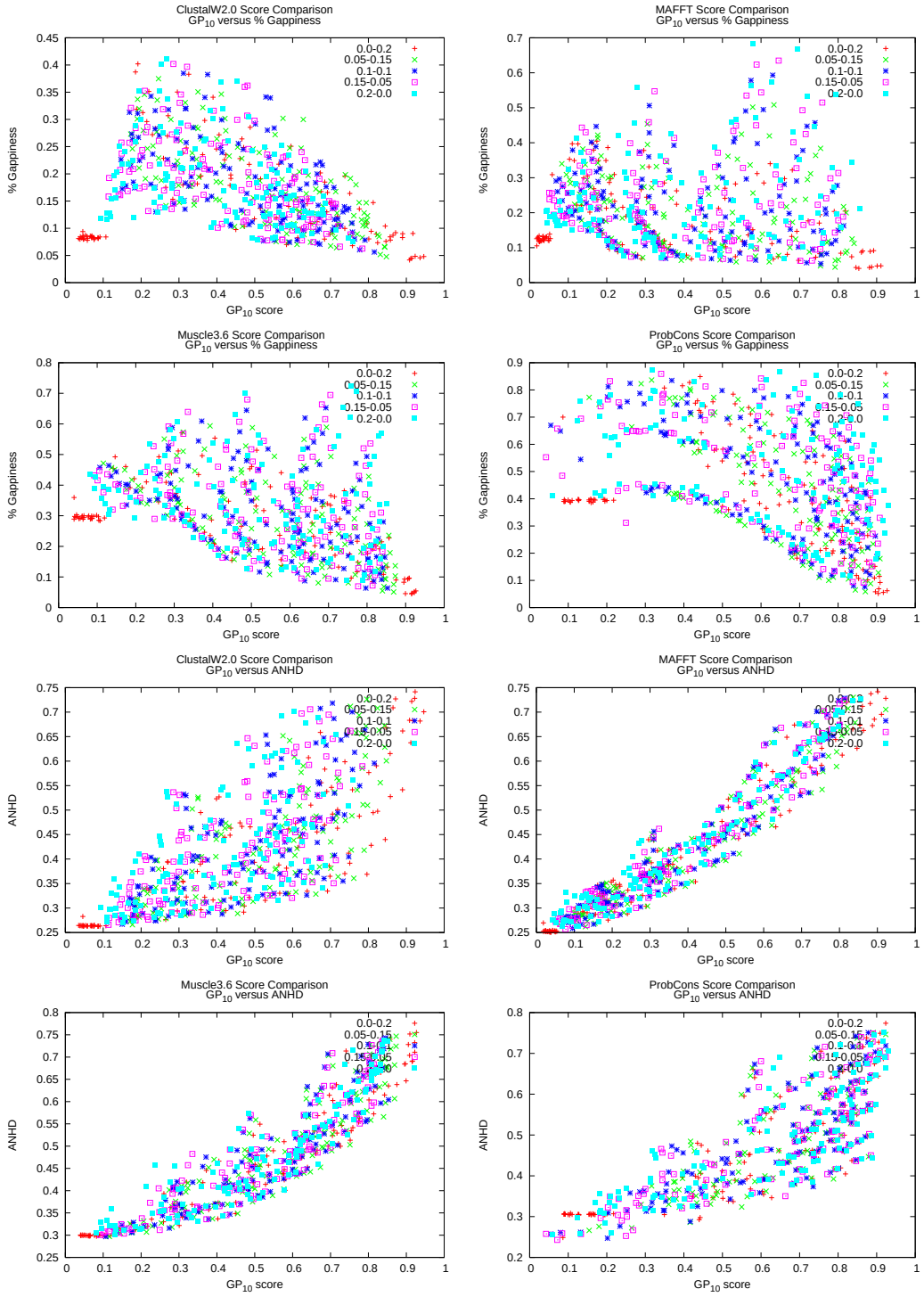

ClustalW2.0 Score Comparison
GP10 versus % Gappiness
MAFFT Score Comparison
GP10 versus % Gappiness
Muscle3.6 Score Comparison
GP10 versus % Gappiness
ProbCons Score Comparison
GP10 versus % Gappiness
ClustalW2.0 Score Comparison
GP10 versus ANHD
MAFFT Score Comparison
GP10 versus ANHD
Muscle3.6 Score Comparison
GP10 versus ANHD
ProbCons Score Comparison
GP10 versus ANHD
% Gappiness
ANHD
GP10 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

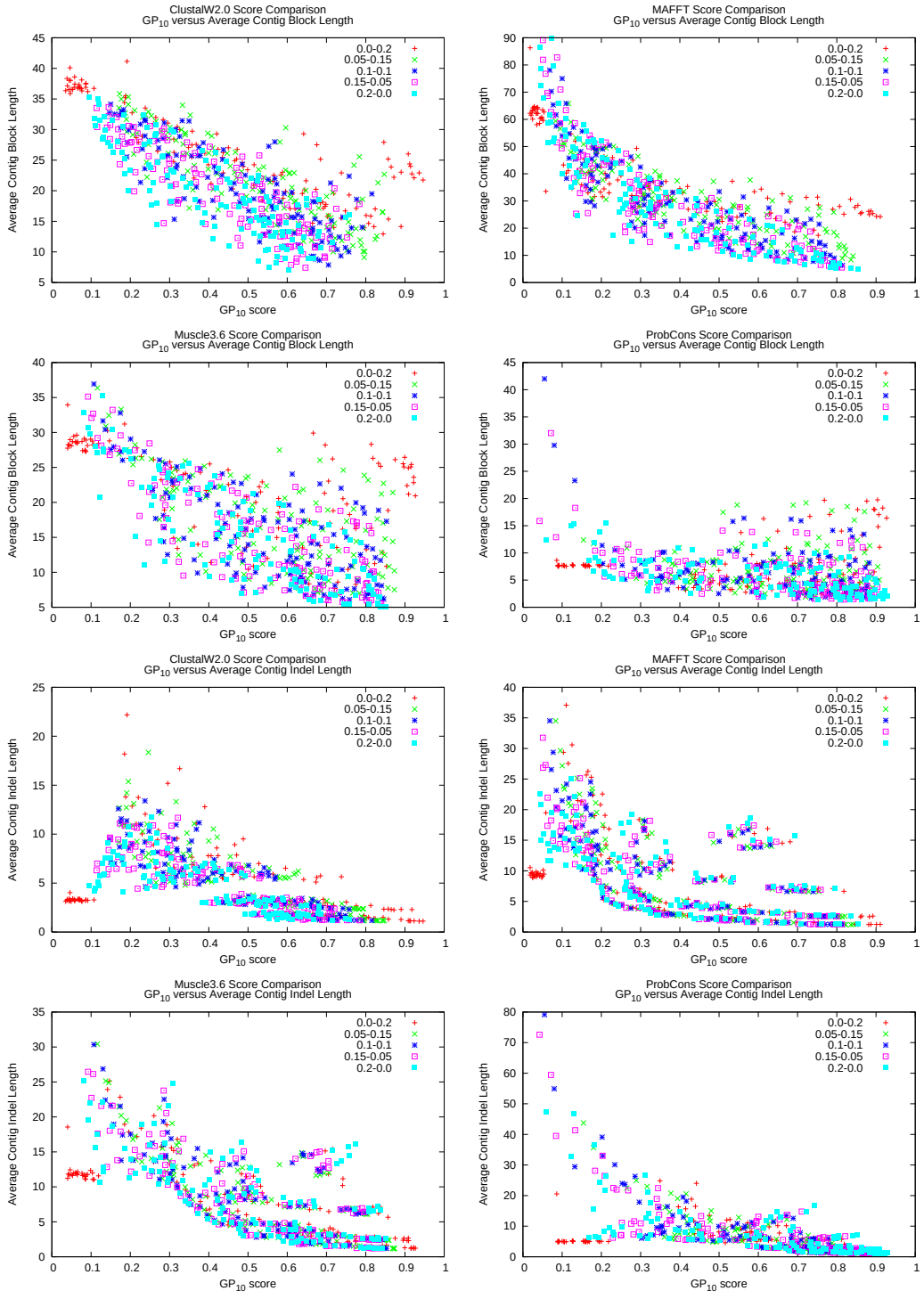

ClustalW2.0 Score Comparison
GP10 versus Average Contig Block Length
MAFFT Score Comparison
GP10 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Block Length
ProbCons Score Comparison
GP10 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP10 versus Average Contig Indel Length
MAFFT Score Comparison
GP10 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Indel Length
ProbCons Score Comparison
GP10 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP10 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

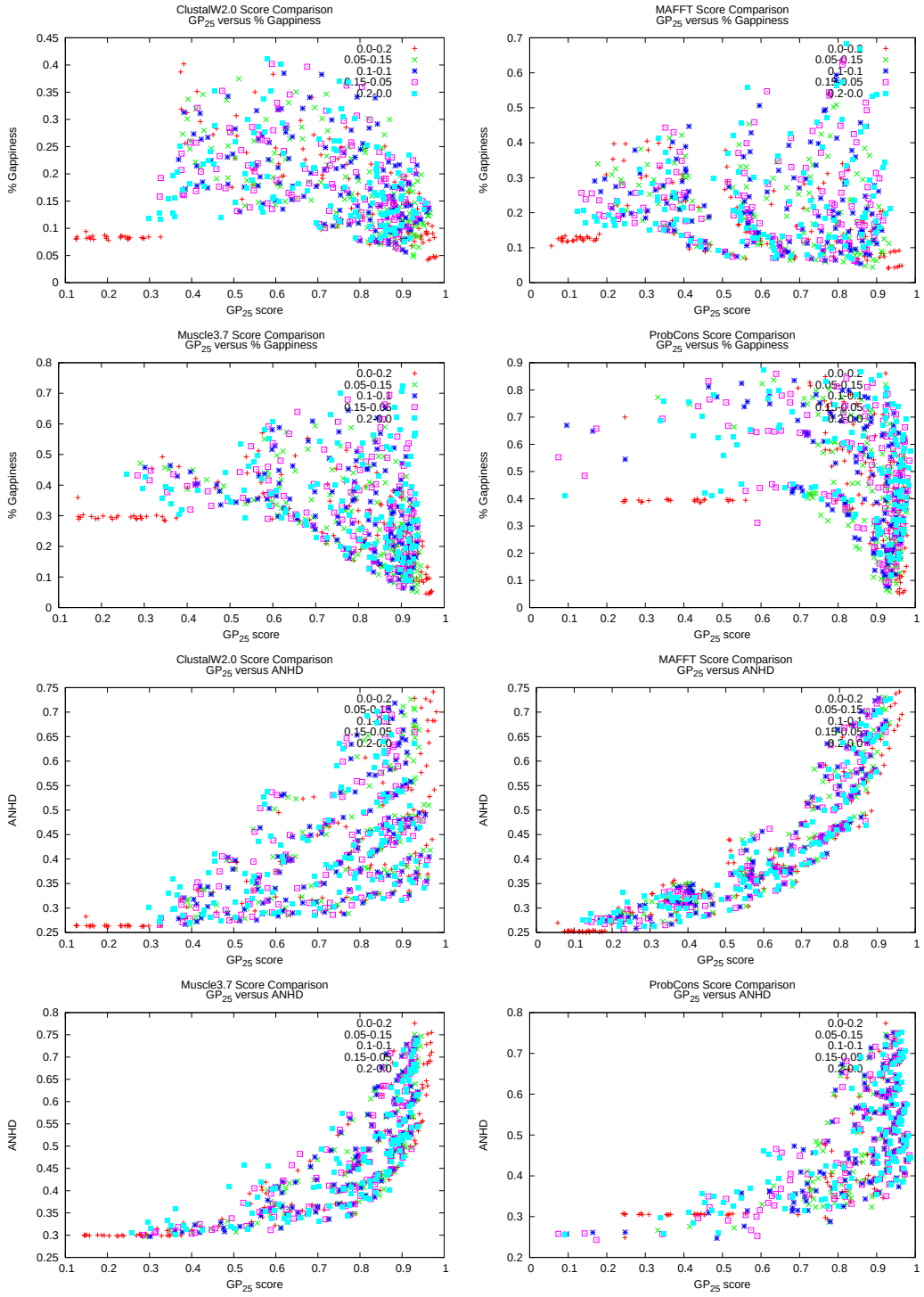

ClustalW2.0 Score Comparison
GP25 versus % Gappiness
MAFFT Score Comparison
GP25 versus % Gappiness
Muscle3.7 Score Comparison
GP25 versus % Gappiness
ProbCons Score Comparison
GP25 versus % Gappiness
ClustalW2.0 Score Comparison
GP25 versus ANHD
MAFFT Score Comparison
GP25 versus ANHD
Muscle3.7 Score Comparison
GP25 versus ANHD
ProbCons Score Comparison
GP25 versus ANHD
% Gappiness
ANHD
GP25 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

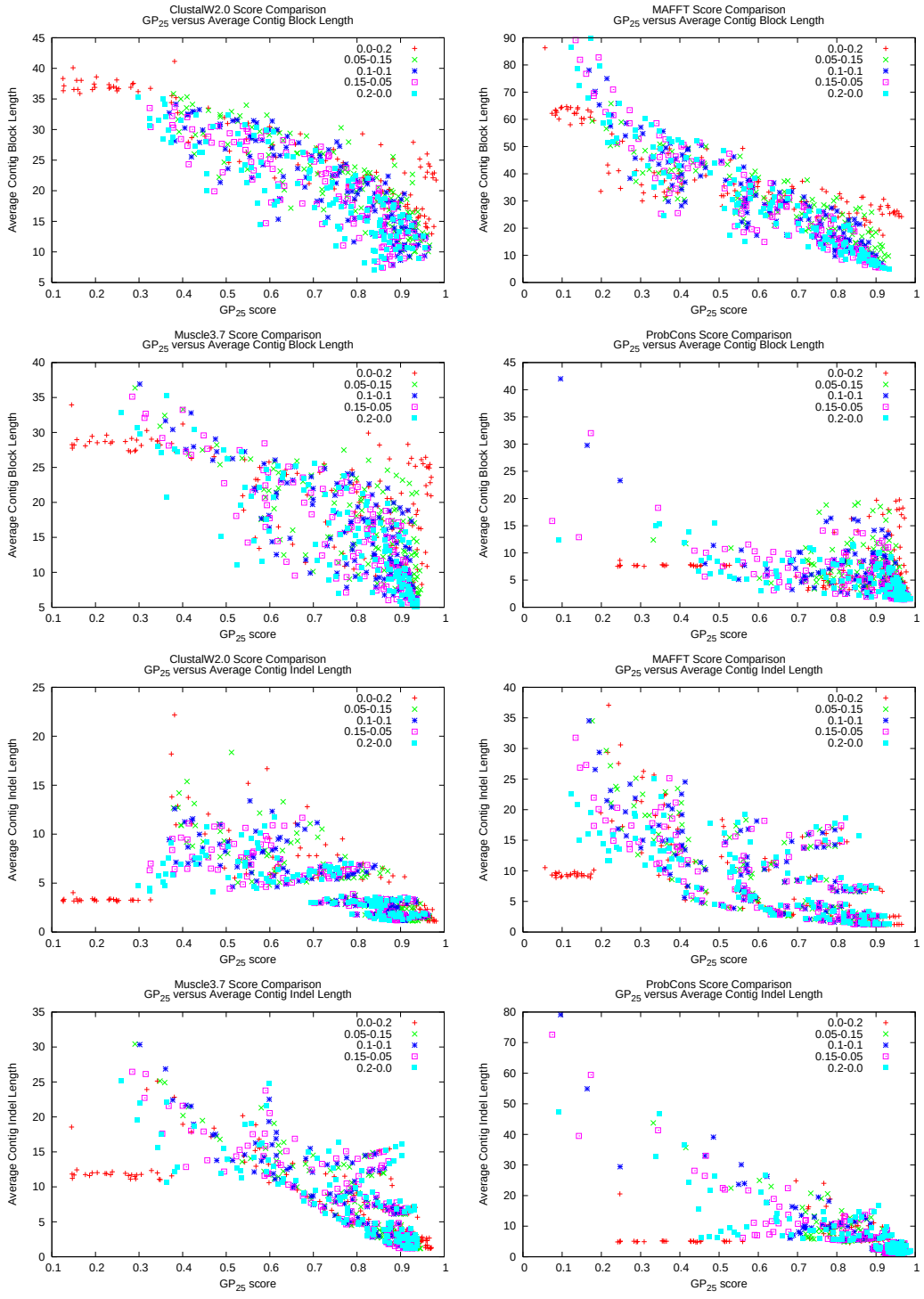

ClustalW2.0 Score Comparison
GP25 versus Average Contig Block Length
MAFFT Score Comparison
GP25 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Block Length
ProbCons Score Comparison
GP25 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP25 versus Average Contig Indel Length
MAFFT Score Comparison
GP25 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Indel Length
ProbCons Score Comparison
GP25 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP25 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

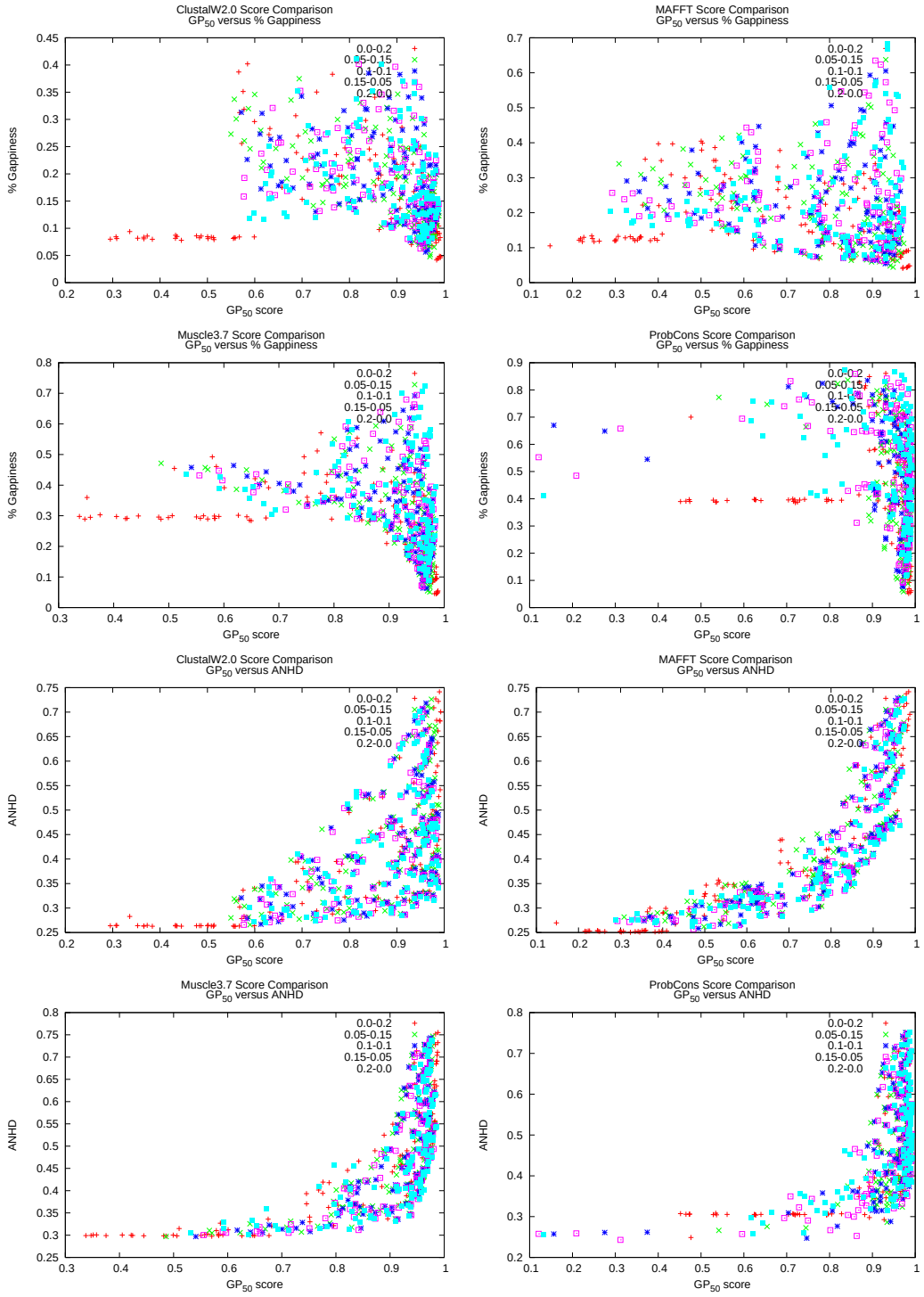

ClustalW2.0 Score Comparison
GP50 versus % Gappiness
MAFFT Score Comparison
GP50 versus % Gappiness
Muscle3.7 Score Comparison
GP50 versus % Gappiness
ProbCons Score Comparison
GP50 versus % Gappiness
ClustalW2.0 Score Comparison
GP50 versus ANHD
MAFFT Score Comparison
GP50 versus ANHD
Muscle3.7 Score Comparison
GP50 versus ANHD
ProbCons Score Comparison
GP50 versus ANHD
% Gappiness
ANHD
GP50 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

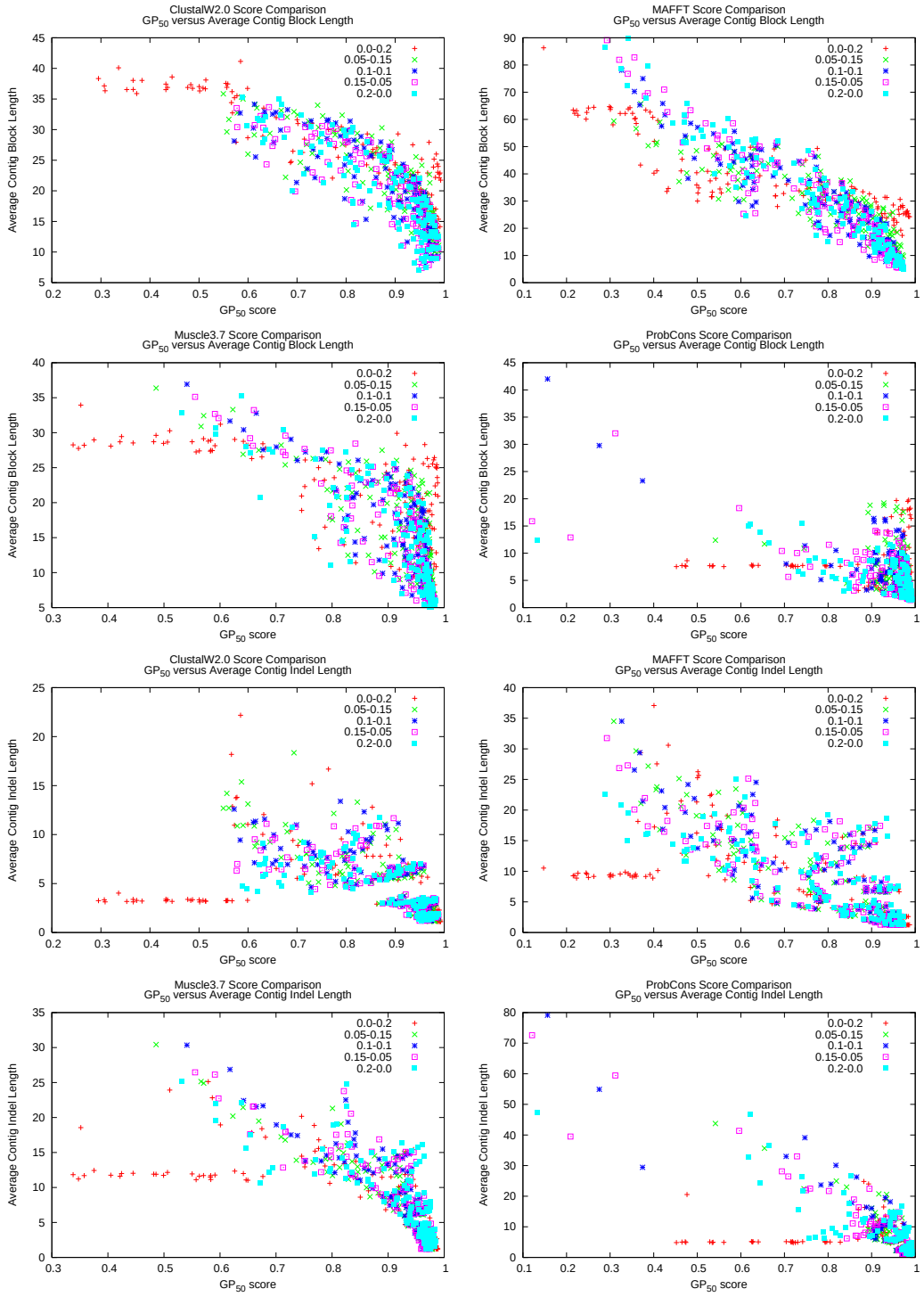

ClustalW2.0 Score Comparison
GP50 versus Average Contig Block Length
MAFFT Score Comparison
GP50 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Block Length
ProbCons Score Comparison
GP50 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP50 versus Average Contig Indel Length
MAFFT Score Comparison
GP50 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Indel Length
ProbCons Score Comparison
GP50 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP50 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

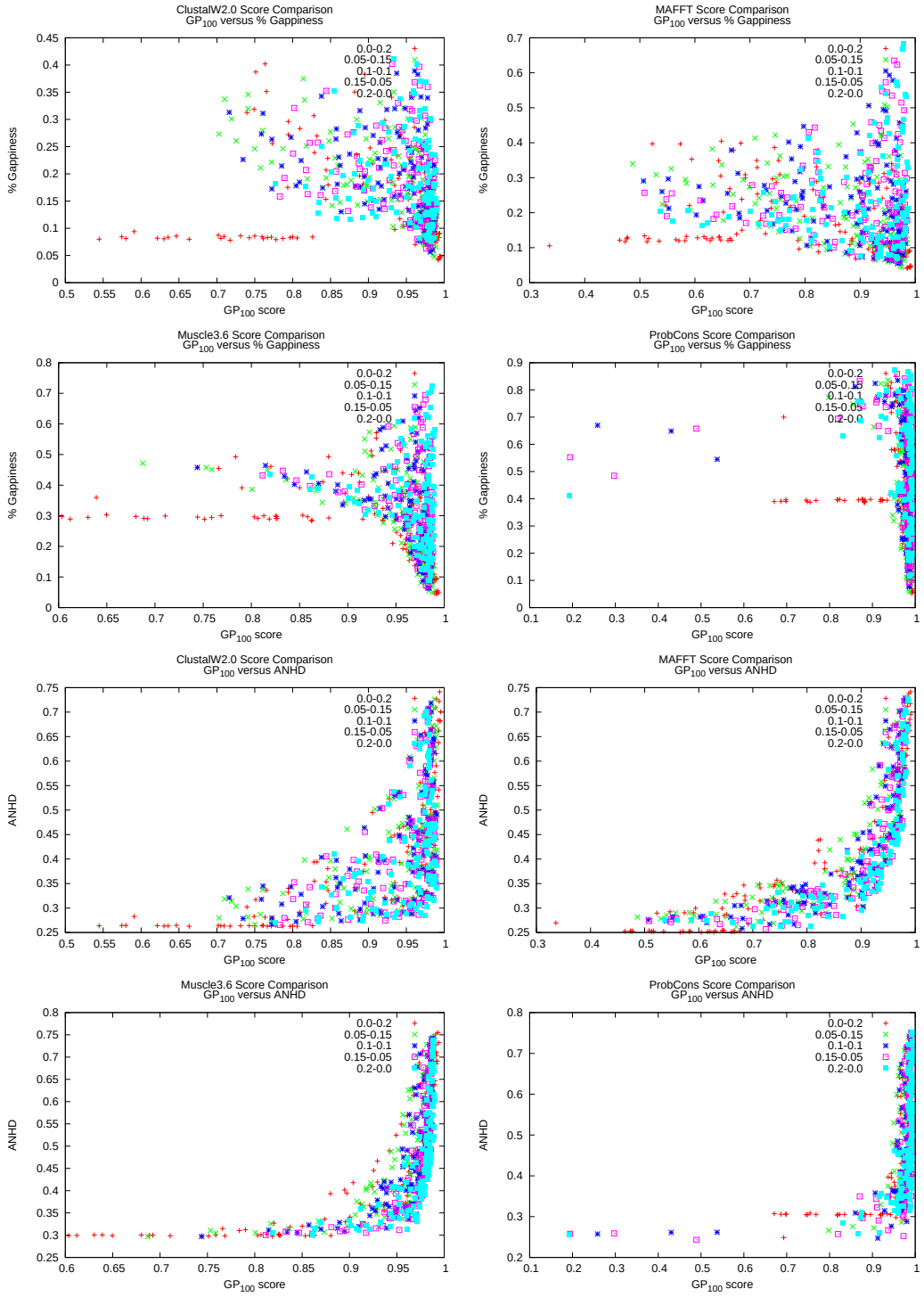

ClustalW2.0 Score Comparison
GP100 versus % Gappiness
MAFFT Score Comparison
GP100 versus % Gappiness
Muscle3.6 Score Comparison
GP100 versus % Gappiness
ProbCons Score Comparison
GP100 versus % Gappiness
ClustalW2.0 Score Comparison
GP100 versus ANHD
MAFFT Score Comparison
GP100 versus ANHD
Muscle3.6 Score Comparison
GP100 versus ANHD
ProbCons Score Comparison
GP100 versus ANHD
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
% Gappiness
ANHD
GP100 score

Figure B.1

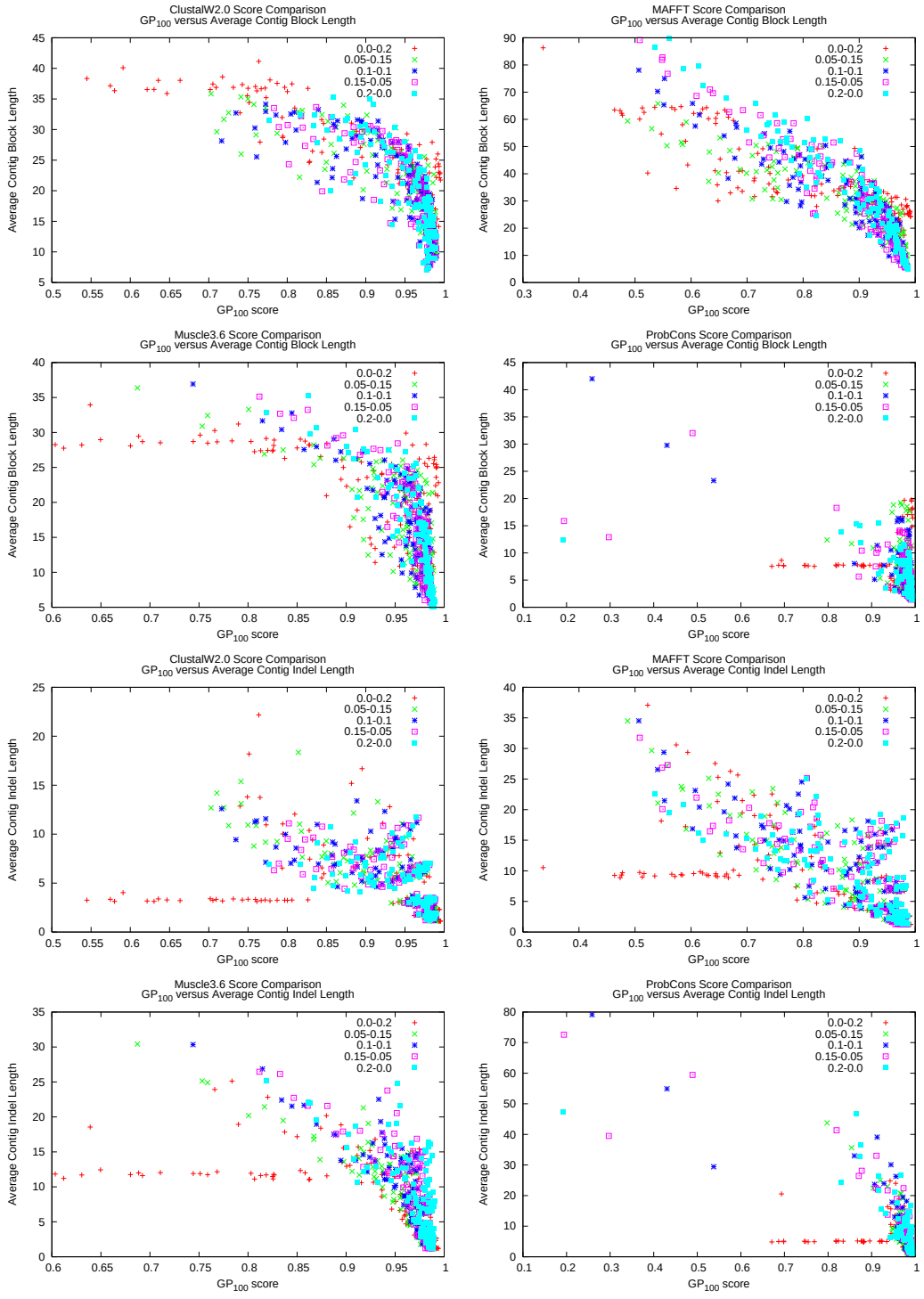

ClustalW2.0 Score Comparison
GP100 versus Average Contig Block Length
MAFFT Score Comparison
GP100 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Block Length
ProbCons Score Comparison
GP100 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP100 versus Average Contig Indel Length
MAFFT Score Comparison
GP100 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Indel Length
ProbCons Score Comparison
GP100 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP100 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

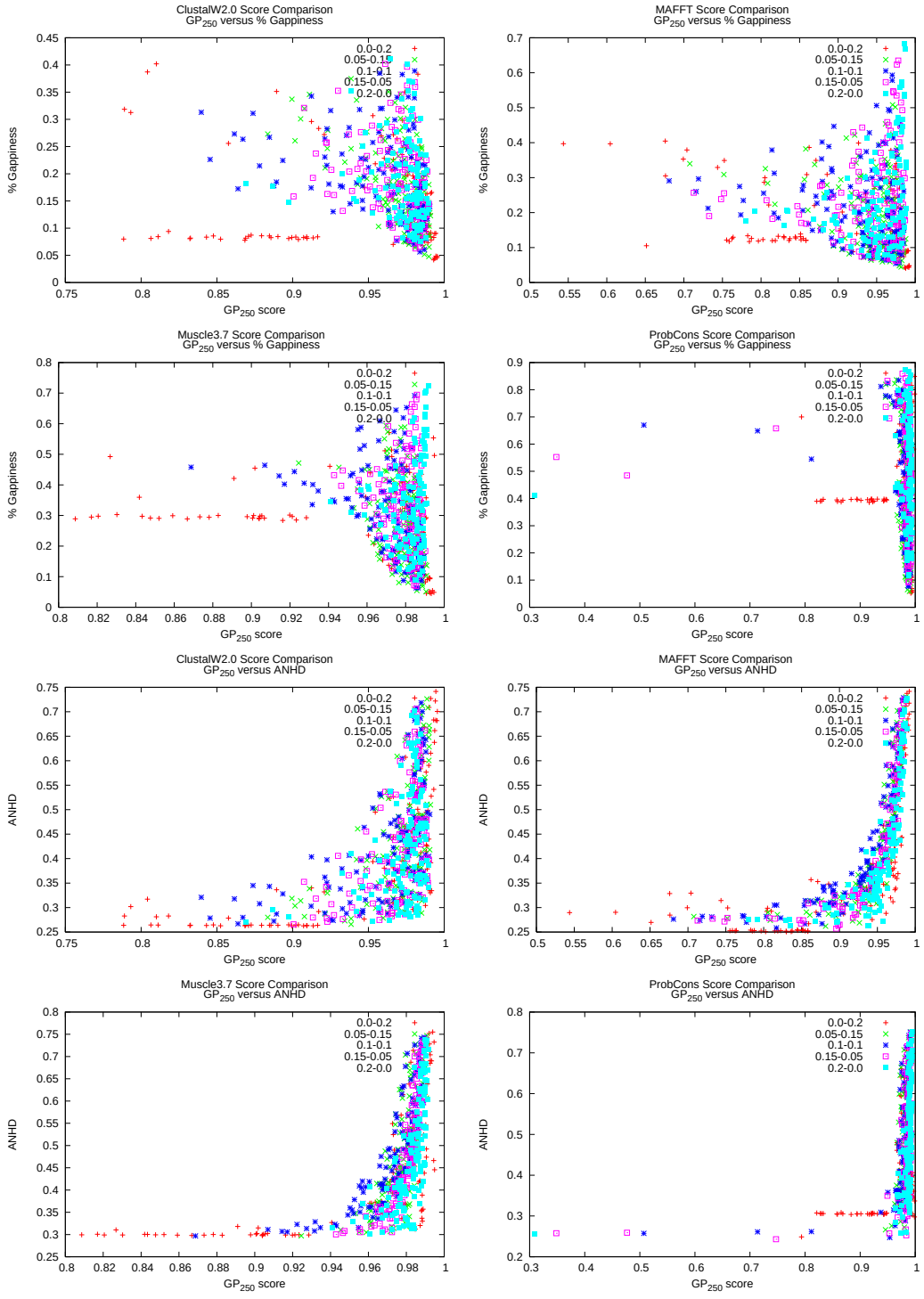

ClustalW2.0 Score Comparison
GP250 versus % Gappiness
MAFFT Score Comparison
GP250 versus % Gappiness
Muscle3.7 Score Comparison
GP250 versus % Gappiness
ProbCons Score Comparison
GP250 versus % Gappiness
ClustalW2.0 Score Comparison
GP250 versus ANHD
MAFFT Score Comparison
GP250 versus ANHD
Muscle3.7 Score Comparison
GP250 versus ANHD
ProbCons Score Comparison
GP250 versus ANHD
% Gappiness
ANHD
GP250 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

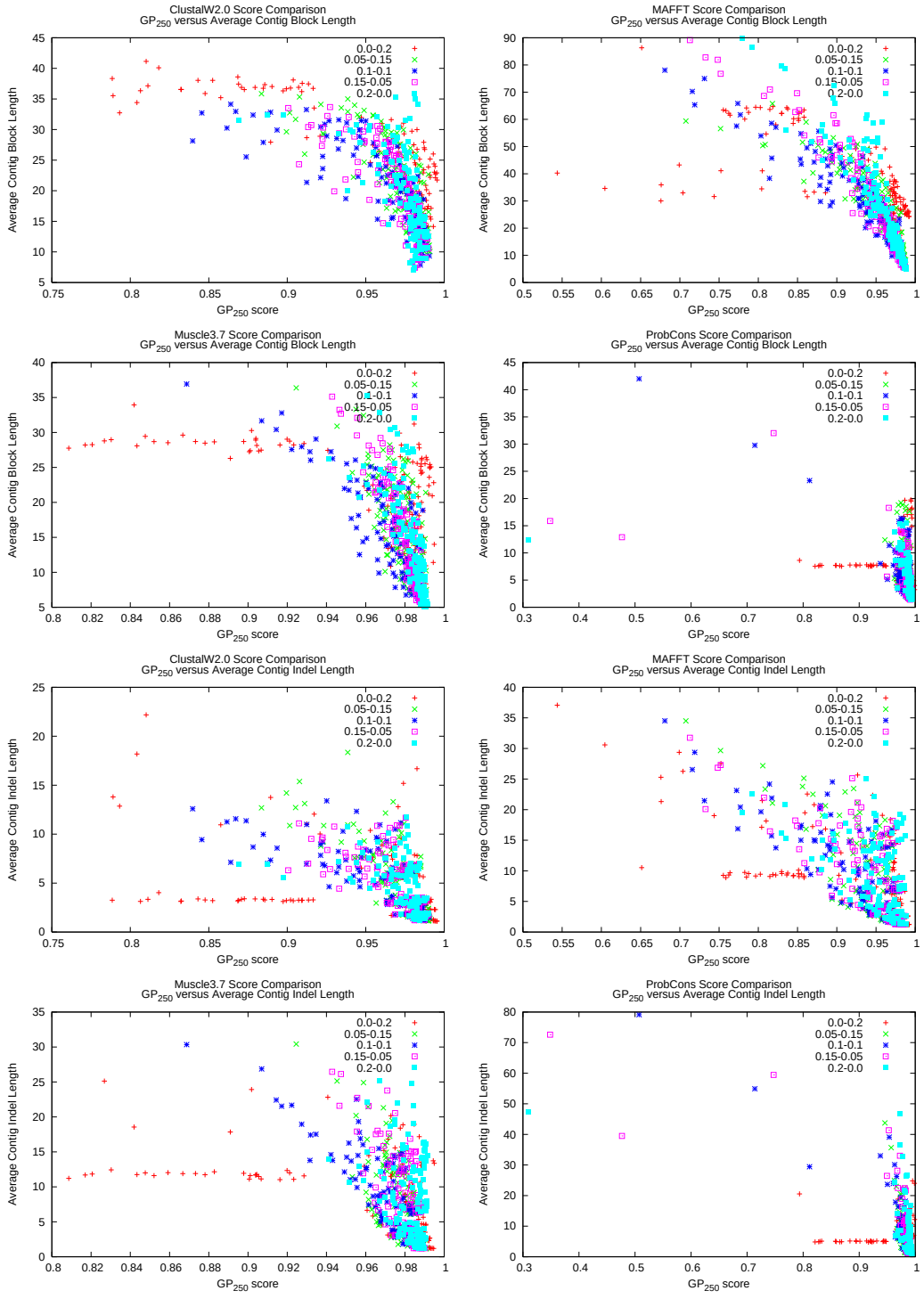

ClustalW2.0 Score Comparison
GP250 versus Average Contig Block Length
MAFFT Score Comparison
GP250 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Block Length
ProbCons Score Comparison
GP250 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP250 versus Average Contig Indel Length
MAFFT Score Comparison
GP250 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Indel Length
ProbCons Score Comparison
GP250 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP250 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

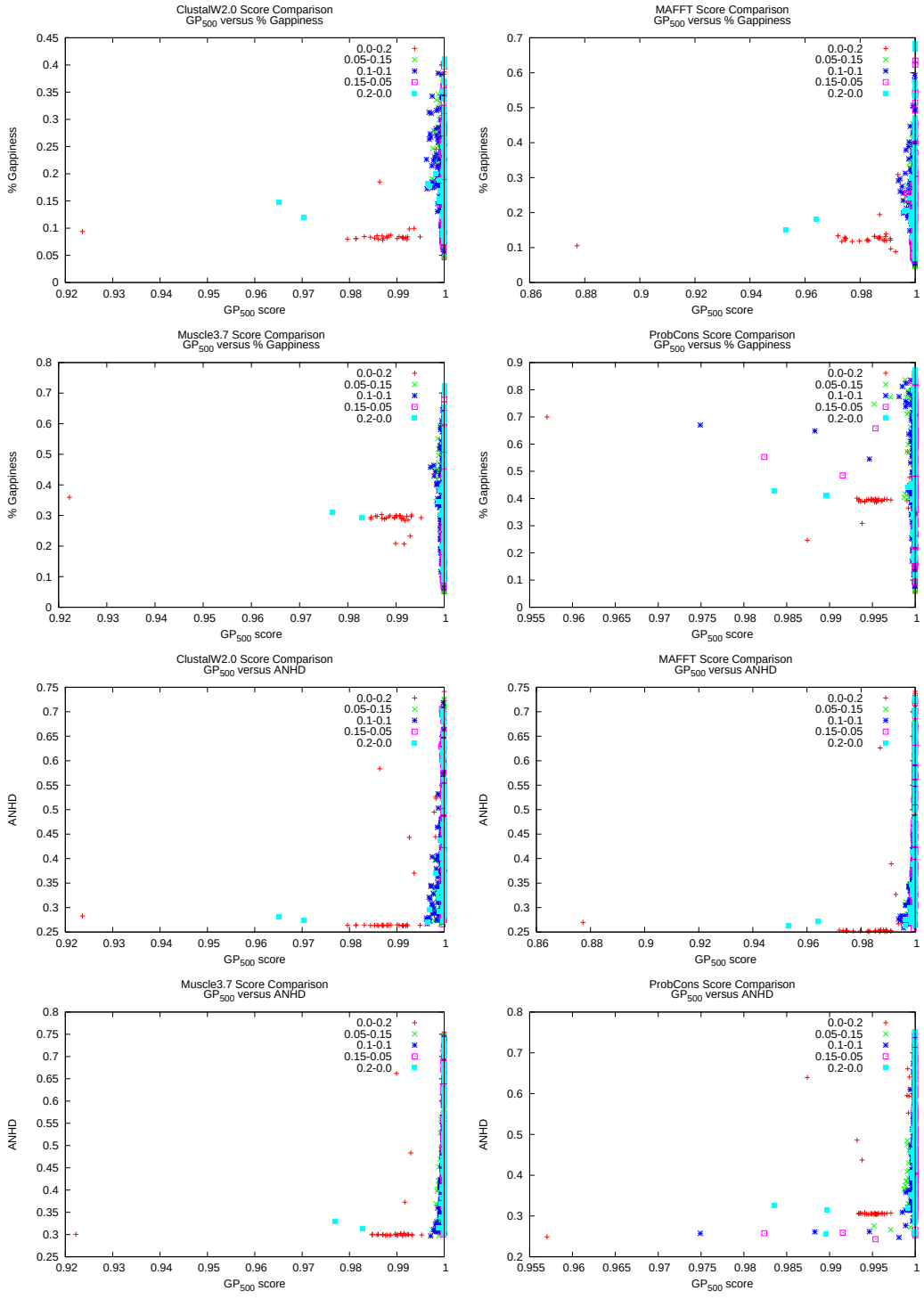

ClustalW2.0 Score Comparison
GP500 versus % Gappiness
MAFFT Score Comparison
GP500 versus % Gappiness
Muscle3.7 Score Comparison
GP500 versus % Gappiness
ProbCons Score Comparison
GP500 versus % Gappiness
ClustalW2.0 Score Comparison
GP500 versus ANHD
MAFFT Score Comparison
GP500 versus ANHD
Muscle3.7 Score Comparison
GP500 versus ANHD
ProbCons Score Comparison
GP500 versus ANHD
% Gappiness
ANHD
GP500 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

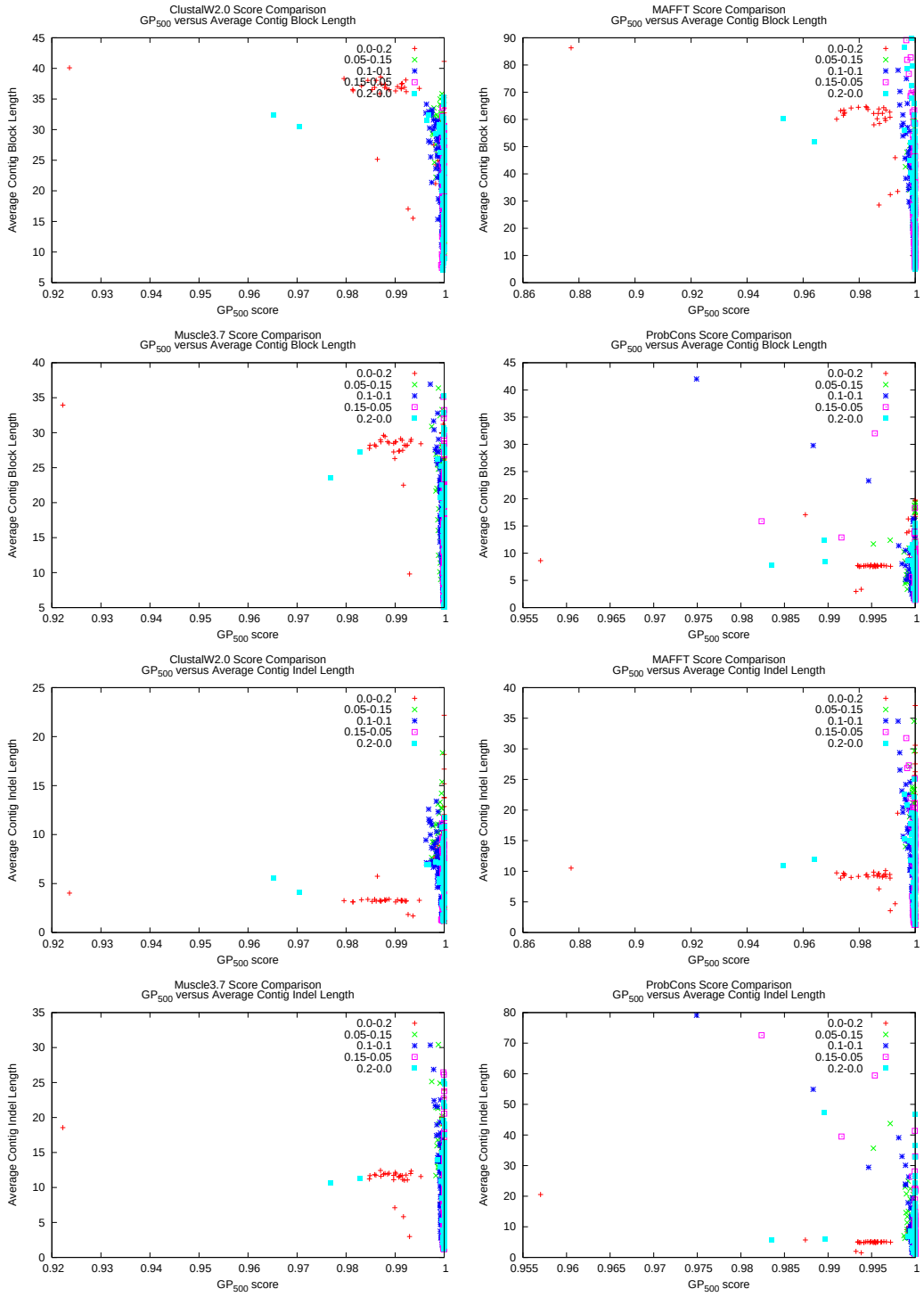

ClustalW2.0 Score Comparison
GP500 versus Average Contig Block Length
MAFFT Score Comparison
GP500 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Block Length
ProbCons Score Comparison
GP500 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP500 versus Average Contig Indel Length
MAFFT Score Comparison
GP500 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Indel Length
ProbCons Score Comparison
GP500 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP500 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

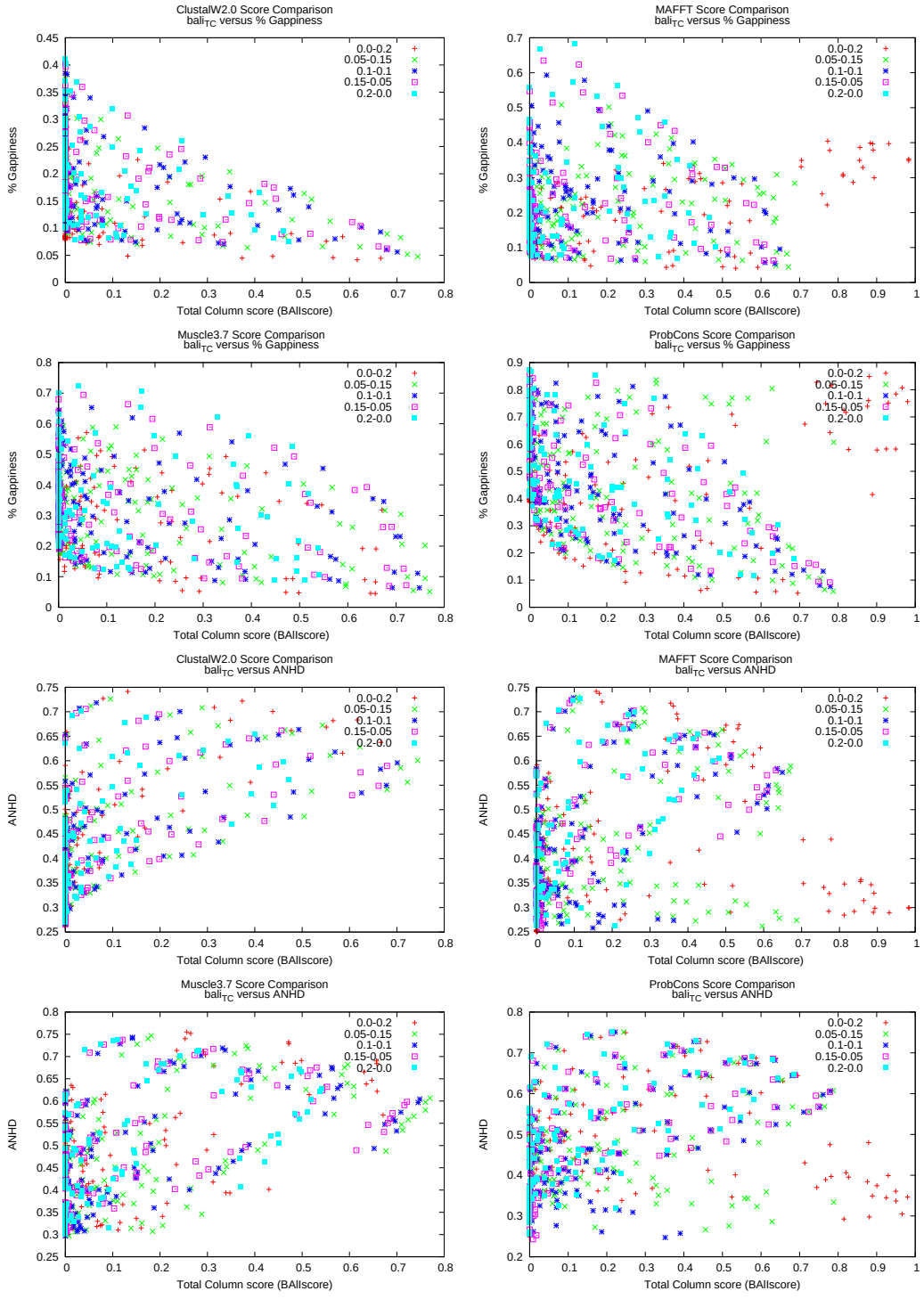

ClustalW2.0 Score Comparison
bali_TC versus % Gappiness
MAFFT Score Comparison
bali_TC versus % Gappiness
Muscle3.7 Score Comparison
bali_TC versus % Gappiness
ProbCons Score Comparison
bali_TC versus % Gappiness
ClustalW2.0 Score Comparison
bali_TC versus ANHD
MAFFT Score Comparison
bali_TC versus ANHD
Muscle3.7 Score Comparison
bali_TC versus ANHD
ProbCons Score Comparison
bali_TC versus ANHD
% Gappiness
ANHD
Total Column score (BAIIscore)
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

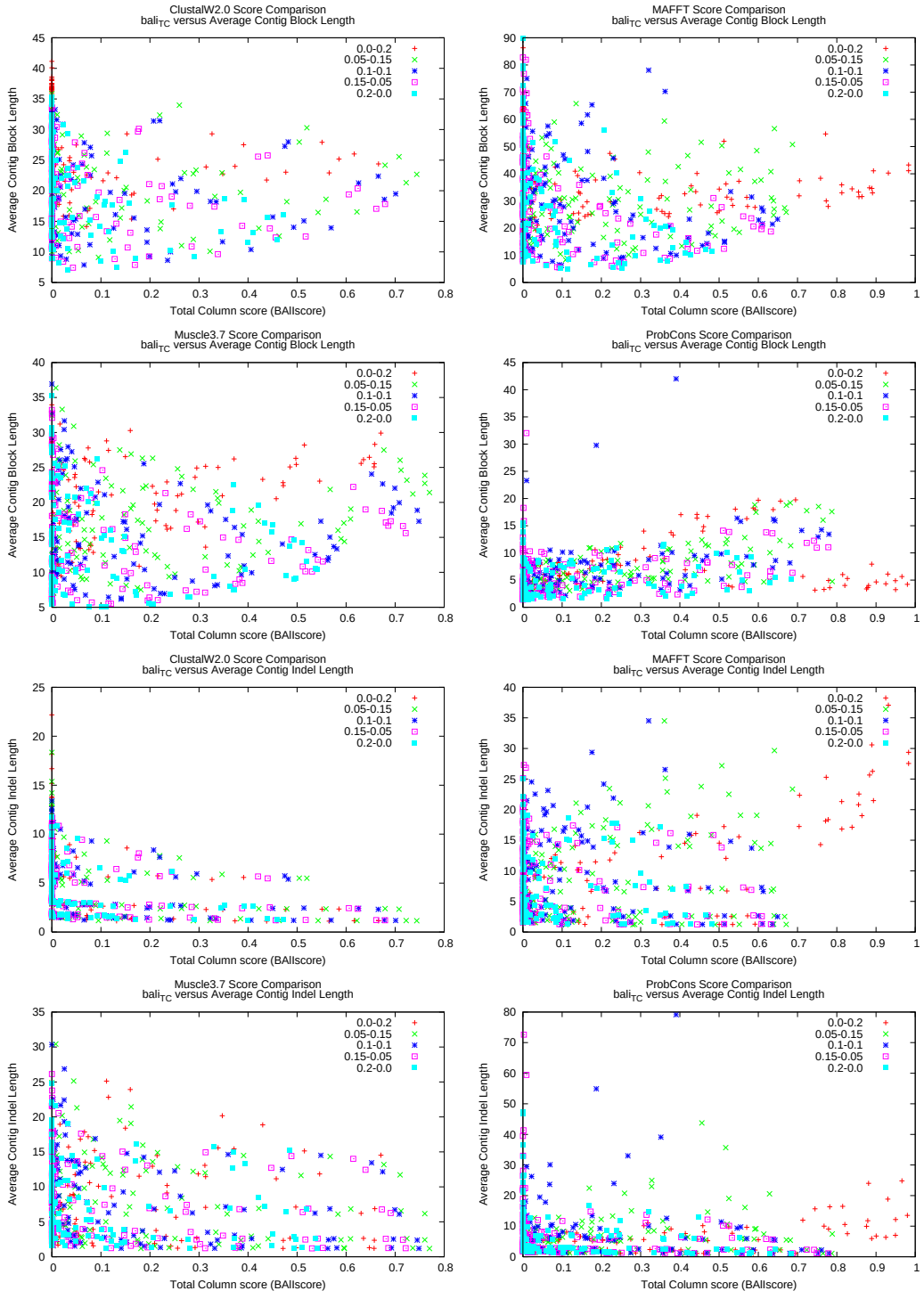

ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Block Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Total Column score (BAIIscore)
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

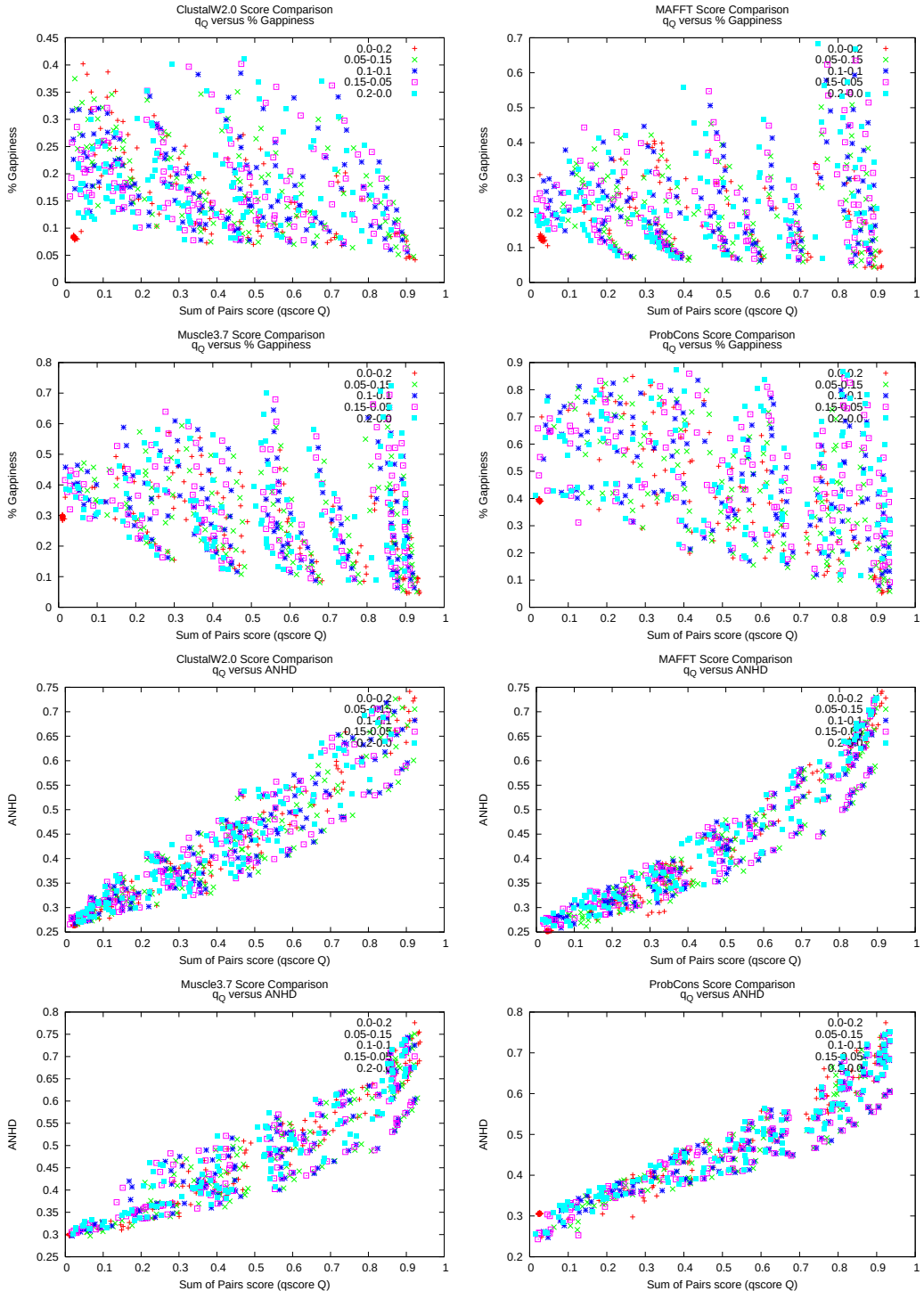

ClustalW2.0 Score Comparison
$q_Q$ versus % Gappiness
MAFFT Score Comparison
$q_Q$ versus % Gappiness
Muscle3.7 Score Comparison
$q_Q$ versus % Gappiness
ProbCons Score Comparison
$q_Q$ versus % Gappiness
ClustalW2.0 Score Comparison
$q_Q$ versus ANHD
MAFFT Score Comparison
$q_Q$ versus ANHD
Muscle3.7 Score Comparison
$q_Q$ versus ANHD
ProbCons Score Comparison
$q_Q$ versus ANHD
% Gappiness
ANHD
Sum of Pairs score (qscore Q)
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

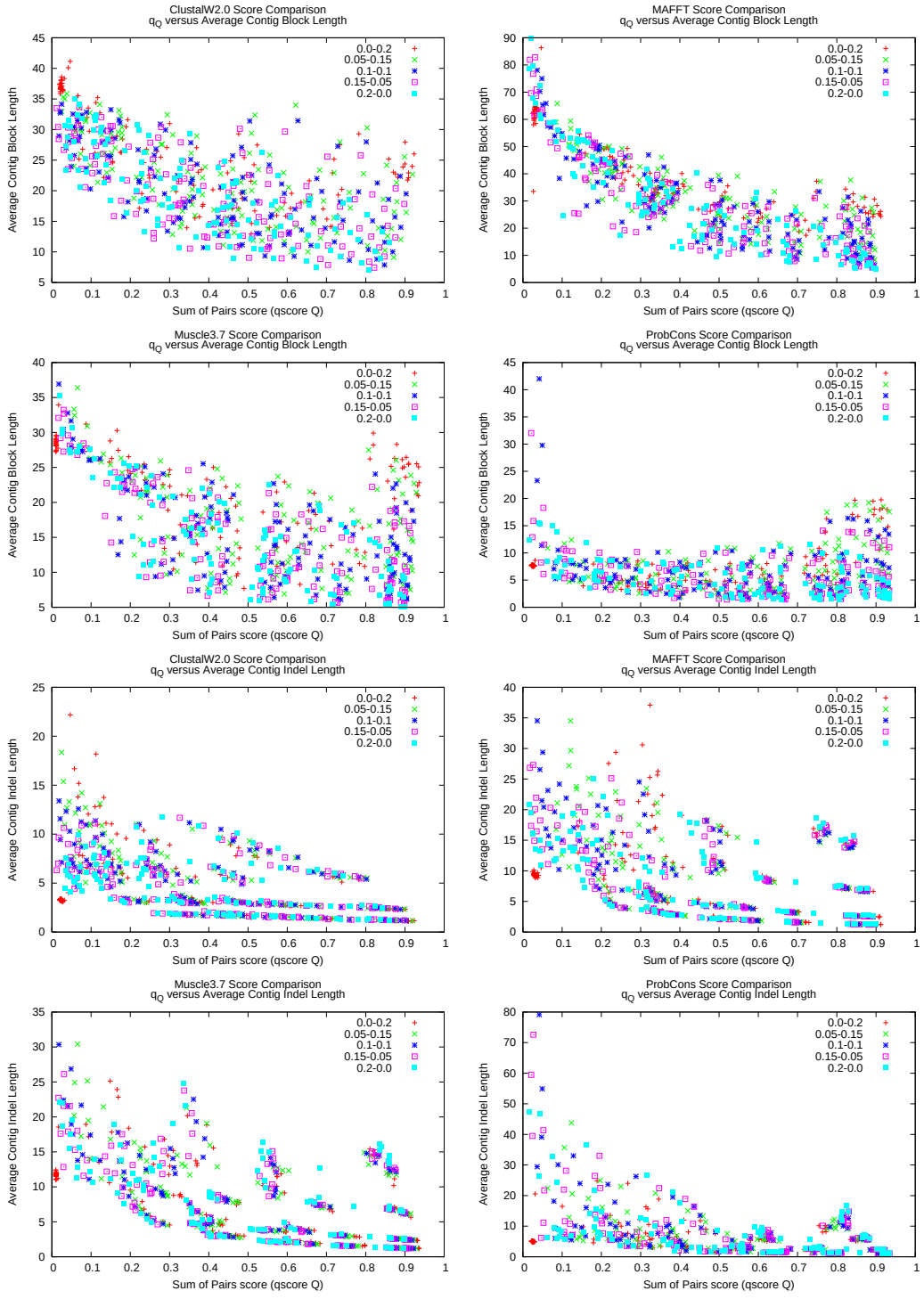

ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Block Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Block Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Indel Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Indel Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Sum of Pairs score (qscore Q)
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

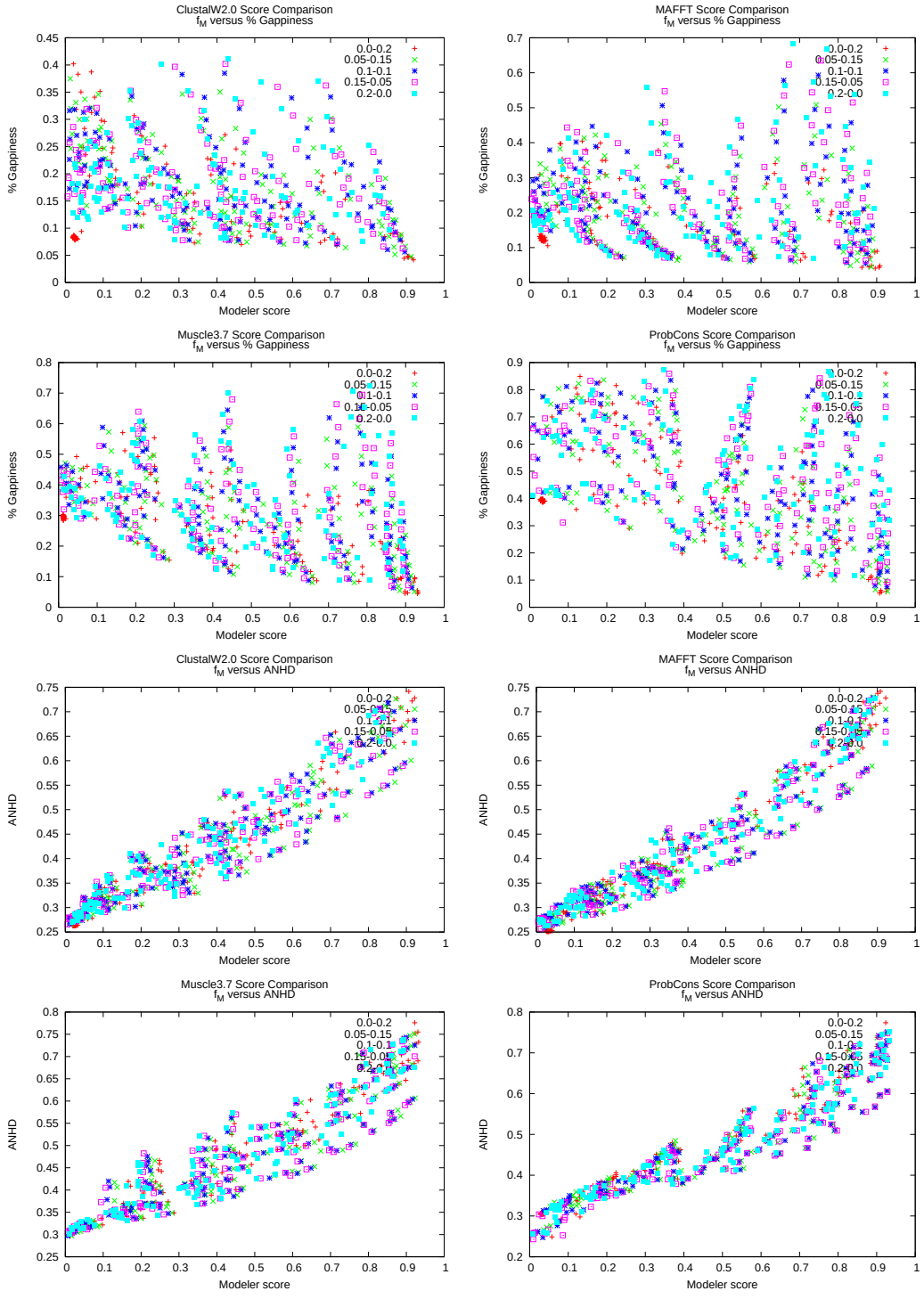

ClustalW2.0 Score Comparison
$f_M$ versus % Gappiness
MAFFT Score Comparison
$f_M$ versus % Gappiness
Muscle3.7 Score Comparison
$f_M$ versus % Gappiness
ProbCons Score Comparison
$f_M$ versus % Gappiness
ClustalW2.0 Score Comparison
$f_M$ versus ANHD
MAFFT Score Comparison
$f_M$ versus ANHD
Muscle3.7 Score Comparison
$f_M$ versus ANHD
ProbCons Score Comparison
$f_M$ versus ANHD
% Gappiness
ANHD
Modeler score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

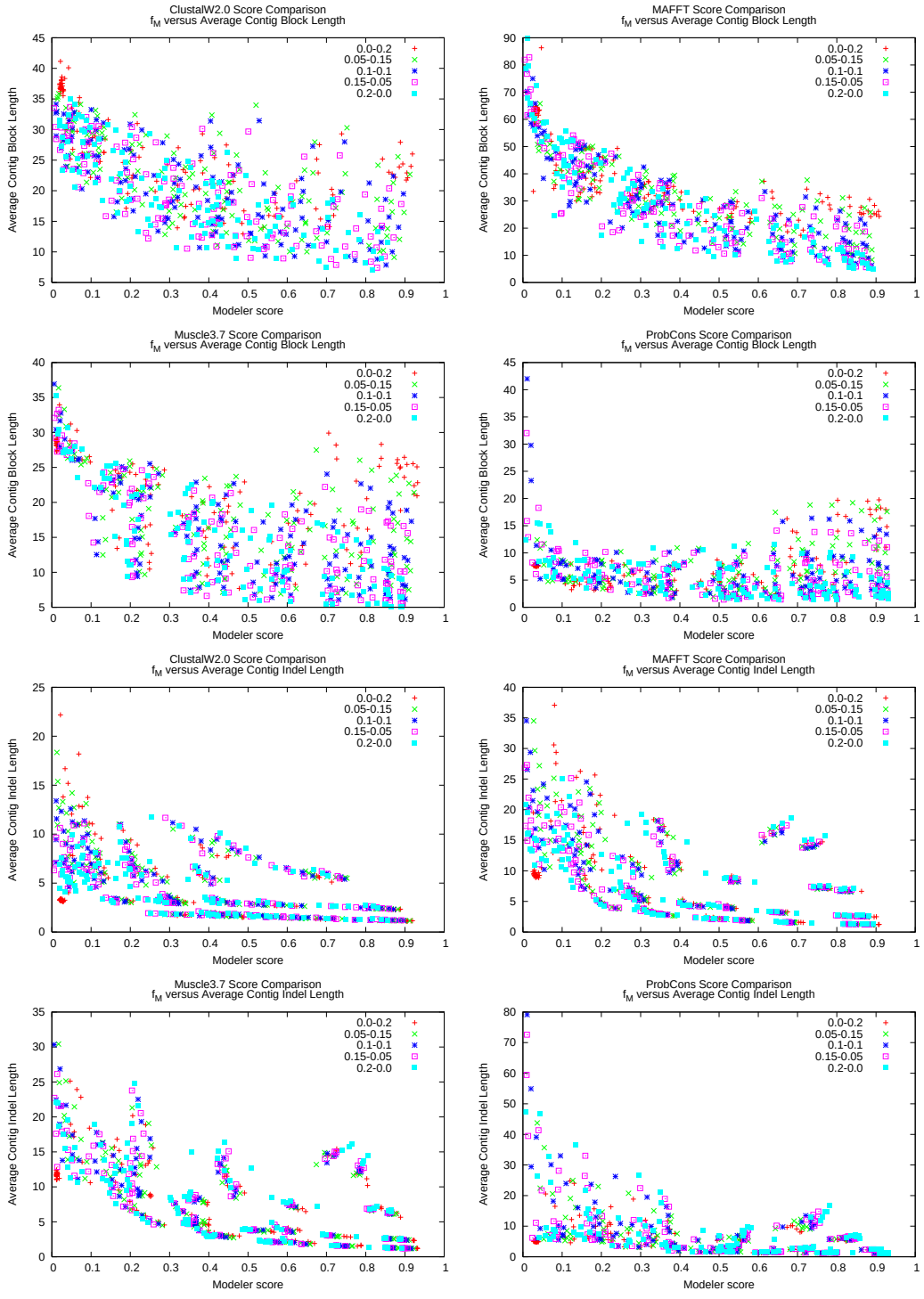

ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Block Length
MAFFT Score Comparison
$f_M$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Block Length
ProbCons Score Comparison
$f_M$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Indel Length
MAFFT Score Comparison
$f_M$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Indel Length
ProbCons Score Comparison
$f_M$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Modeler score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

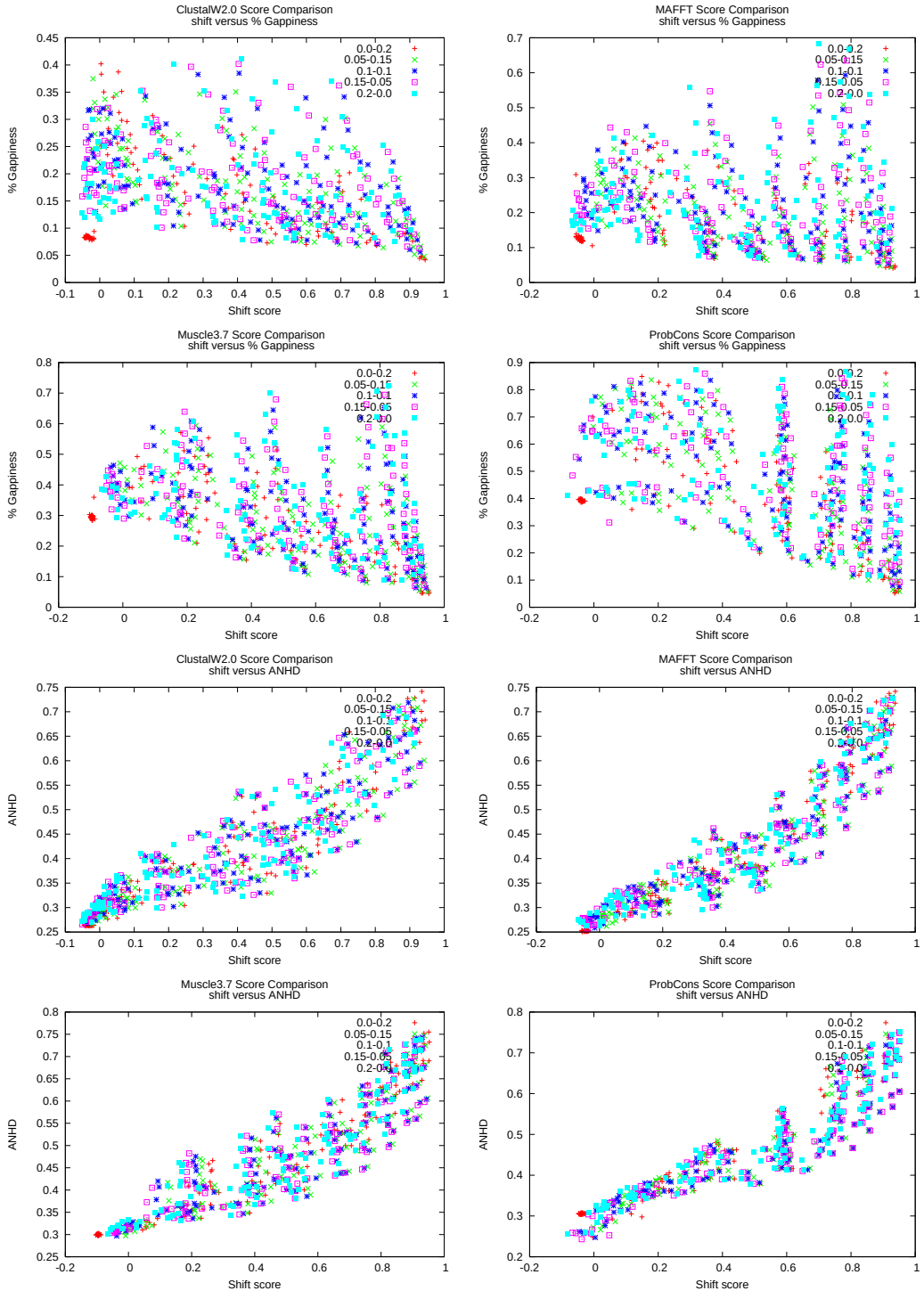

ClustalW2.0 Score Comparison
shift versus % Gappiness
MAFFT Score Comparison
shift versus % Gappiness
Muscle3.7 Score Comparison
shift versus % Gappiness
ProbCons Score Comparison
shift versus % Gappiness
ClustalW2.0 Score Comparison
shift versus ANHD
MAFFT Score Comparison
shift versus ANHD
Muscle3.7 Score Comparison
shift versus ANHD
ProbCons Score Comparison
shift versus ANHD
% Gappiness
ANHD
Shift score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

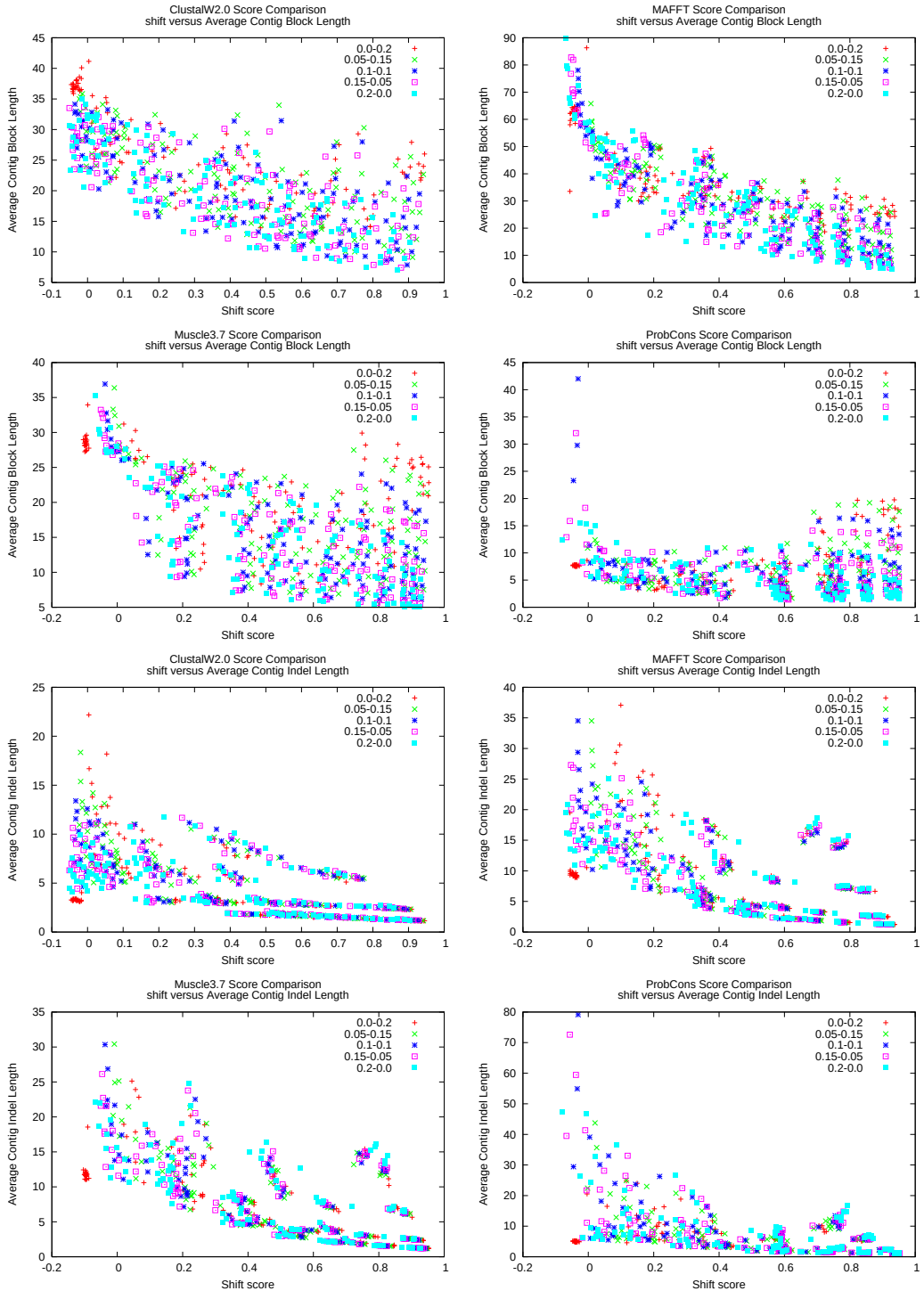

ClustalW2.0 Score Comparison
shift versus Average Contig Block Length
MAFFT Score Comparison
shift versus Average Contig Block Length
Muscle3.7 Score Comparison
shift versus Average Contig Block Length
ProbCons Score Comparison
shift versus Average Contig Block Length
ClustalW2.0 Score Comparison
shift versus Average Contig Indel Length
MAFFT Score Comparison
shift versus Average Contig Indel Length
Muscle3.7 Score Comparison
shift versus Average Contig Indel Length
ProbCons Score Comparison
shift versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Shift score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.1

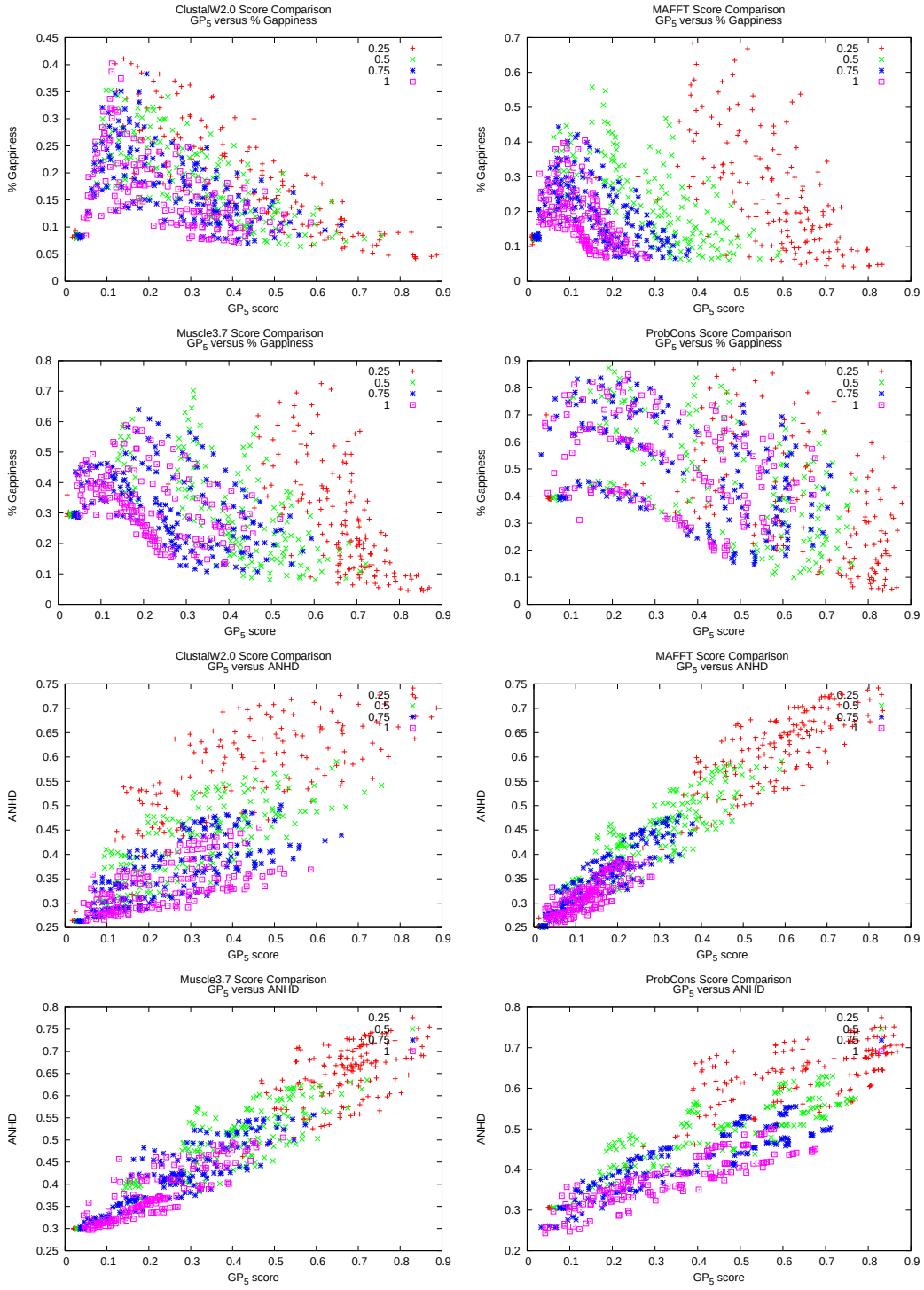

ClustalW2.0 Score Comparison
GP5 versus % Gappiness
MAFFT Score Comparison
GP5 versus % Gappiness
Muscle3.7 Score Comparison
GP5 versus % Gappiness
ProbCons Score Comparison
GP5 versus % Gappiness
ClustalW2.0 Score Comparison
GP5 versus ANHD
MAFFT Score Comparison
GP5 versus ANHD
Muscle3.7 Score Comparison
GP5 versus ANHD
ProbCons Score Comparison
GP5 versus ANHD
% Gappiness
ANHD
GP5 score
0.25
0.5
0.75
1

Figure B.1

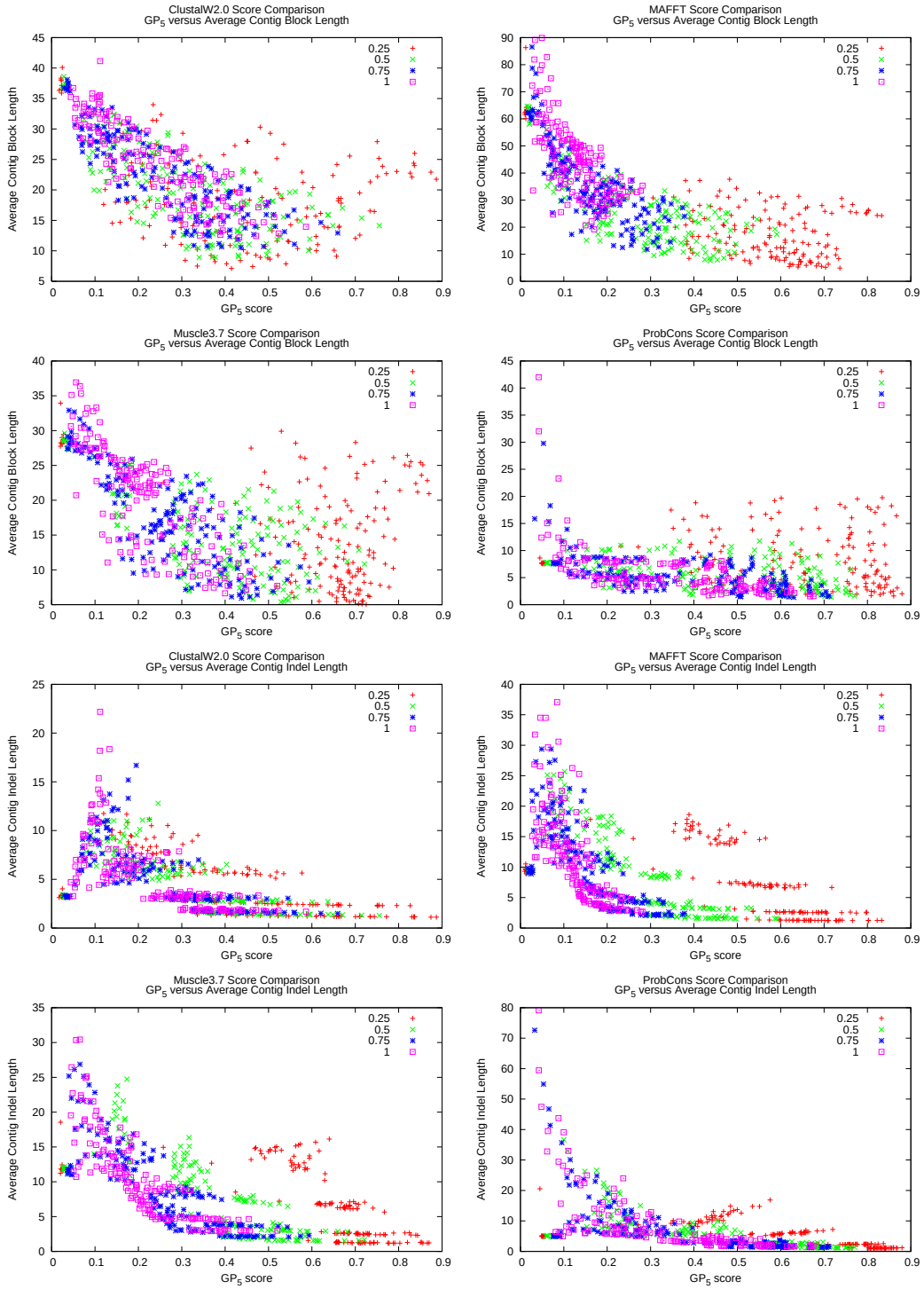

ClustalW2.0 Score Comparison
GP5 versus Average Contig Block Length
MAFFT Score Comparison
GP5 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Block Length
ProbCons Score Comparison
GP5 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP5 versus Average Contig Indel Length
MAFFT Score Comparison
GP5 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Indel Length
ProbCons Score Comparison
GP5 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP5 score
0.25
0.5
0.75
1

Figure B.1

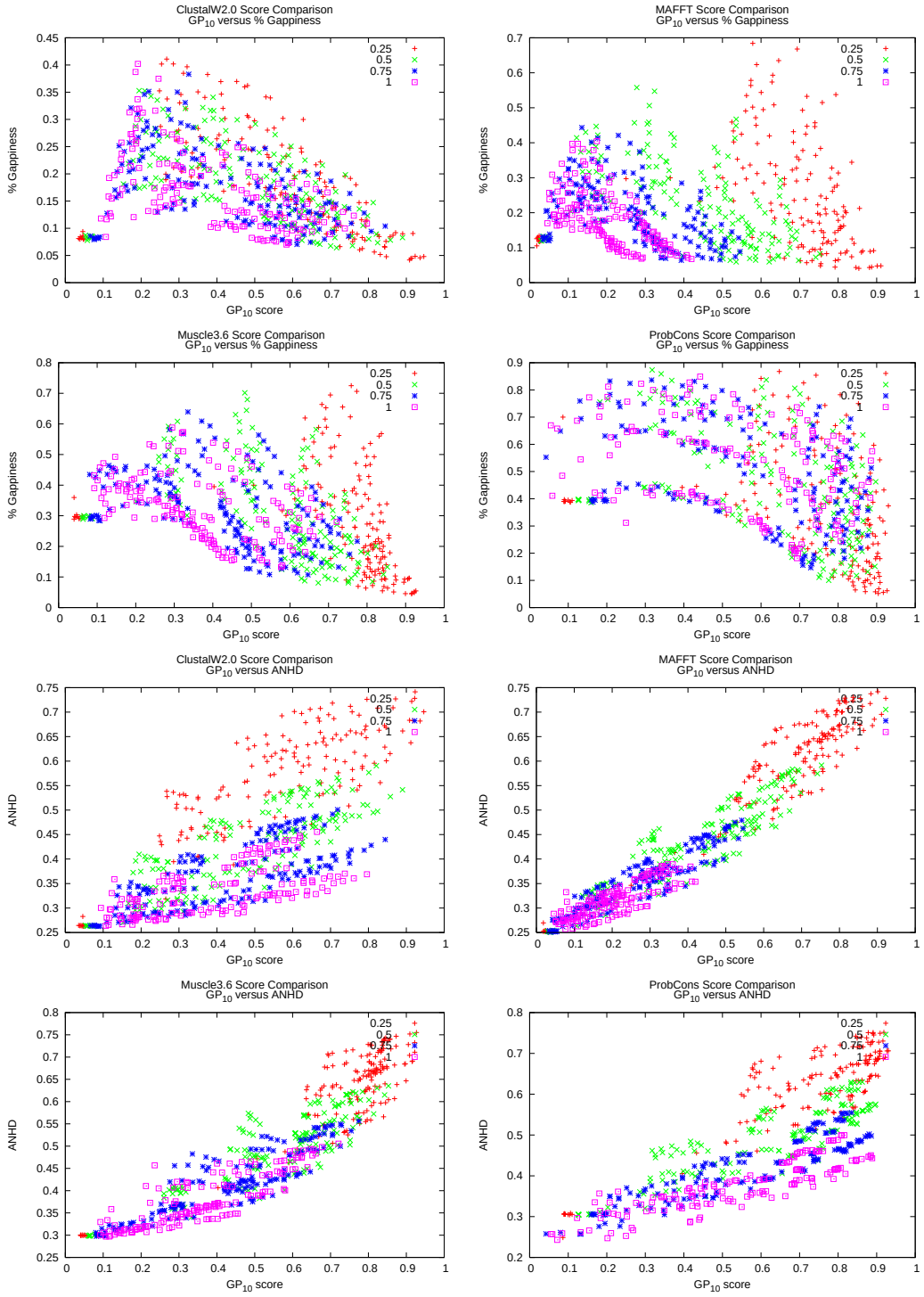

ClustalW2.0 Score Comparison
GP10 versus % Gappiness
MAFFT Score Comparison
GP10 versus % Gappiness
Muscle3.6 Score Comparison
GP10 versus % Gappiness
ProbCons Score Comparison
GP10 versus % Gappiness
ClustalW2.0 Score Comparison
GP10 versus ANHD
MAFFT Score Comparison
GP10 versus ANHD
Muscle3.6 Score Comparison
GP10 versus ANHD
ProbCons Score Comparison
GP10 versus ANHD
% Gappiness
ANHD
GP10 score
0.25
0.5
0.75
1

Figure B.1

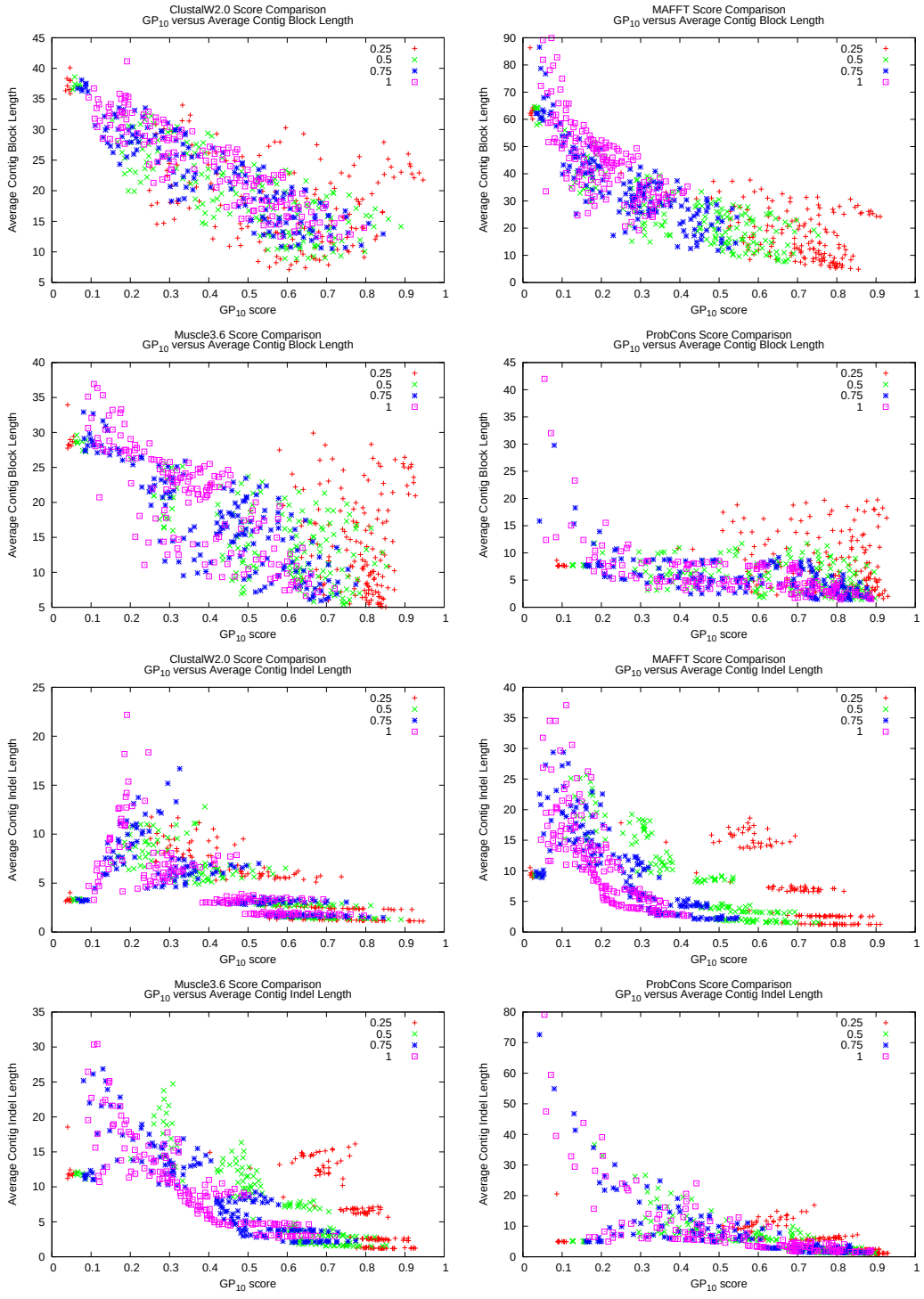

ClustalW2.0 Score Comparison
GP10 versus Average Contig Block Length
MAFFT Score Comparison
GP10 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Block Length
ProbCons Score Comparison
GP10 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP10 versus Average Contig Indel Length
MAFFT Score Comparison
GP10 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Indel Length
ProbCons Score Comparison
GP10 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP10 score
0.25
0.5
0.75
1

Figure B.1

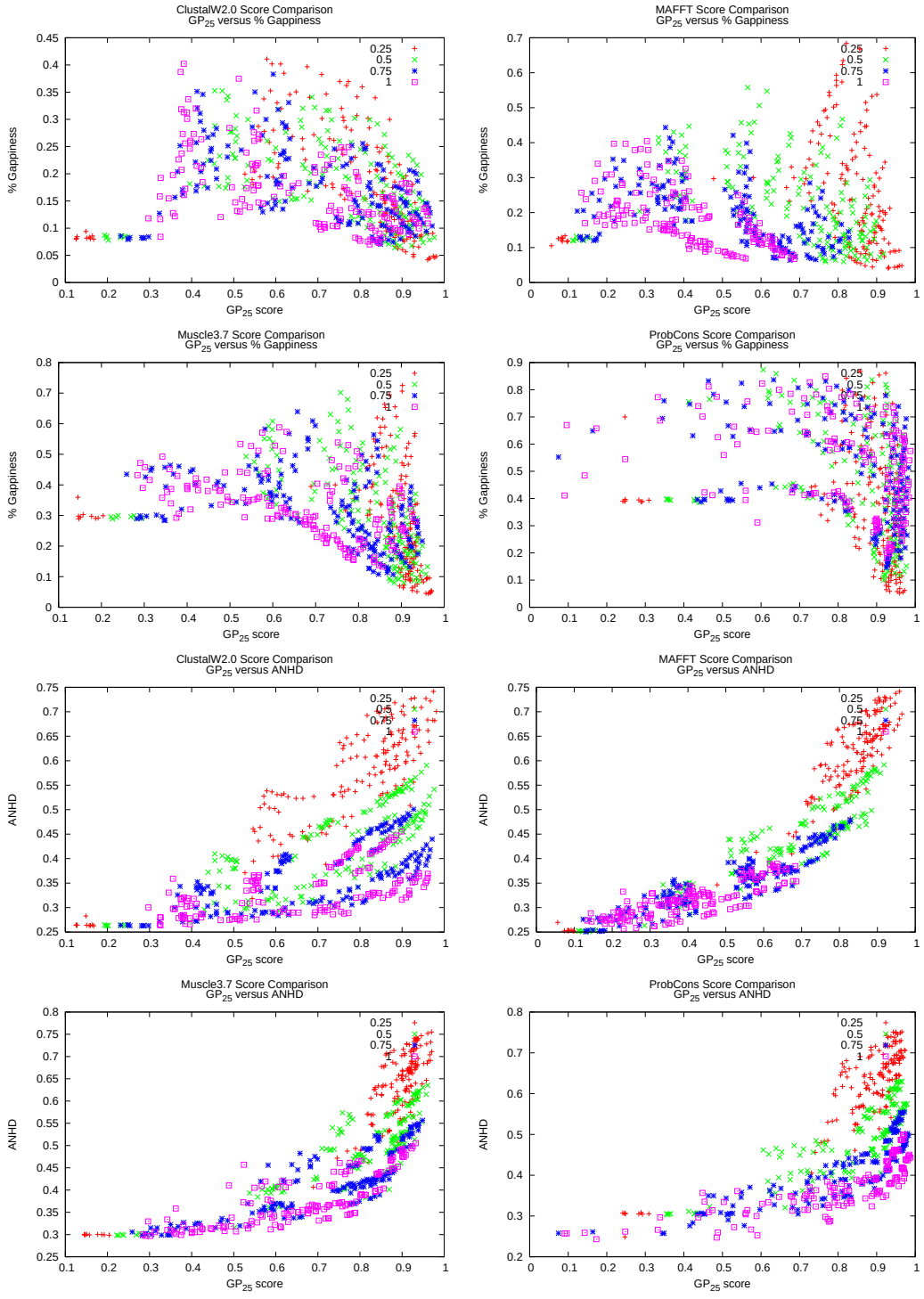

ClustalW2.0 Score Comparison
GP25 versus % Gappiness
MAFFT Score Comparison
GP25 versus % Gappiness
Muscle3.7 Score Comparison
GP25 versus % Gappiness
ProbCons Score Comparison
GP25 versus % Gappiness
ClustalW2.0 Score Comparison
GP25 versus ANHD
MAFFT Score Comparison
GP25 versus ANHD
Muscle3.7 Score Comparison
GP25 versus ANHD
ProbCons Score Comparison
GP25 versus ANHD
% Gappiness
ANHD
GP25 score
0.25
0.5
0.75
1

Figure B.1

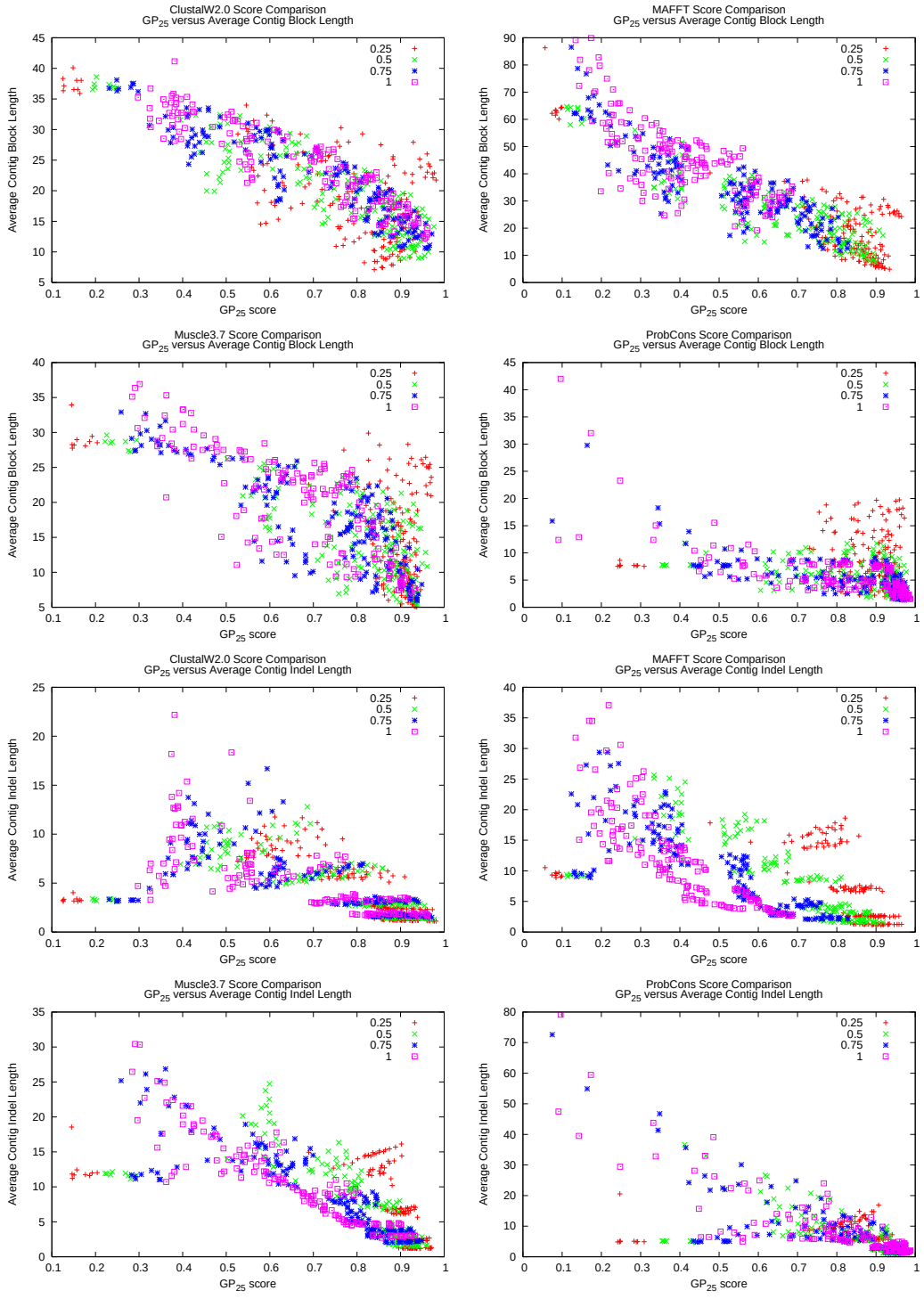

ClustalW2.0 Score Comparison
GP25 versus Average Contig Block Length
MAFFT Score Comparison
GP25 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Block Length
ProbCons Score Comparison
GP25 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP25 versus Average Contig Indel Length
MAFFT Score Comparison
GP25 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Indel Length
ProbCons Score Comparison
GP25 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP25 score
0.25
0.5
0.75
1

Figure B.1

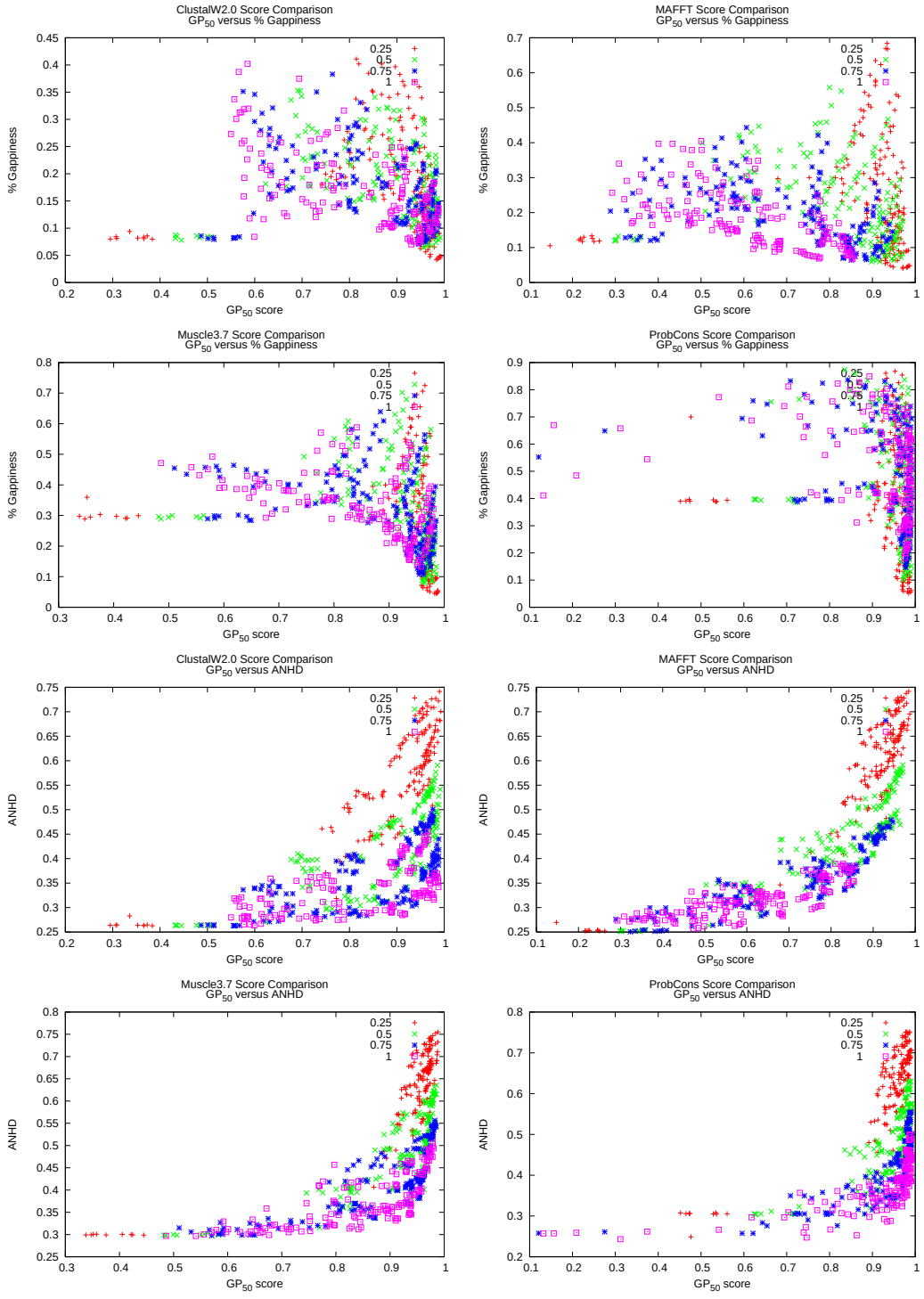

ClustalW2.0 Score Comparison
GP50 versus % Gappiness
MAFFT Score Comparison
GP50 versus % Gappiness
Muscle3.7 Score Comparison
GP50 versus % Gappiness
ProbCons Score Comparison
GP50 versus % Gappiness
ClustalW2.0 Score Comparison
GP50 versus ANHD
MAFFT Score Comparison
GP50 versus ANHD
Muscle3.7 Score Comparison
GP50 versus ANHD
ProbCons Score Comparison
GP50 versus ANHD
% Gappiness
ANHD
GP50 score
0.25
0.5
0.75
1

Figure B.1

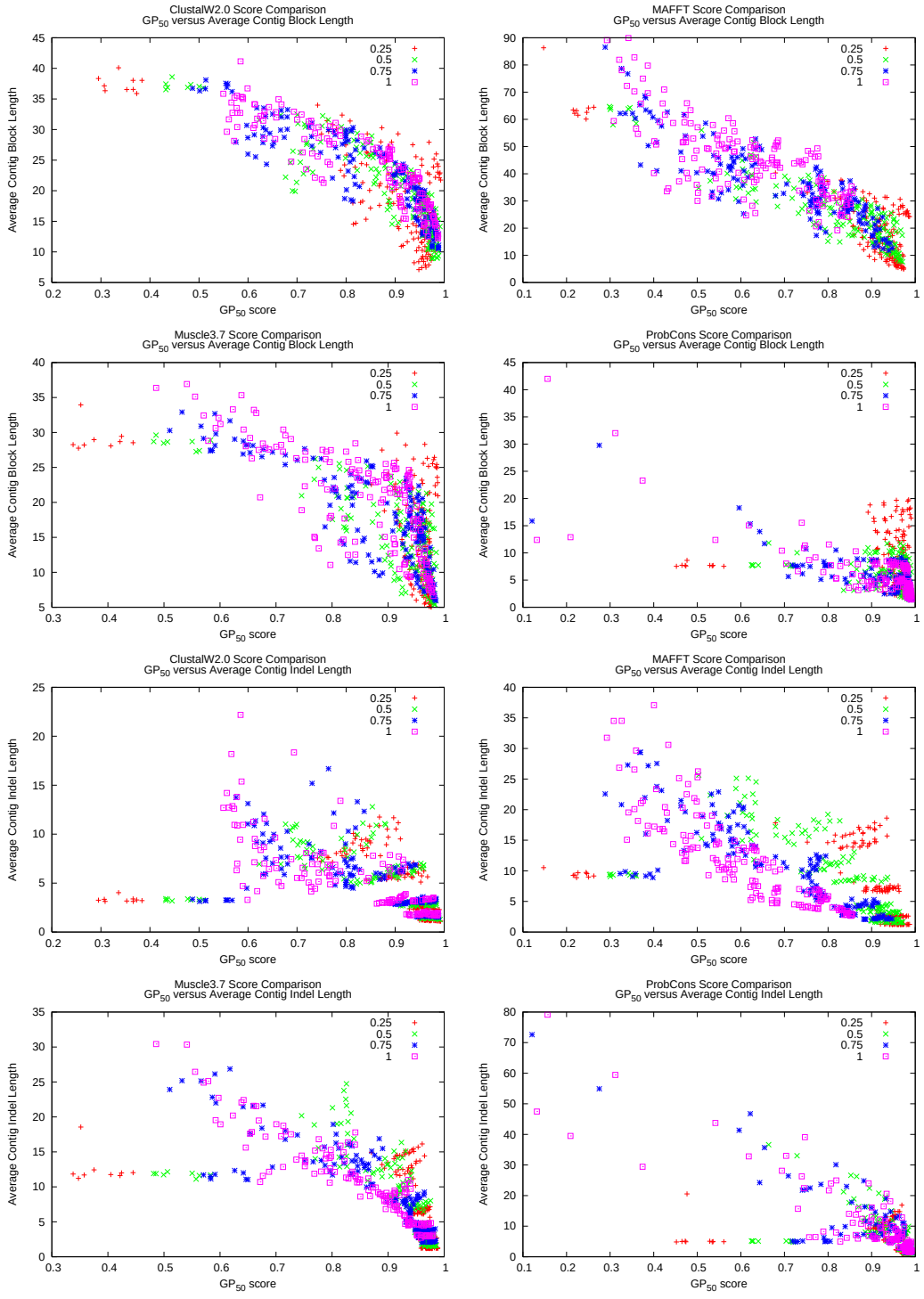

ClustalW2.0 Score Comparison
GP50 versus Average Contig Block Length
MAFFT Score Comparison
GP50 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Block Length
ProbCons Score Comparison
GP50 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP50 versus Average Contig Indel Length
MAFFT Score Comparison
GP50 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Indel Length
ProbCons Score Comparison
GP50 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP50 score
0.25
0.5
0.75
1

Figure B.1

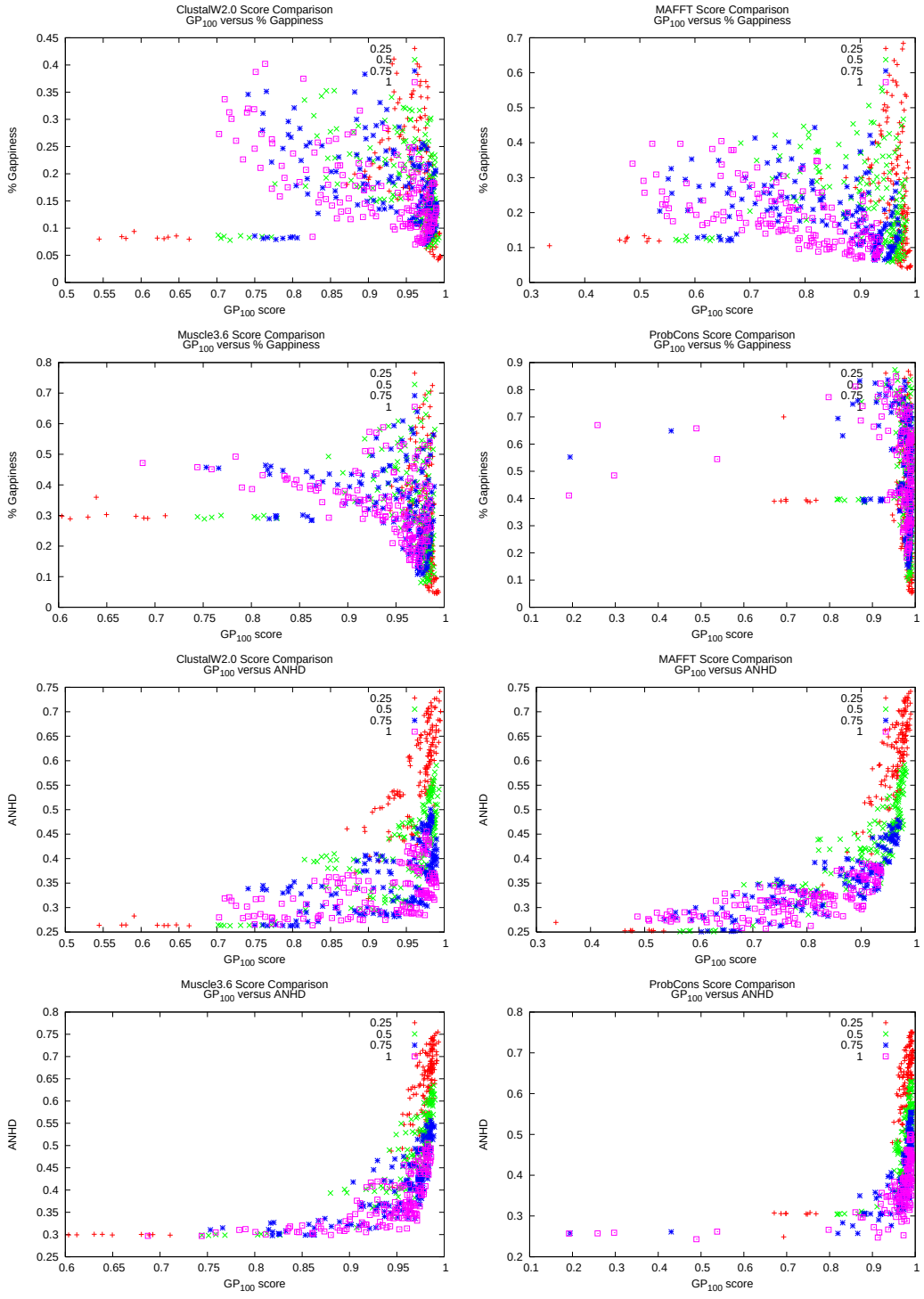

ClustalW2.0 Score Comparison
GP100 versus % Gappiness
MAFFT Score Comparison
GP100 versus % Gappiness
Muscle3.6 Score Comparison
GP100 versus % Gappiness
ProbCons Score Comparison
GP100 versus % Gappiness
ClustalW2.0 Score Comparison
GP100 versus ANHD
MAFFT Score Comparison
GP100 versus ANHD
Muscle3.6 Score Comparison
GP100 versus ANHD
ProbCons Score Comparison
GP100 versus ANHD
% Gappiness
ANHD
GP100 score
0.25
0.5
0.75
1

Figure B.1

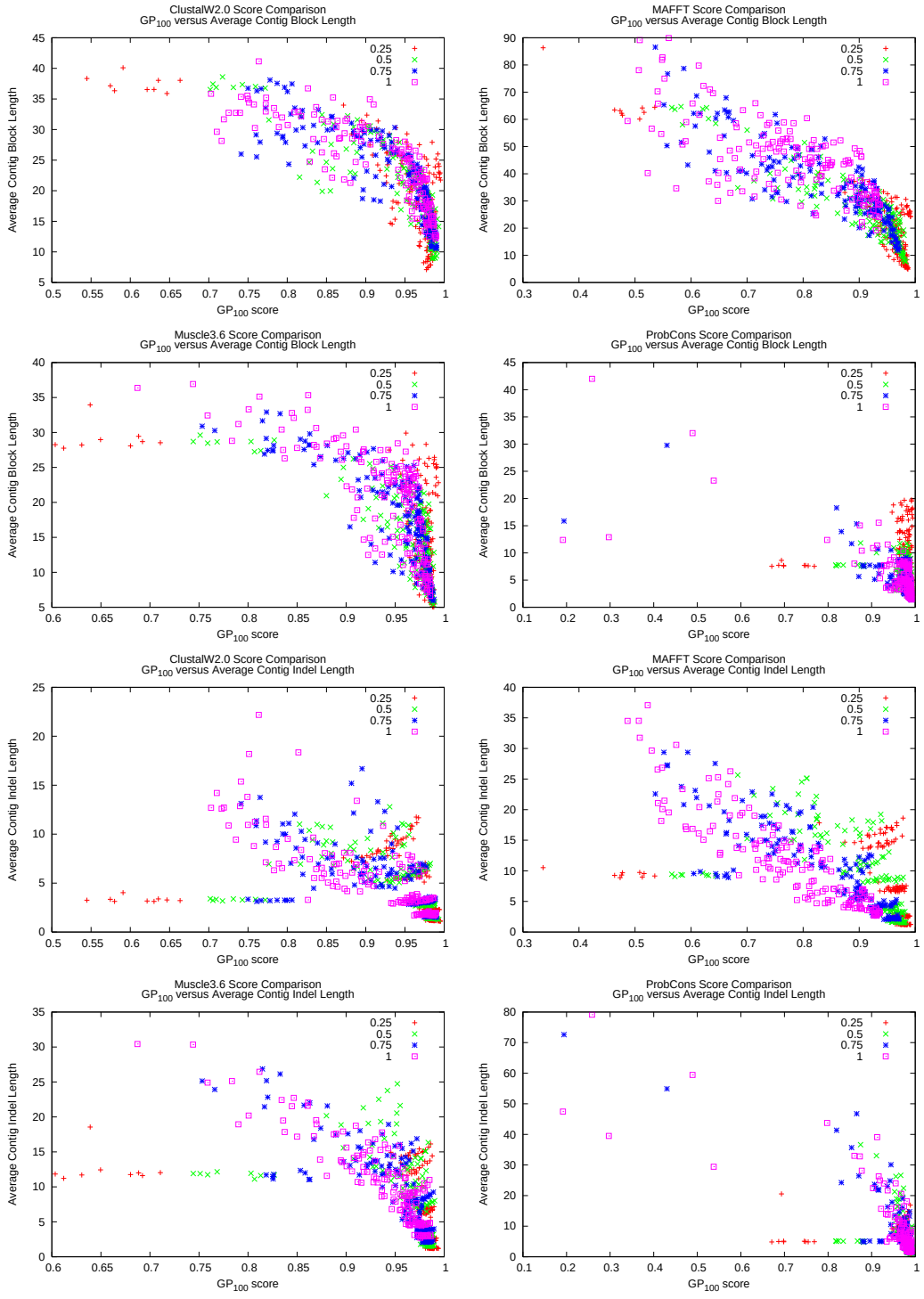

ClustalW2.0 Score Comparison
GP100 versus Average Contig Block Length
MAFFT Score Comparison
GP100 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Block Length
ProbCons Score Comparison
GP100 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP100 versus Average Contig Indel Length
MAFFT Score Comparison
GP100 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Indel Length
ProbCons Score Comparison
GP100 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP100 score
0.25
0.5
0.75
1

Figure B.1

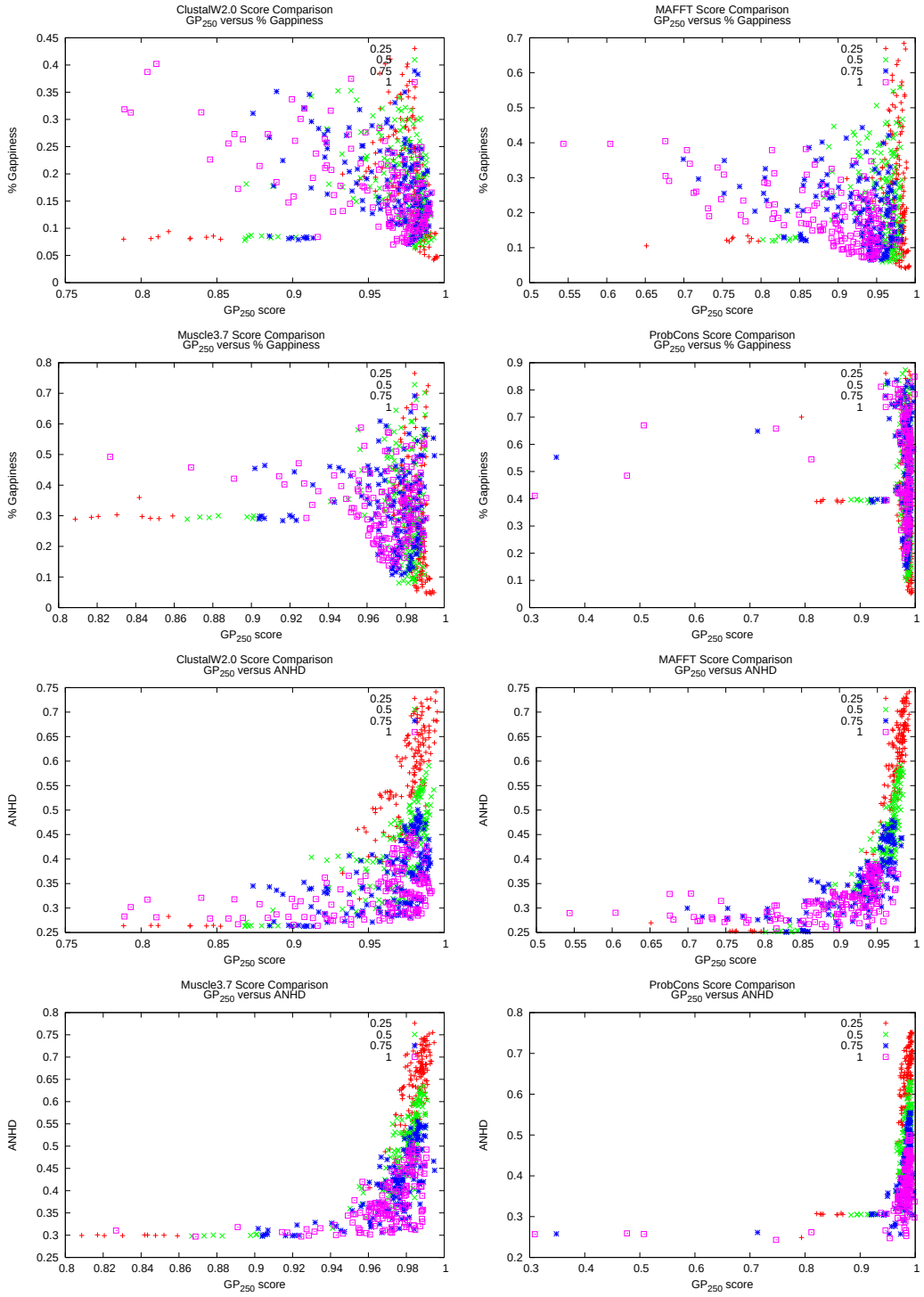

ClustalW2.0 Score Comparison
GP250 versus % Gappiness
MAFFT Score Comparison
GP250 versus % Gappiness
Muscle3.7 Score Comparison
GP250 versus % Gappiness
ProbCons Score Comparison
GP250 versus % Gappiness
ClustalW2.0 Score Comparison
GP250 versus ANHD
MAFFT Score Comparison
GP250 versus ANHD
Muscle3.7 Score Comparison
GP250 versus ANHD
ProbCons Score Comparison
GP250 versus ANHD
% Gappiness
ANHD
GP250 score
0.25
0.5
0.75
1

Figure B.1

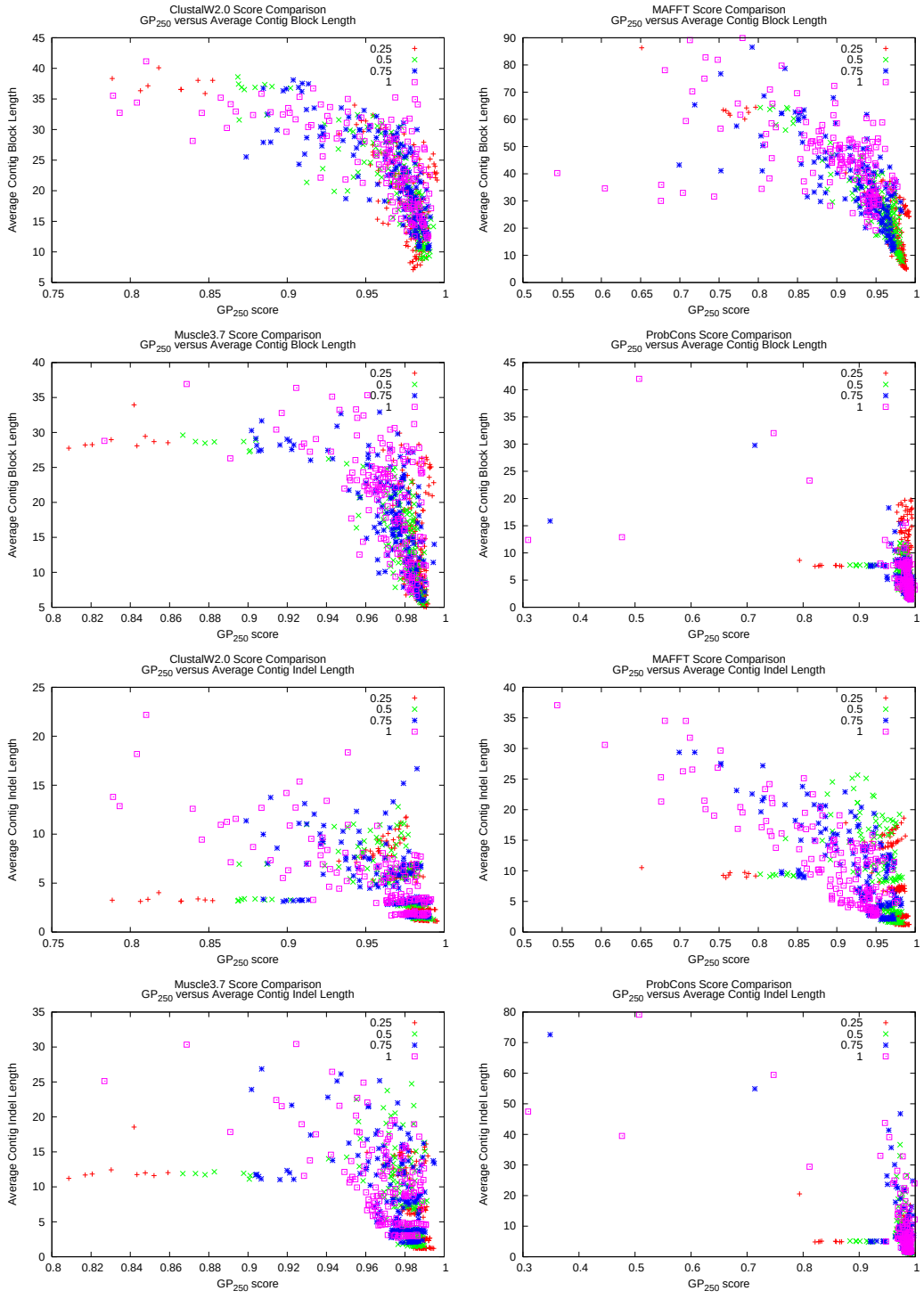

ClustalW2.0 Score Comparison
GP250 versus Average Contig Block Length
MAFFT Score Comparison
GP250 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Block Length
ProbCons Score Comparison
GP250 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP250 versus Average Contig Indel Length
MAFFT Score Comparison
GP250 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Indel Length
ProbCons Score Comparison
GP250 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP250 score
0.25
0.5
0.75
1

Figure B.1

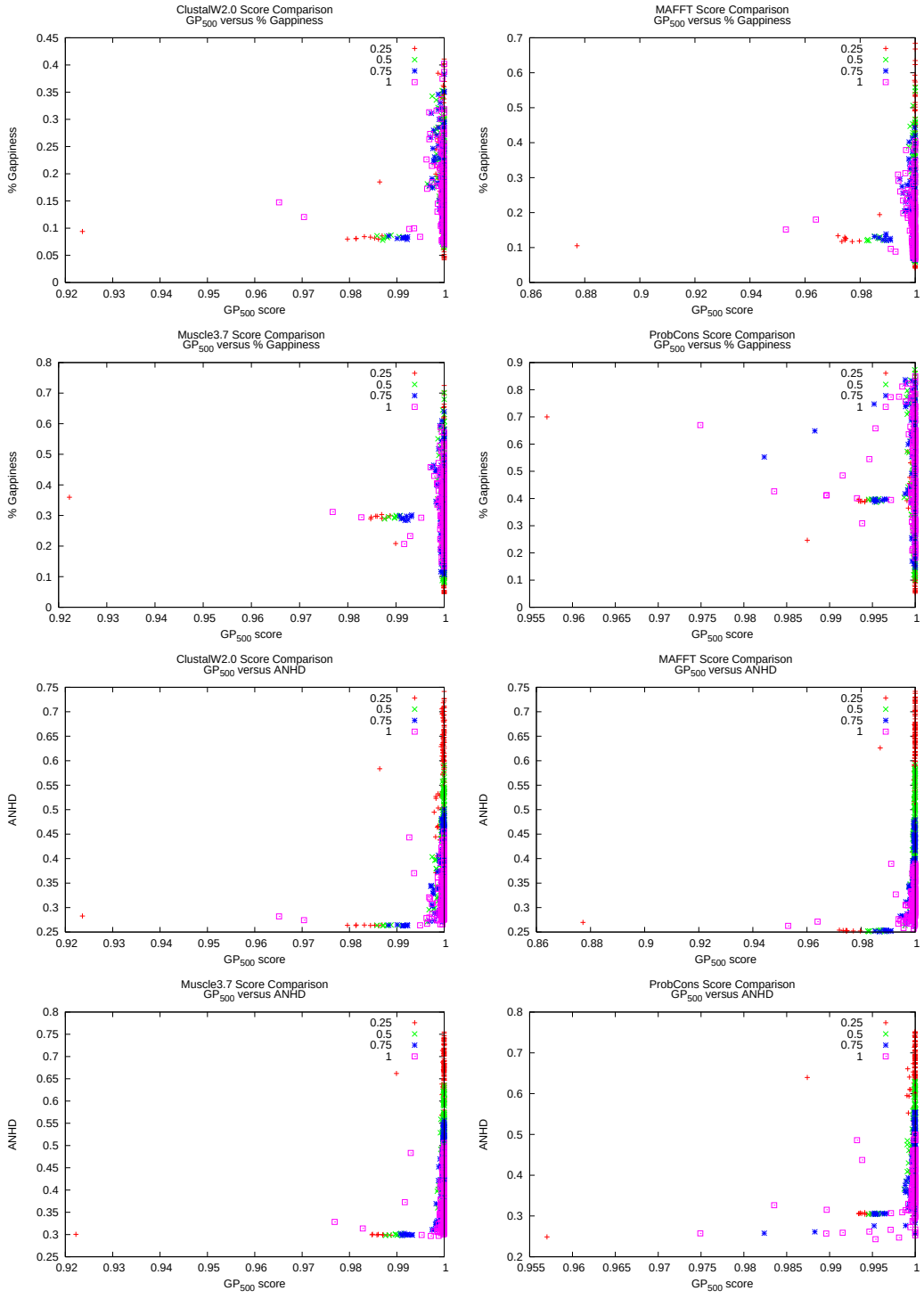

ClustalW2.0 Score Comparison
GP500 versus % Gappiness
MAFFT Score Comparison
GP500 versus % Gappiness
Muscle3.7 Score Comparison
GP500 versus % Gappiness
ProbCons Score Comparison
GP500 versus % Gappiness
ClustalW2.0 Score Comparison
GP500 versus ANHD
MAFFT Score Comparison
GP500 versus ANHD
Muscle3.7 Score Comparison
GP500 versus ANHD
ProbCons Score Comparison
GP500 versus ANHD
% Gappiness
ANHD
GP500 score
0.25
0.5
0.75
1

Figure B.1

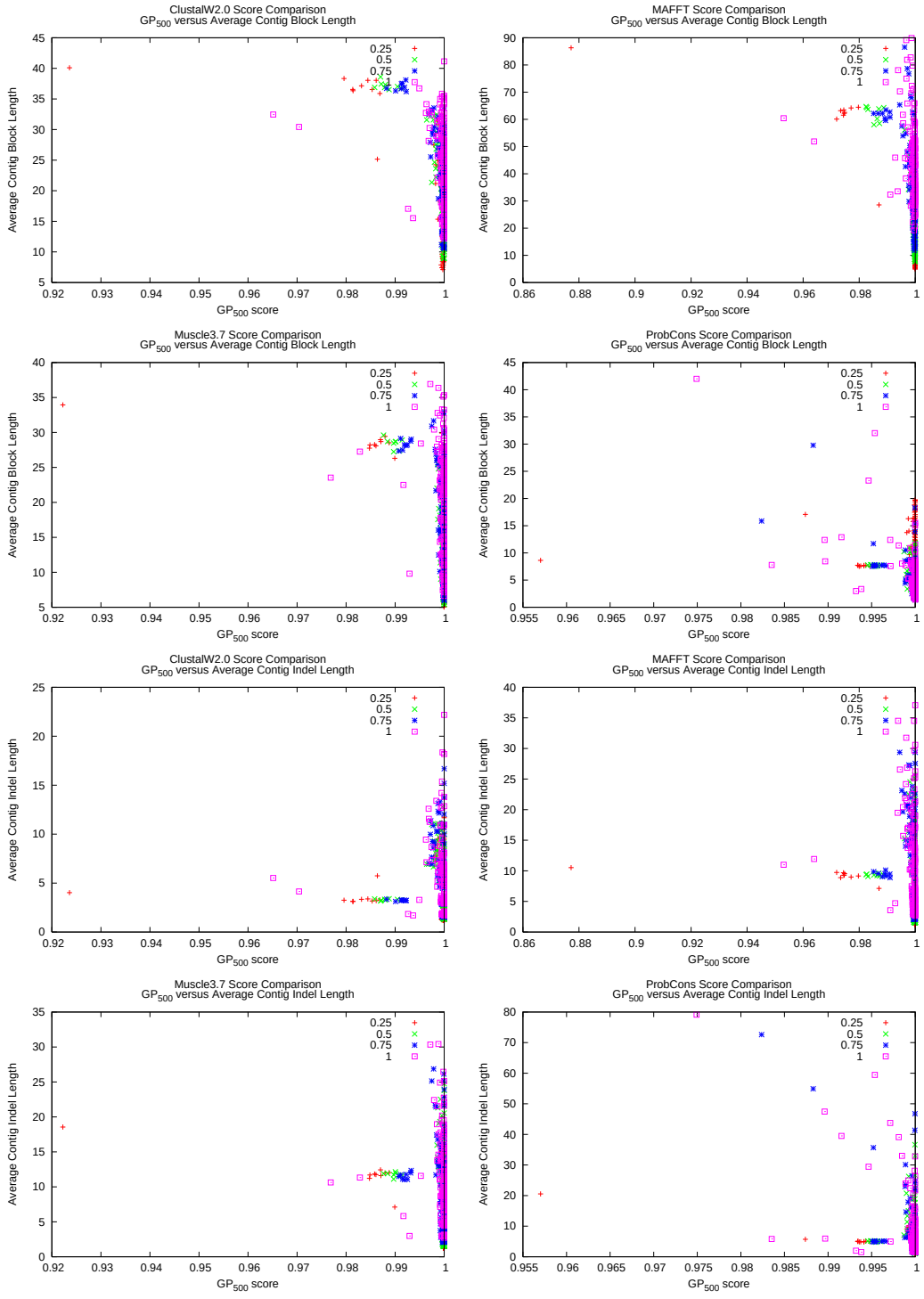

ClustalW2.0 Score Comparison
GP500 versus Average Contig Block Length
MAFFT Score Comparison
GP500 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Block Length
ProbCons Score Comparison
GP500 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP500 versus Average Contig Indel Length
MAFFT Score Comparison
GP500 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Indel Length
ProbCons Score Comparison
GP500 versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
GP500 score
0.25
0.5
0.75
1

Figure B.1

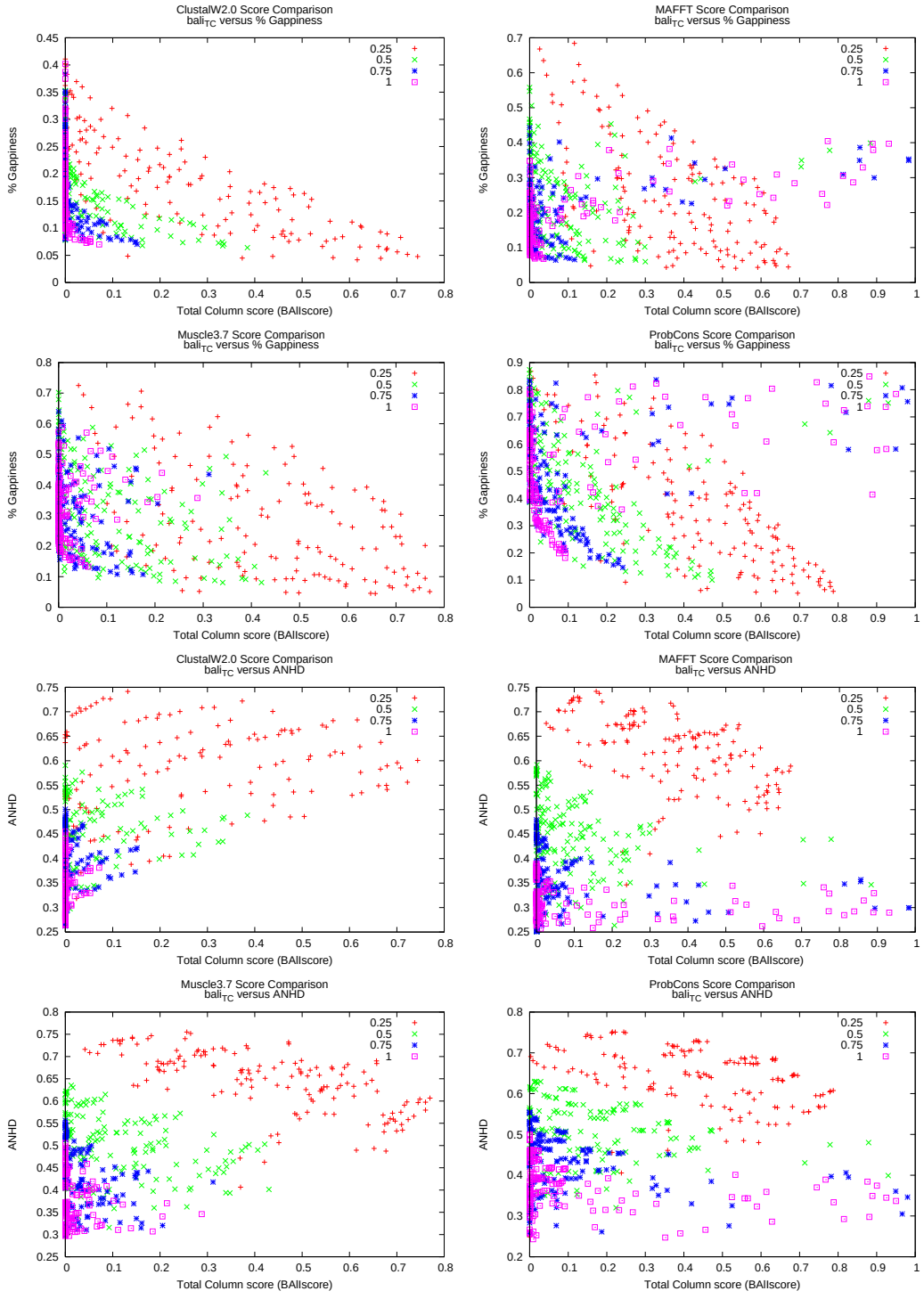

ClustalW2.0 Score Comparison
bali_TC versus % Gappiness
MAFFT Score Comparison
bali_TC versus % Gappiness
Muscle3.7 Score Comparison
bali_TC versus % Gappiness
ProbCons Score Comparison
bali_TC versus % Gappiness
ClustalW2.0 Score Comparison
bali_TC versus ANHD
MAFFT Score Comparison
bali_TC versus ANHD
Muscle3.7 Score Comparison
bali_TC versus ANHD
ProbCons Score Comparison
bali_TC versus ANHD
% Gappiness
ANHD
Total Column score (BAIIscore)
0.25
0.5
0.75
1

Figure B.1

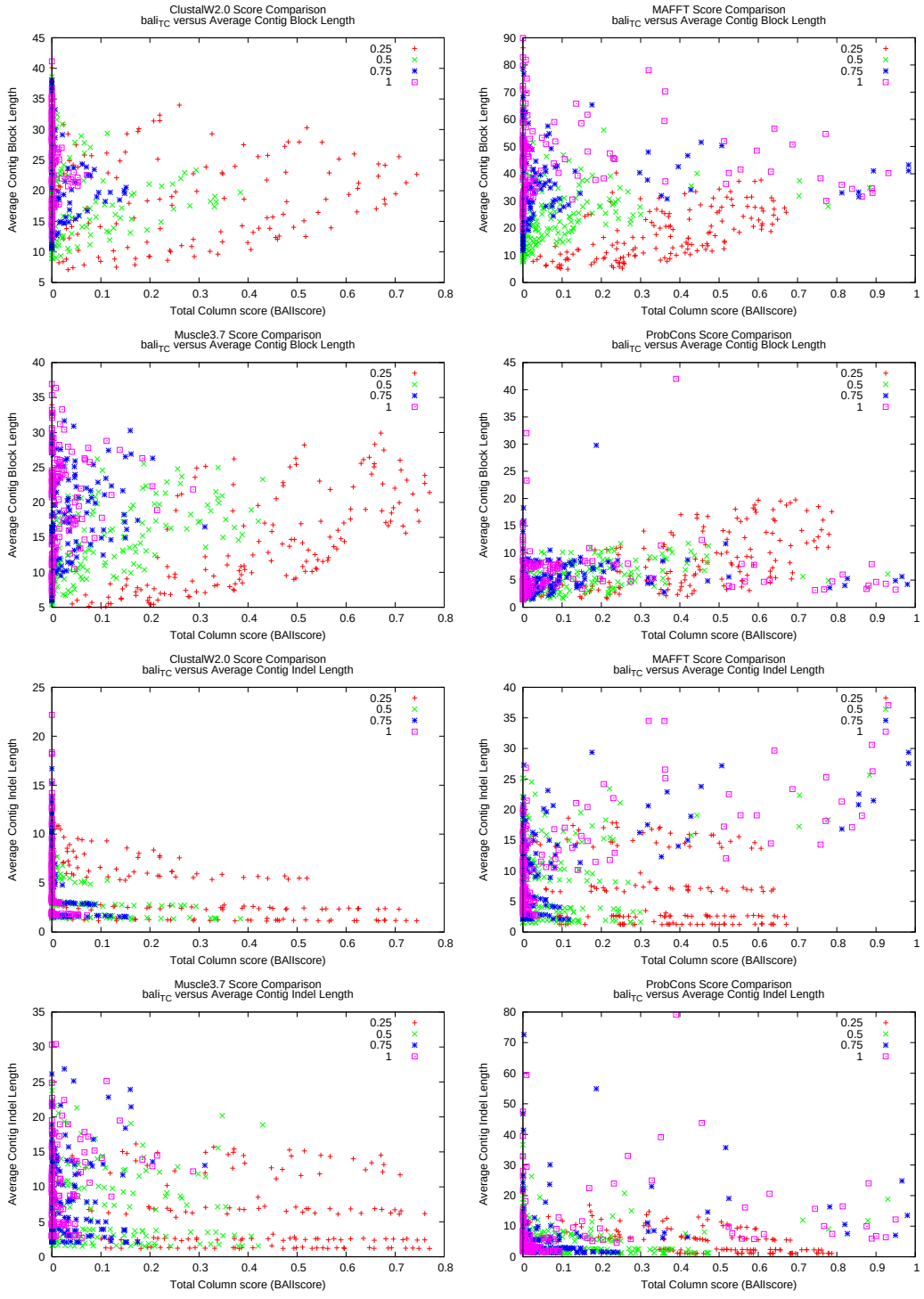

ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Block Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Total Column score (BAIIscore)
0.25
0.5
0.75
1

Figure B.1

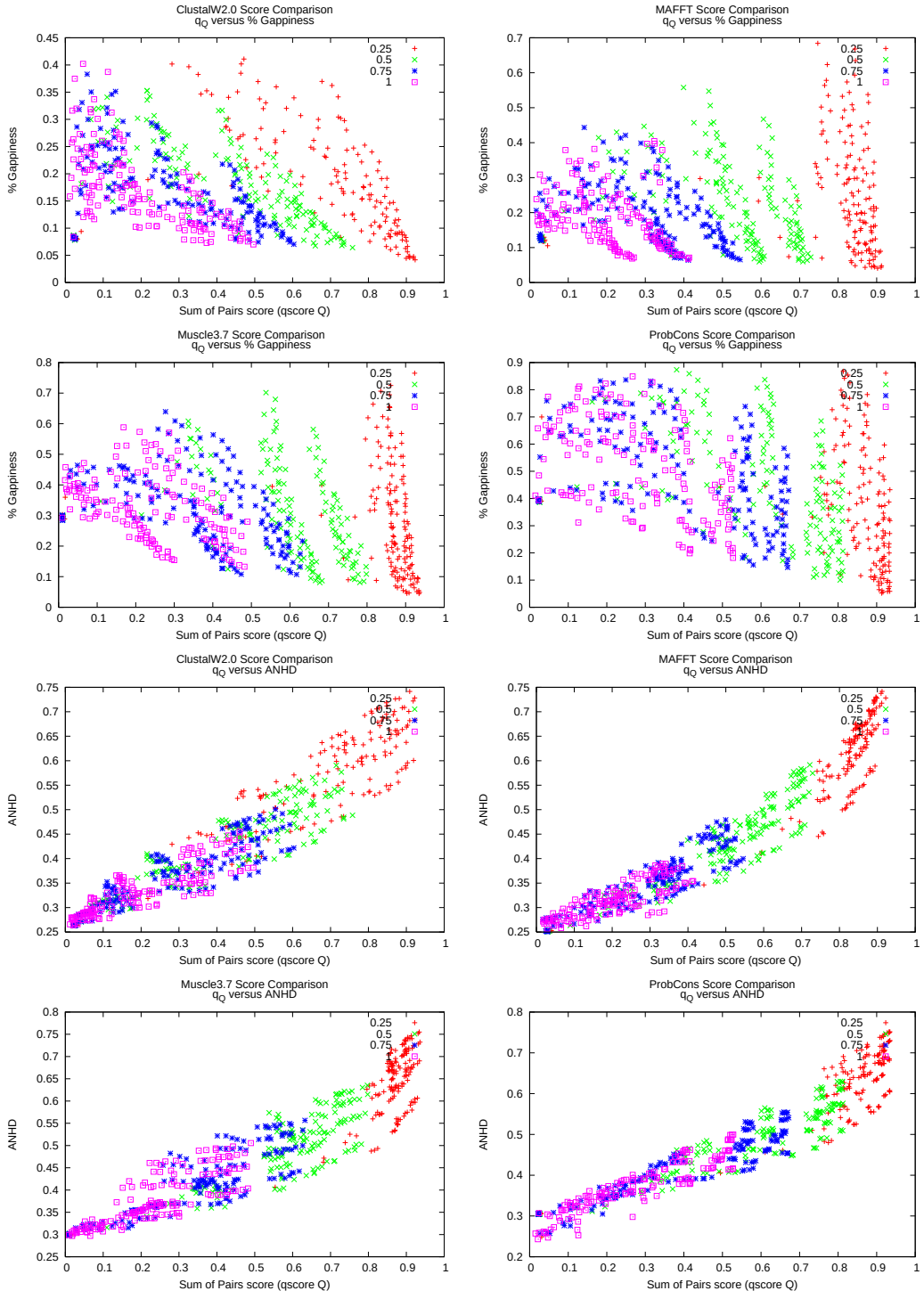

ClustalW2.0 Score Comparison
$q_Q$ versus % Gappiness
MAFFT Score Comparison
$q_Q$ versus % Gappiness
Muscle3.7 Score Comparison
$q_Q$ versus % Gappiness
ProbCons Score Comparison
$q_Q$ versus % Gappiness
ClustalW2.0 Score Comparison
$q_Q$ versus ANHD
MAFFT Score Comparison
$q_Q$ versus ANHD
Muscle3.7 Score Comparison
$q_Q$ versus ANHD
ProbCons Score Comparison
$q_Q$ versus ANHD
% Gappiness
ANHD
Sum of Pairs score (qscore Q)
0.25
0.5
0.75
1

Figure B.1

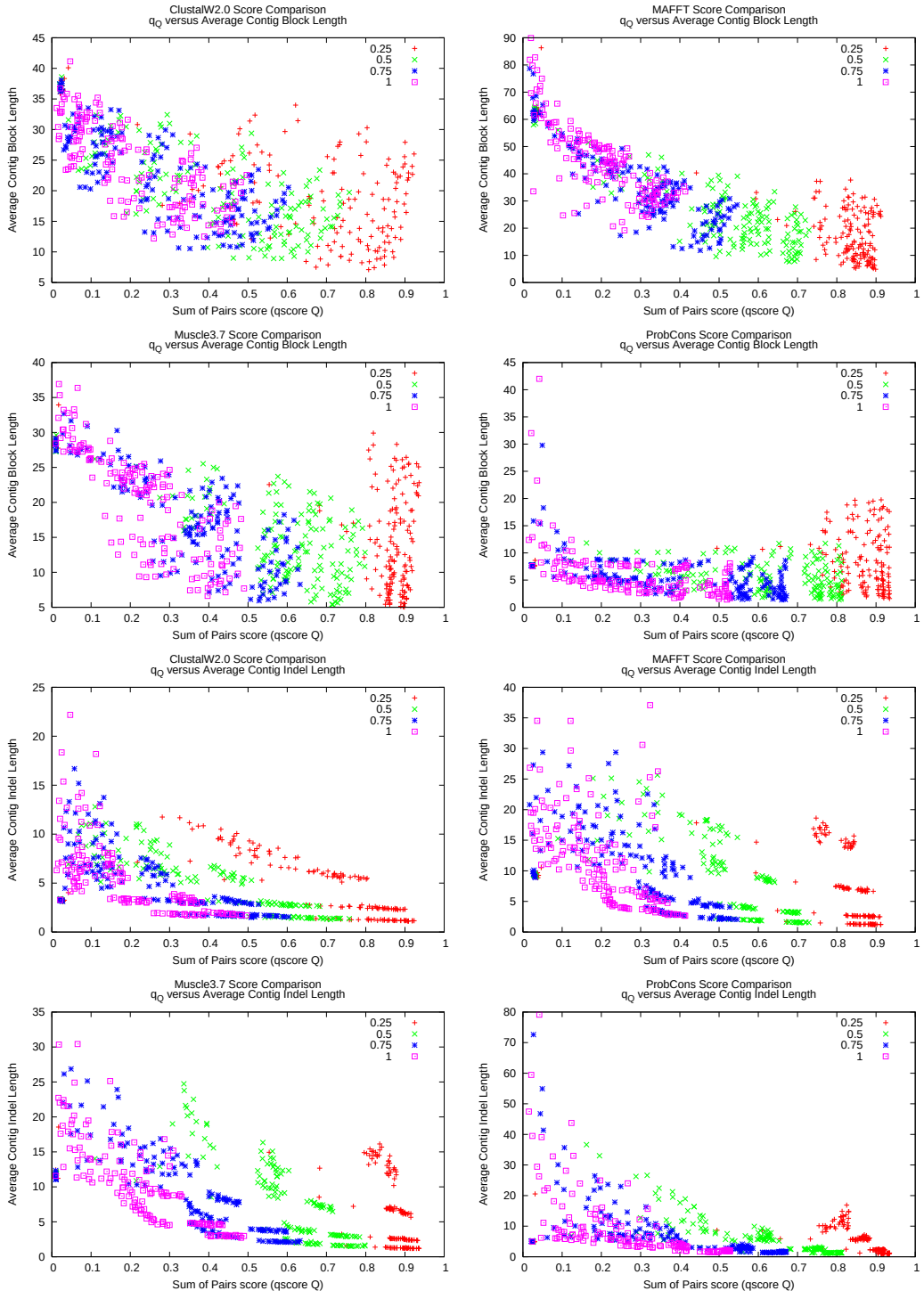

ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Block Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Block Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Indel Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Indel Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Sum of Pairs score (qscore Q)
0.25
0.5
0.75
1

Figure B.1

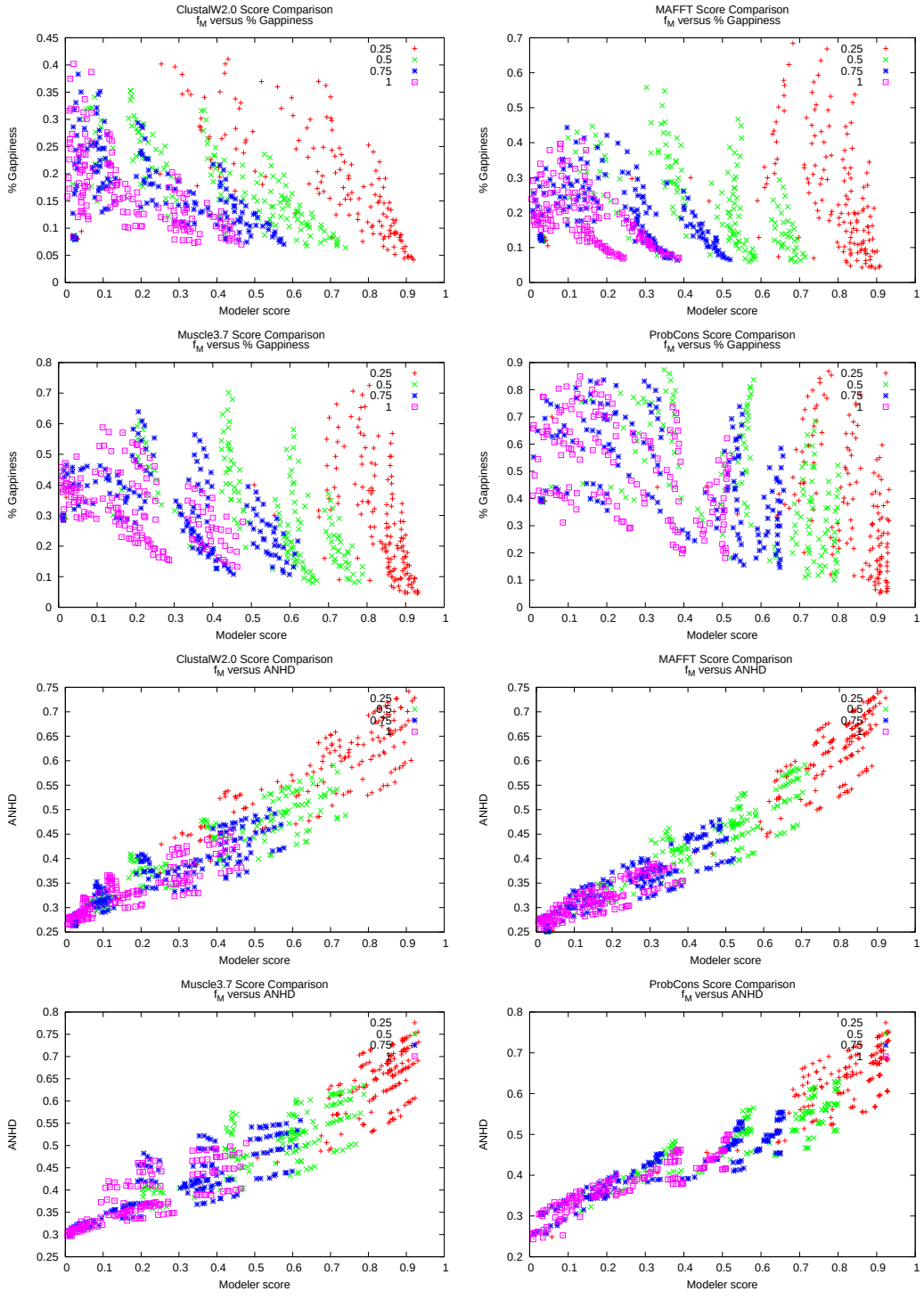

ClustalW2.0 Score Comparison
$f_M$ versus % Gappiness
MAFFT Score Comparison
$f_M$ versus % Gappiness
Muscle3.7 Score Comparison
$f_M$ versus % Gappiness
ProbCons Score Comparison
$f_M$ versus % Gappiness
ClustalW2.0 Score Comparison
$f_M$ versus ANHD
MAFFT Score Comparison
$f_M$ versus ANHD
Muscle3.7 Score Comparison
$f_M$ versus ANHD
ProbCons Score Comparison
$f_M$ versus ANHD
% Gappiness
ANHD
Modeler score
0.25
0.5
0.75
1

Figure B.1

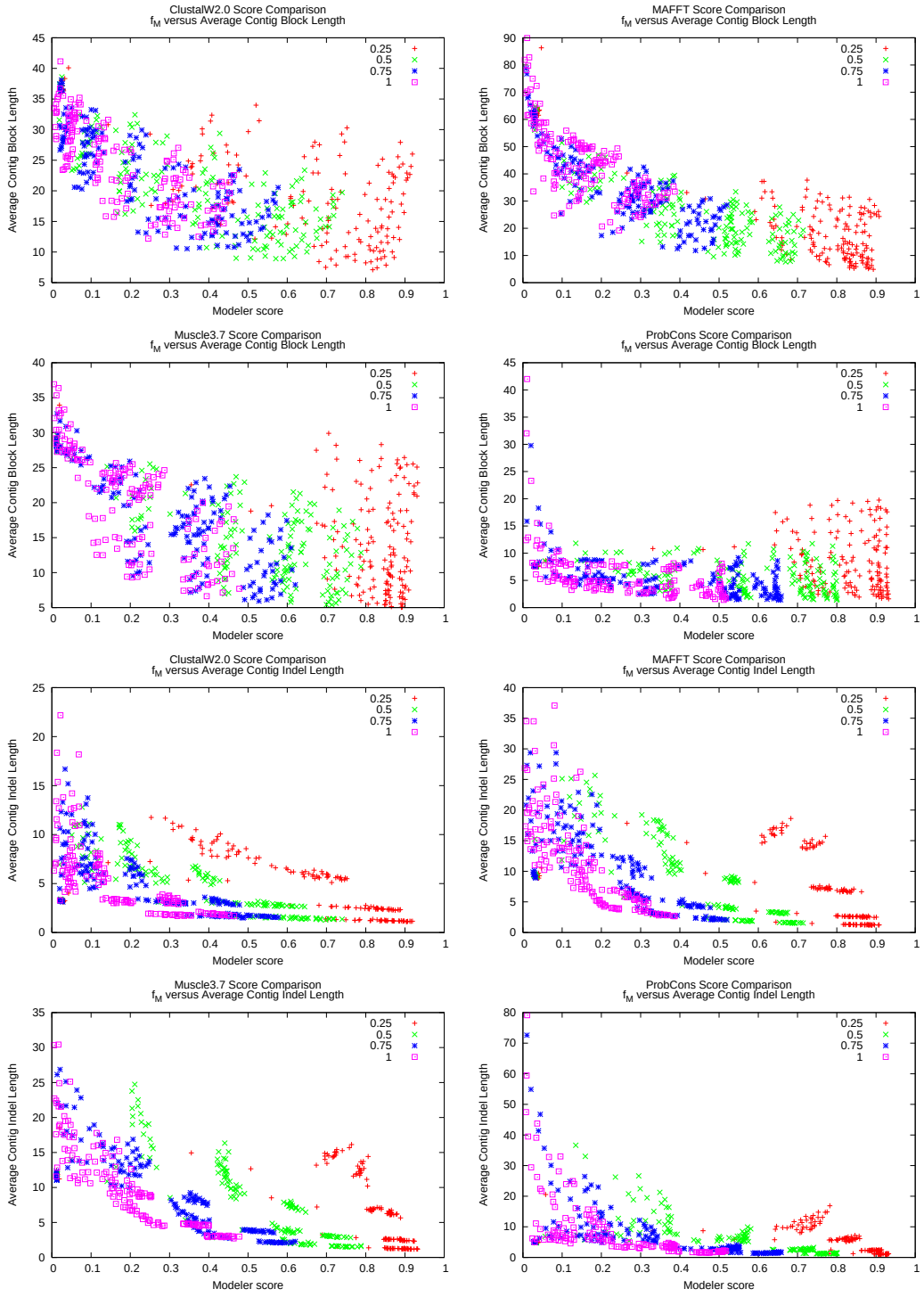

ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Block Length
MAFFT Score Comparison
$f_M$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Block Length
ProbCons Score Comparison
$f_M$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Indel Length
MAFFT Score Comparison
$f_M$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Indel Length
ProbCons Score Comparison
$f_M$ versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Modeler score
0.25
0.5
0.75
1

Figure B.1

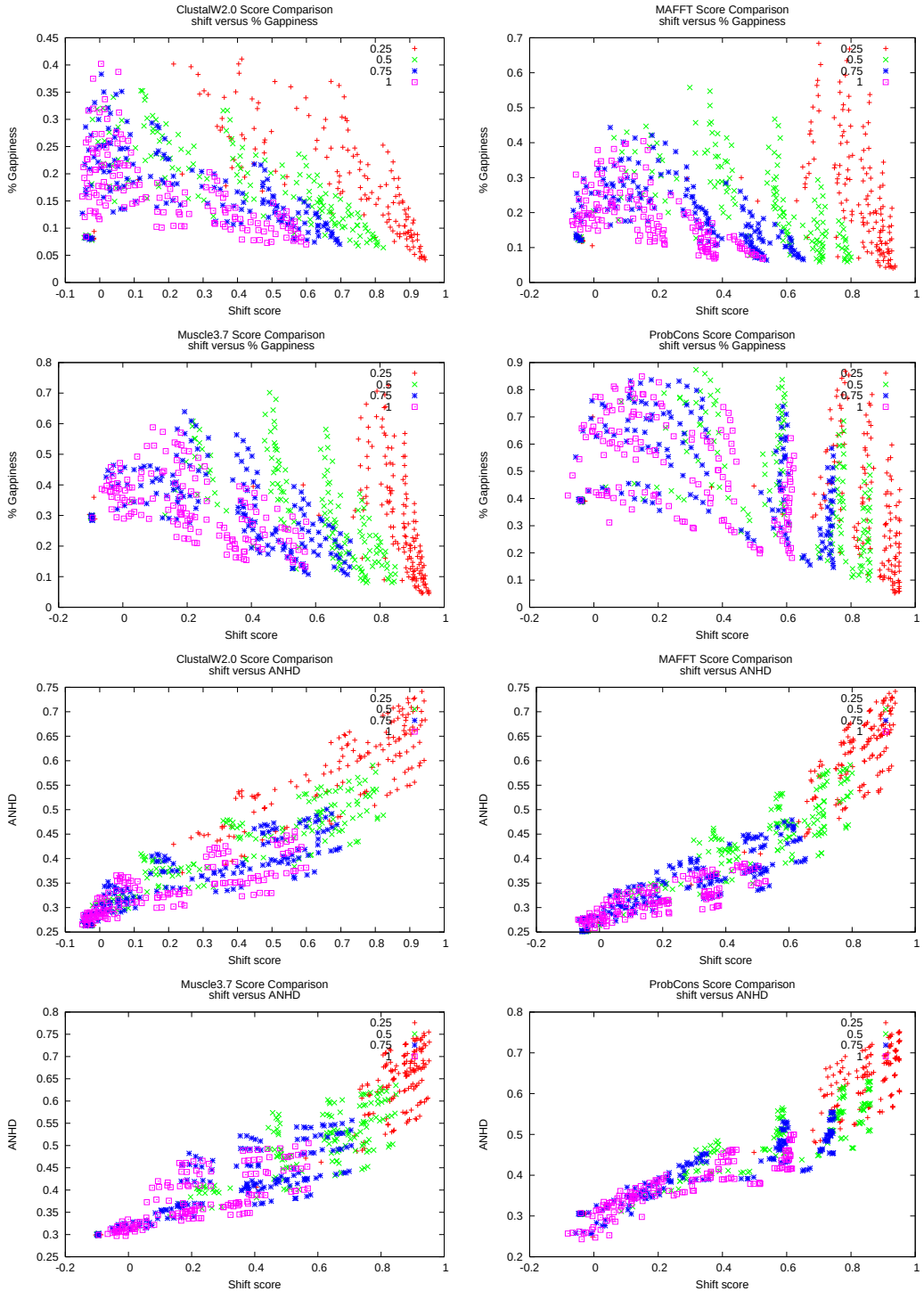

ClustalW2.0 Score Comparison
shift versus % Gappiness
MAFFT Score Comparison
shift versus % Gappiness
Muscle3.7 Score Comparison
shift versus % Gappiness
ProbCons Score Comparison
shift versus % Gappiness
ClustalW2.0 Score Comparison
shift versus ANHD
MAFFT Score Comparison
shift versus ANHD
Muscle3.7 Score Comparison
shift versus ANHD
ProbCons Score Comparison
shift versus ANHD
% Gappiness
ANHD
Shift score
0.25
0.5
0.75
1

Figure B.1

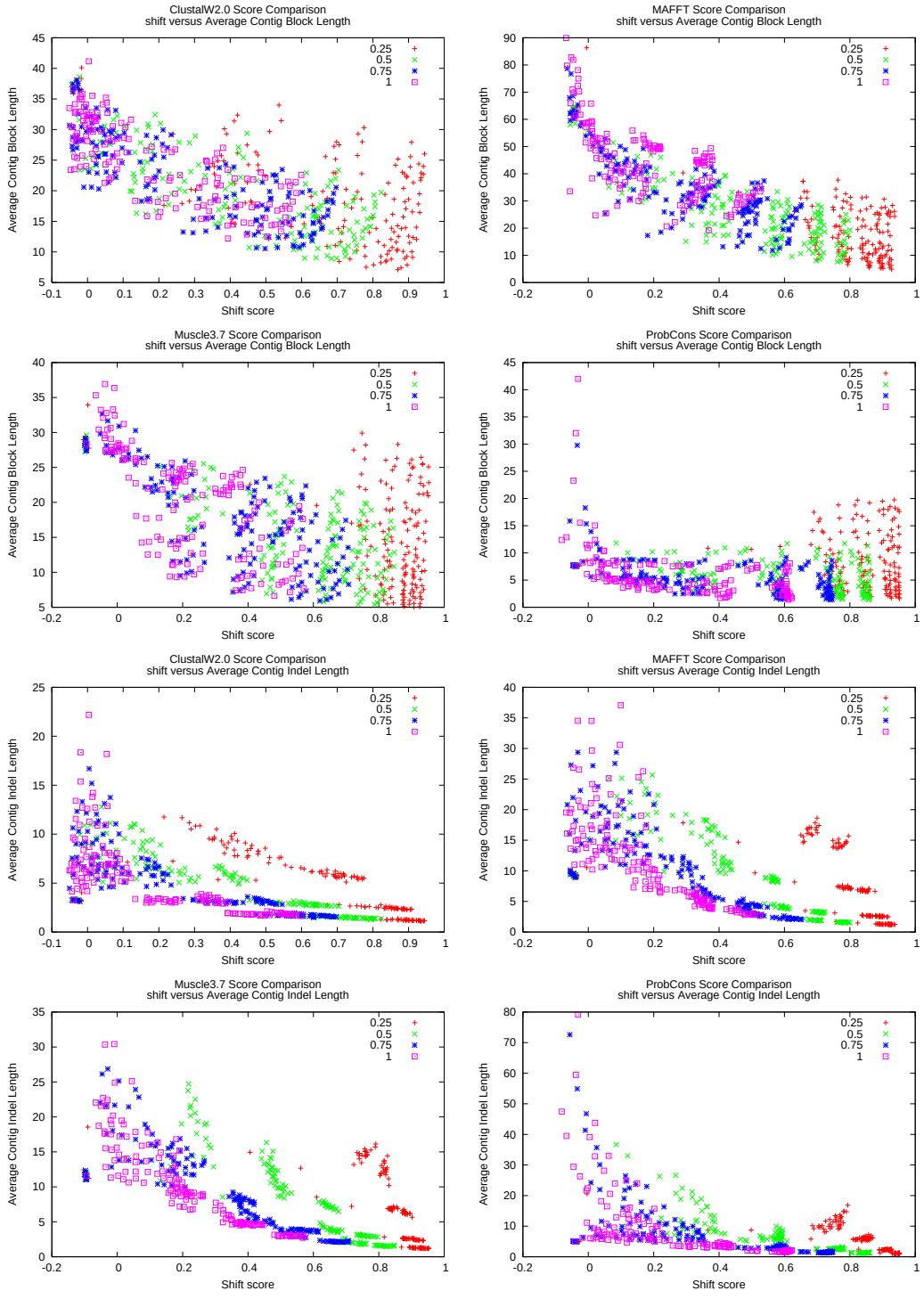

ClustalW2.0 Score Comparison
shift versus Average Contig Block Length
MAFFT Score Comparison
shift versus Average Contig Block Length
Muscle3.7 Score Comparison
shift versus Average Contig Block Length
ProbCons Score Comparison
shift versus Average Contig Block Length
ClustalW2.0 Score Comparison
shift versus Average Contig Indel Length
MAFFT Score Comparison
shift versus Average Contig Indel Length
Muscle3.7 Score Comparison
shift versus Average Contig Indel Length
ProbCons Score Comparison
shift versus Average Contig Indel Length
Average Contig Block Length
Average Contig Indel Length
Shift score
0.25
0.5
0.75
1

Figure B.1

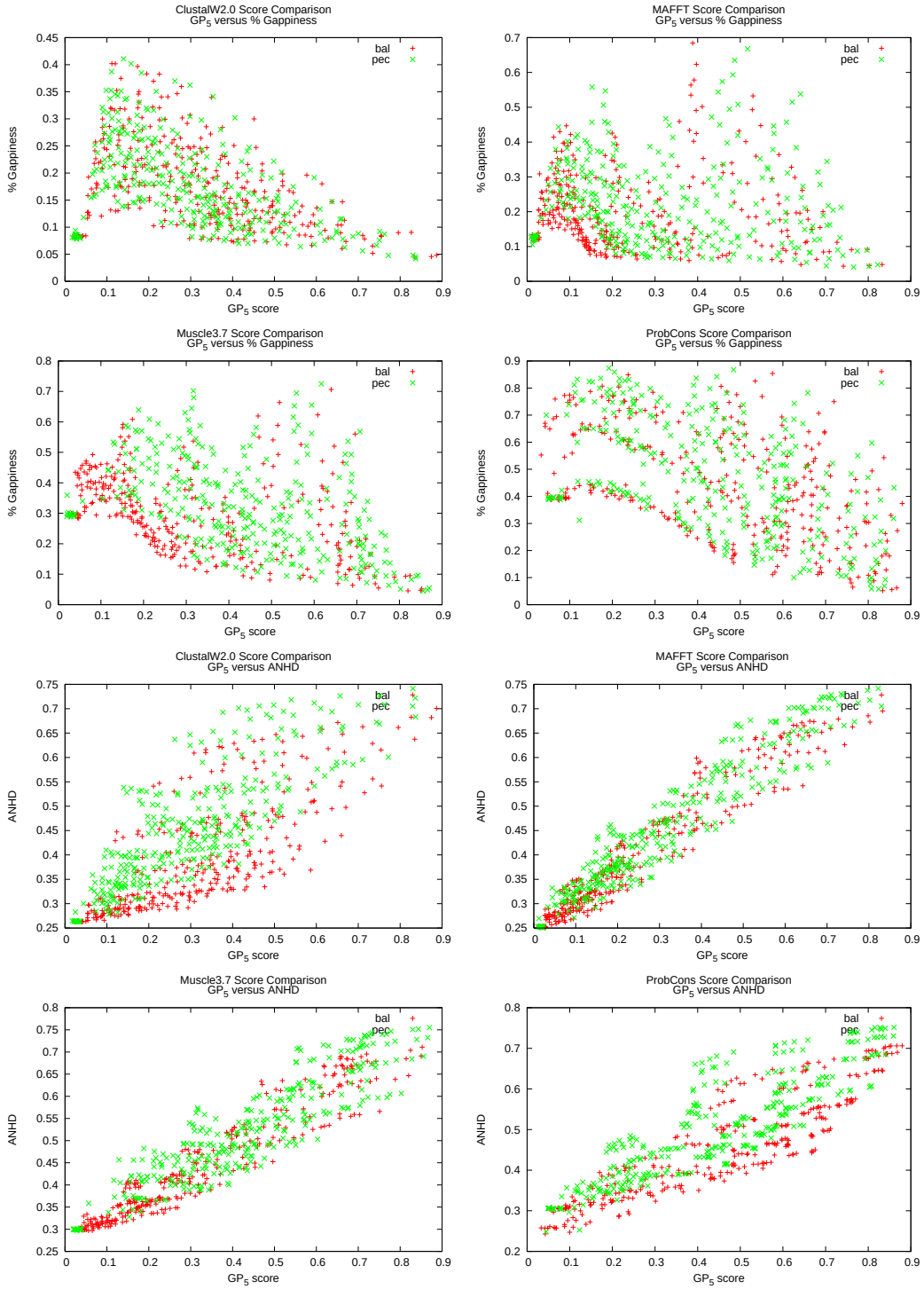

ClustalW2.0 Score Comparison
GP5 versus % Gappiness
MAFFT Score Comparison
GP5 versus % Gappiness
Muscle3.7 Score Comparison
GP5 versus % Gappiness
ProbCons Score Comparison
GP5 versus % Gappiness
ClustalW2.0 Score Comparison
GP5 versus ANHD
MAFFT Score Comparison
GP5 versus ANHD
Muscle3.7 Score Comparison
GP5 versus ANHD
ProbCons Score Comparison
GP5 versus ANHD
bal
pec
% Gappiness
ANHD
GP5 score

Figure B.1

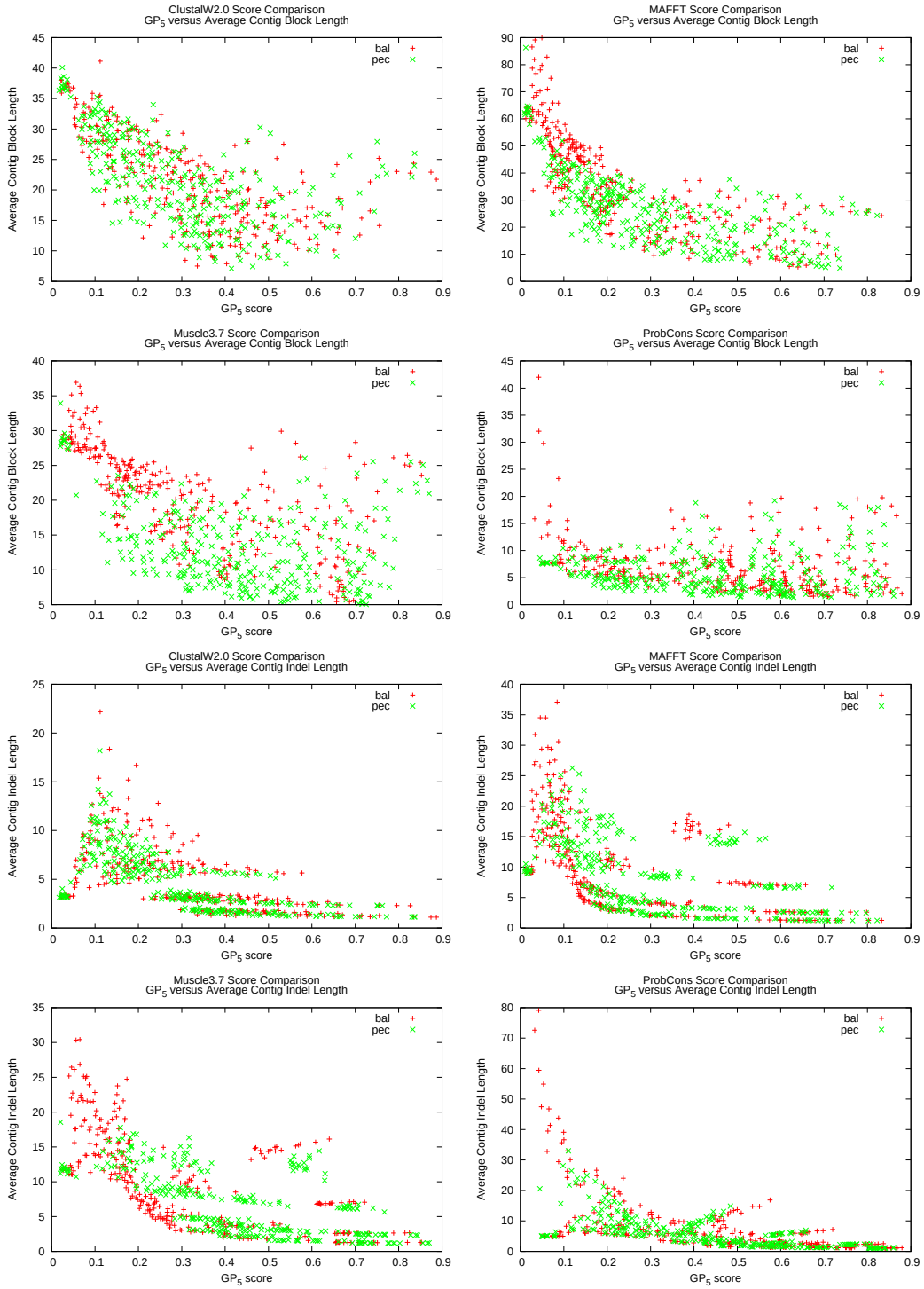

ClustalW2.0 Score Comparison
GP5 versus Average Contig Block Length
MAFFT Score Comparison
GP5 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Block Length
ProbCons Score Comparison
GP5 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP5 versus Average Contig Indel Length
MAFFT Score Comparison
GP5 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP5 versus Average Contig Indel Length
ProbCons Score Comparison
GP5 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP5 score

Figure B.1

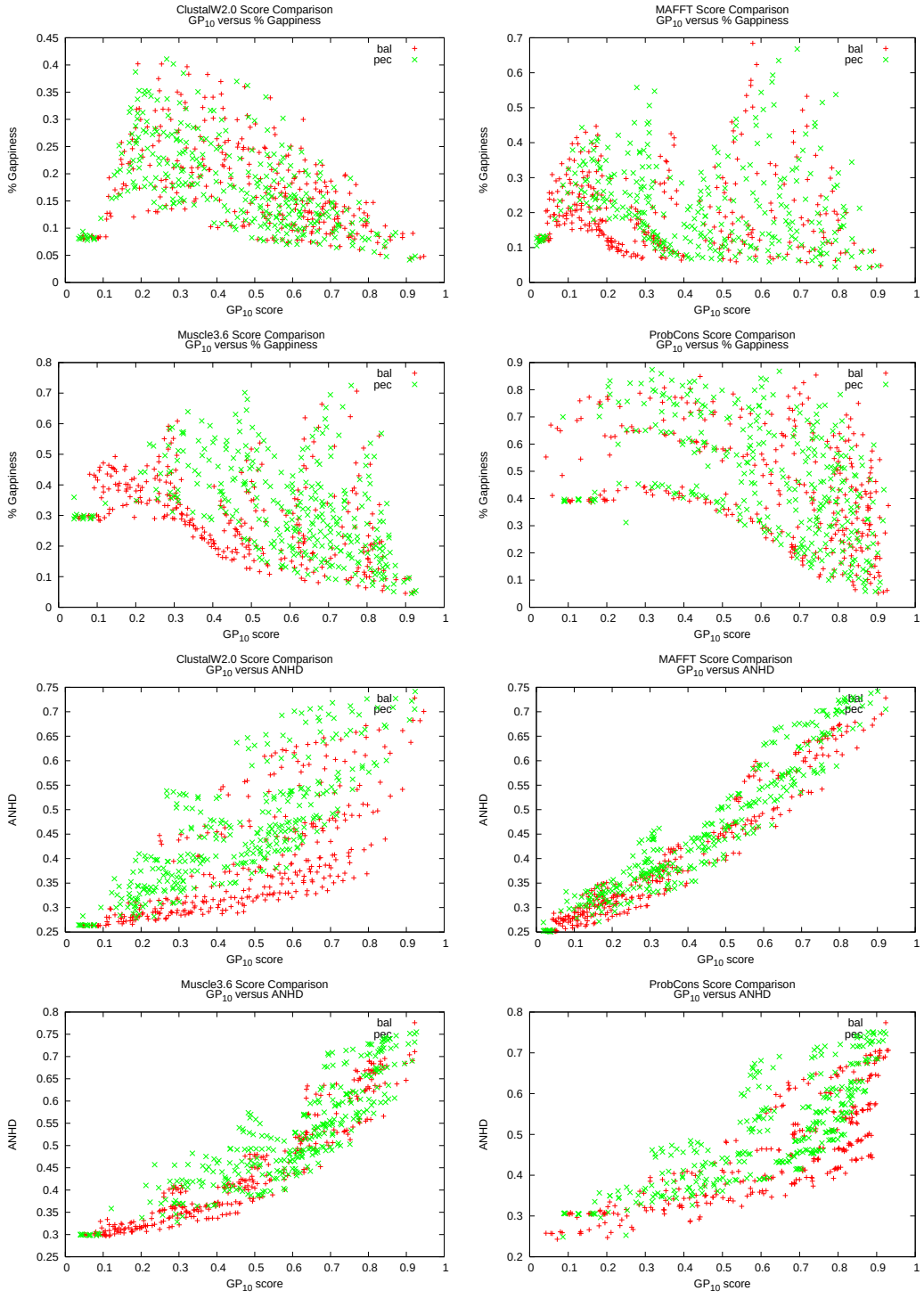

ClustalW2.0 Score Comparison
GP10 versus % Gappiness
MAFFT Score Comparison
GP10 versus % Gappiness
Muscle3.6 Score Comparison
GP10 versus % Gappiness
ProbCons Score Comparison
GP10 versus % Gappiness
ClustalW2.0 Score Comparison
GP10 versus ANHD
MAFFT Score Comparison
GP10 versus ANHD
Muscle3.6 Score Comparison
GP10 versus ANHD
ProbCons Score Comparison
GP10 versus ANHD
% Gappiness
ANHD
GP10 score
bal
pec

Figure B.1

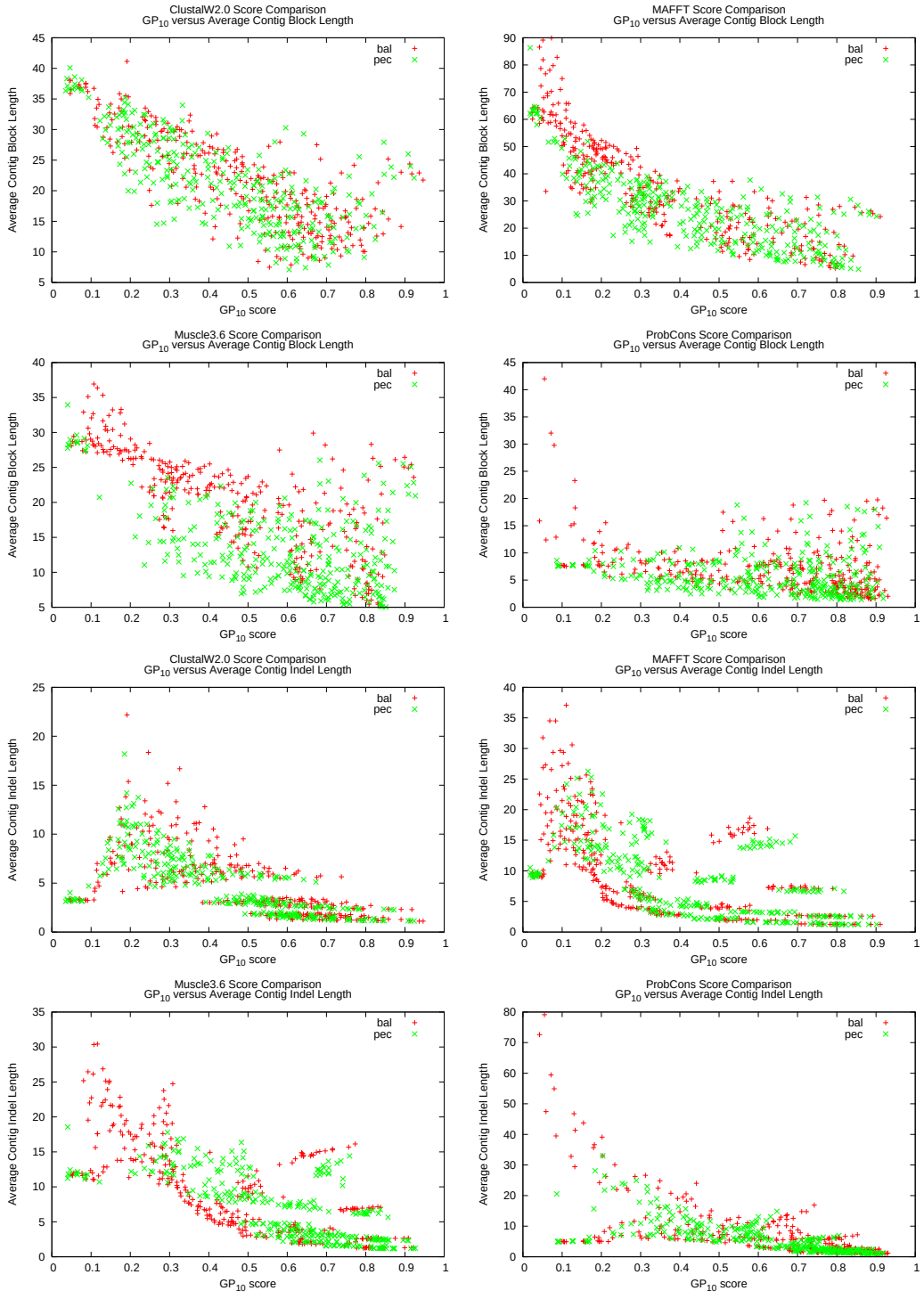

ClustalW2.0 Score Comparison
GP10 versus Average Contig Block Length
MAFFT Score Comparison
GP10 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Block Length
ProbCons Score Comparison
GP10 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP10 versus Average Contig Indel Length
MAFFT Score Comparison
GP10 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP10 versus Average Contig Indel Length
ProbCons Score Comparison
GP10 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP10 score

Figure B.1

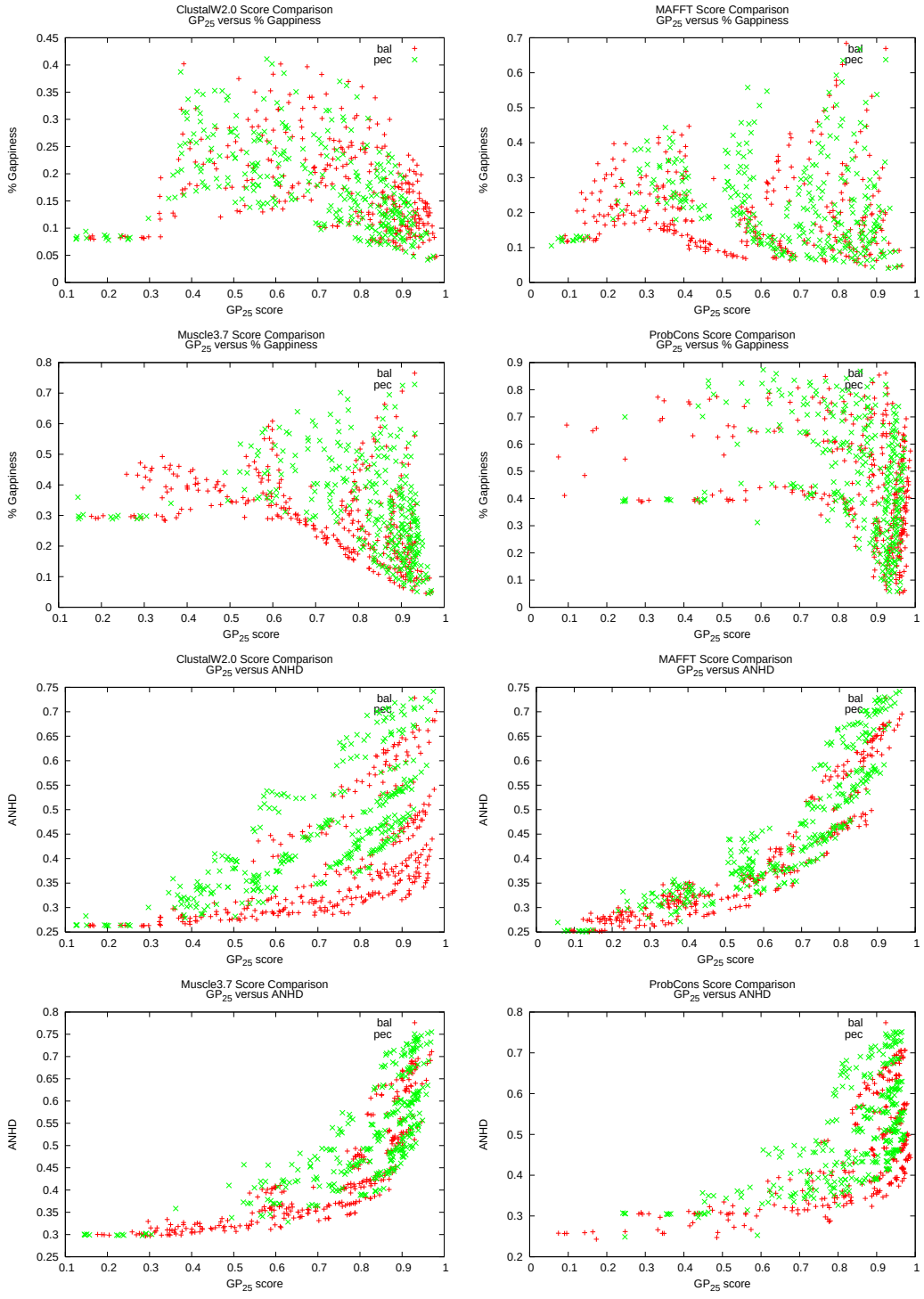

ClustalW2.0 Score Comparison
GP25 versus % Gappiness
MAFFT Score Comparison
GP25 versus % Gappiness
Muscle3.7 Score Comparison
GP25 versus % Gappiness
ProbCons Score Comparison
GP25 versus % Gappiness
ClustalW2.0 Score Comparison
GP25 versus ANHD
MAFFT Score Comparison
GP25 versus ANHD
Muscle3.7 Score Comparison
GP25 versus ANHD
ProbCons Score Comparison
GP25 versus ANHD
% Gappiness
ANHD
GP25 score
bal
pec

Figure B.1

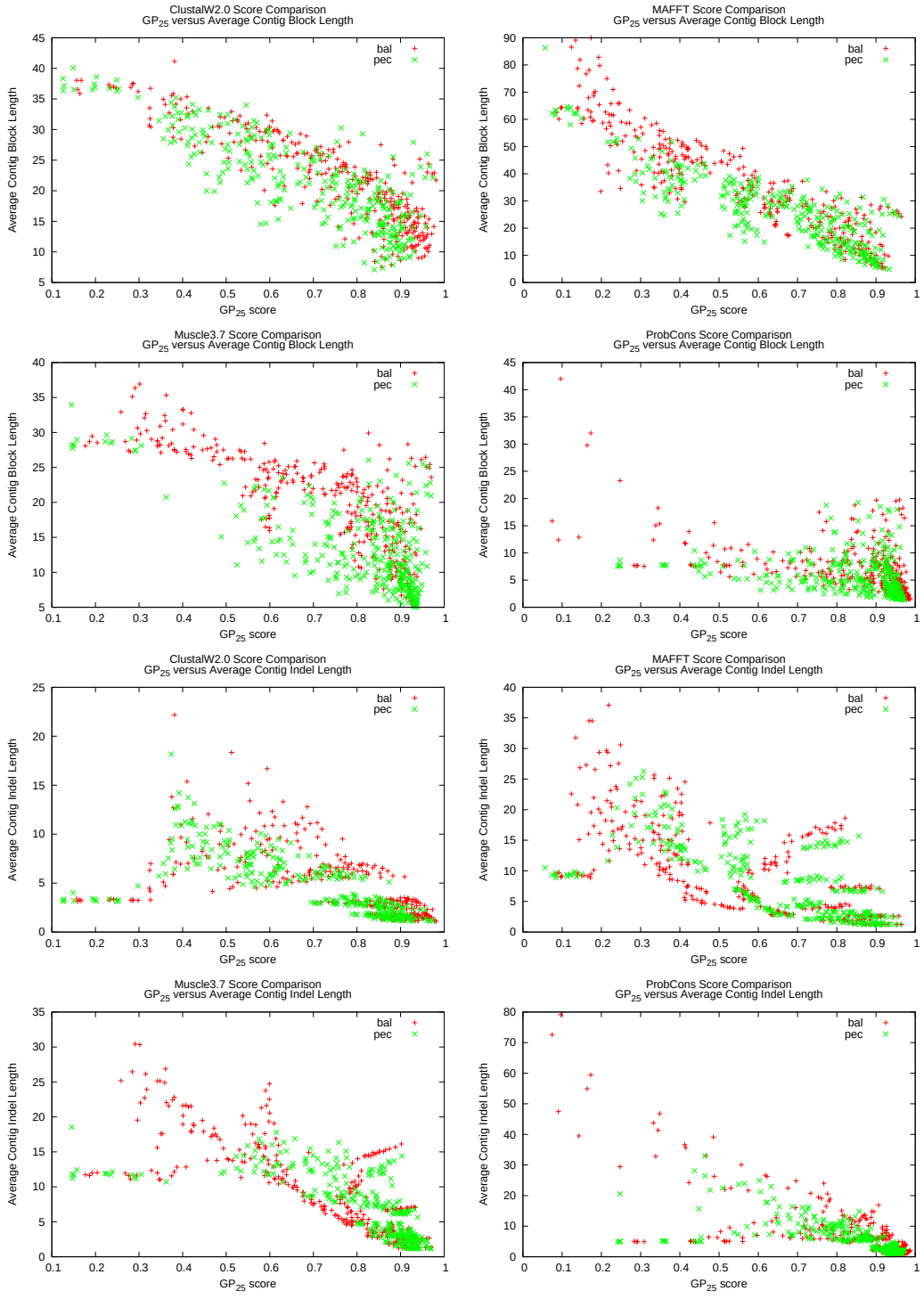

ClustalW2.0 Score Comparison
GP25 versus Average Contig Block Length
MAFFT Score Comparison
GP25 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Block Length
ProbCons Score Comparison
GP25 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP25 versus Average Contig Indel Length
MAFFT Score Comparison
GP25 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP25 versus Average Contig Indel Length
ProbCons Score Comparison
GP25 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP25 score

Figure B.1

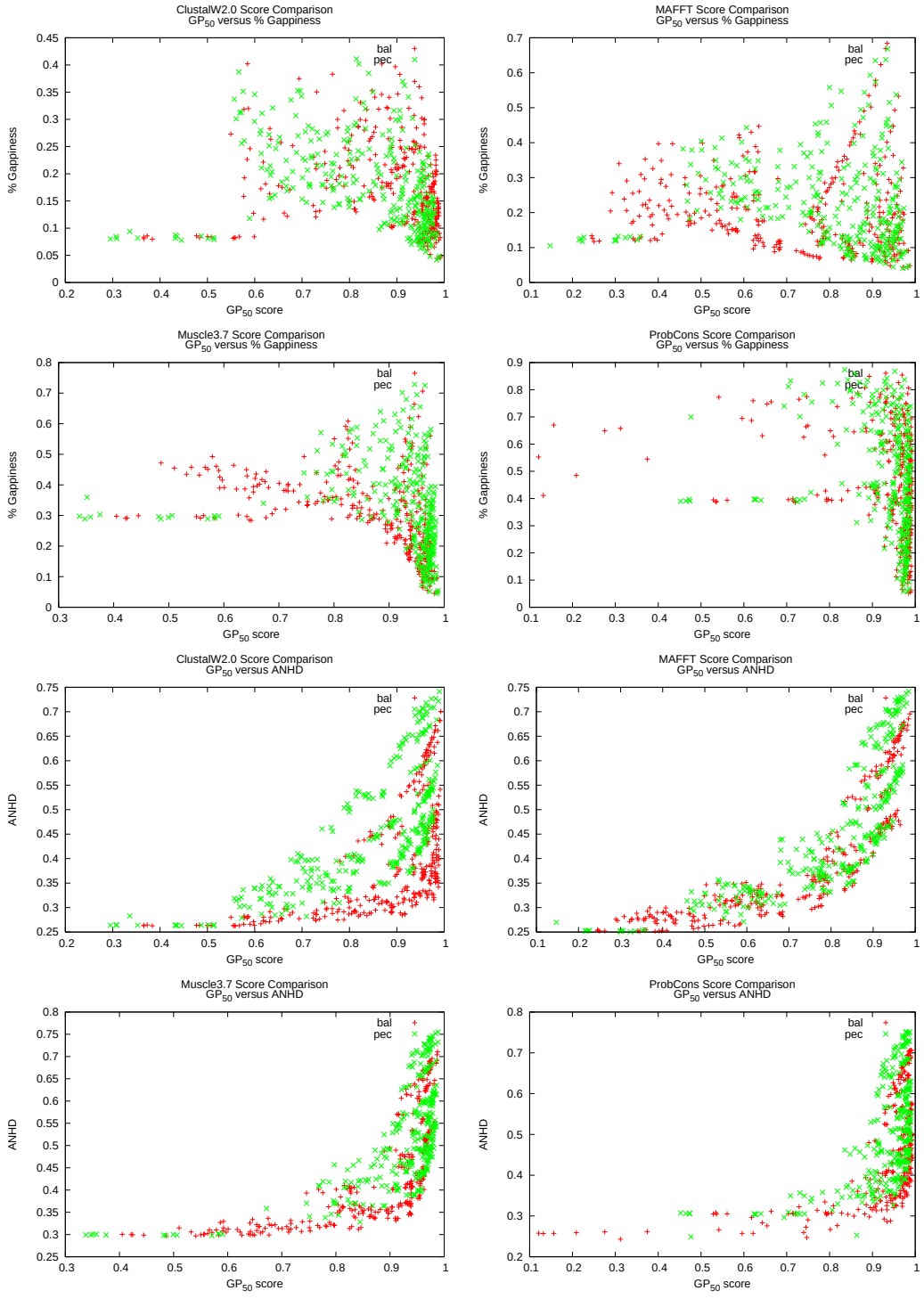

ClustalW2.0 Score Comparison
GP50 versus % Gappiness
MAFFT Score Comparison
GP50 versus % Gappiness
Muscle3.7 Score Comparison
GP50 versus % Gappiness
ProbCons Score Comparison
GP50 versus % Gappiness
ClustalW2.0 Score Comparison
GP50 versus ANHD
MAFFT Score Comparison
GP50 versus ANHD
Muscle3.7 Score Comparison
GP50 versus ANHD
ProbCons Score Comparison
GP50 versus ANHD
% Gappiness
ANHD
GP50 score
bal
pec

Figure B.1

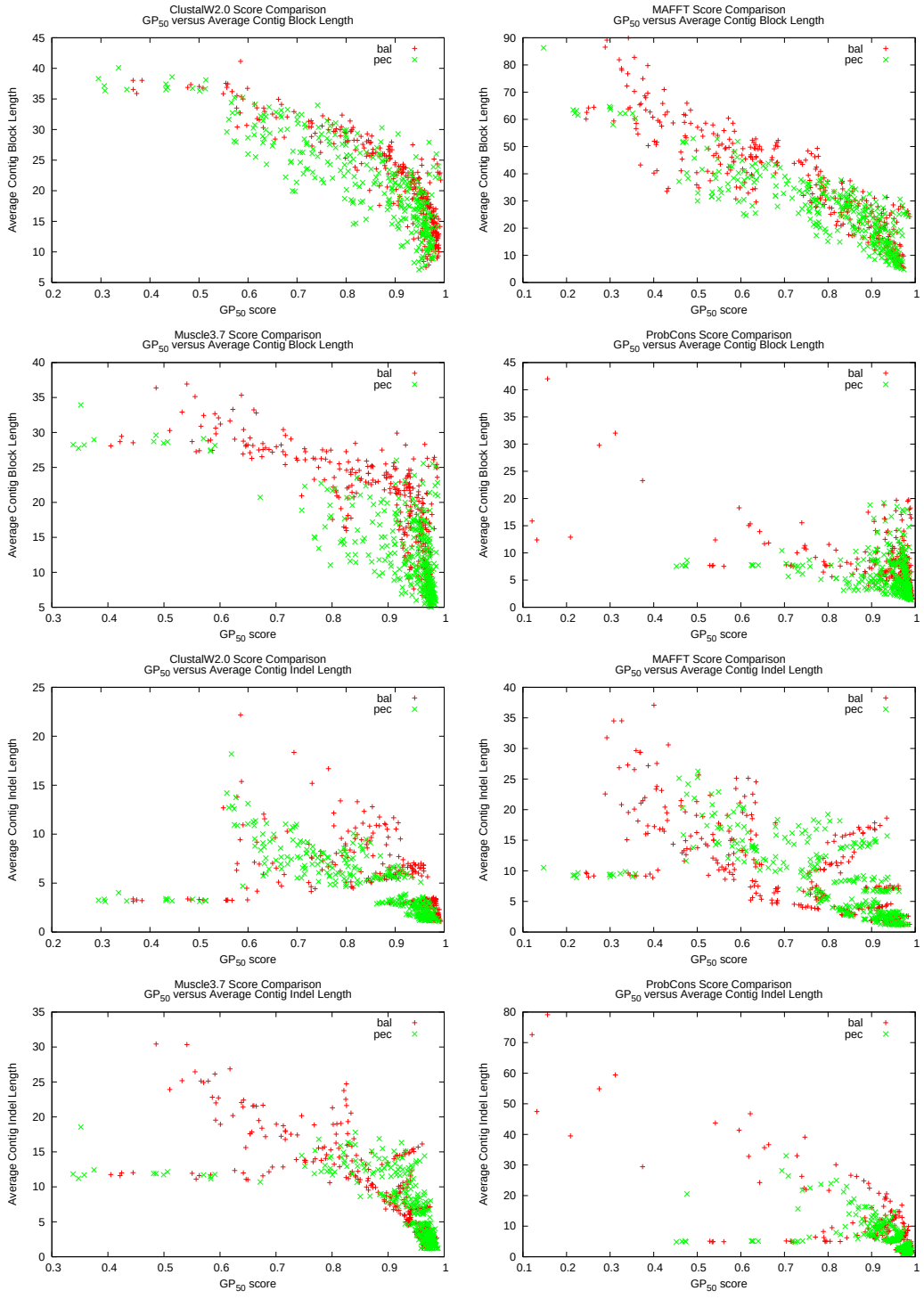

ClustalW2.0 Score Comparison
GP50 versus Average Contig Block Length
MAFFT Score Comparison
GP50 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Block Length
ProbCons Score Comparison
GP50 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP50 versus Average Contig Indel Length
MAFFT Score Comparison
GP50 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP50 versus Average Contig Indel Length
ProbCons Score Comparison
GP50 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP50 score

Figure B.1

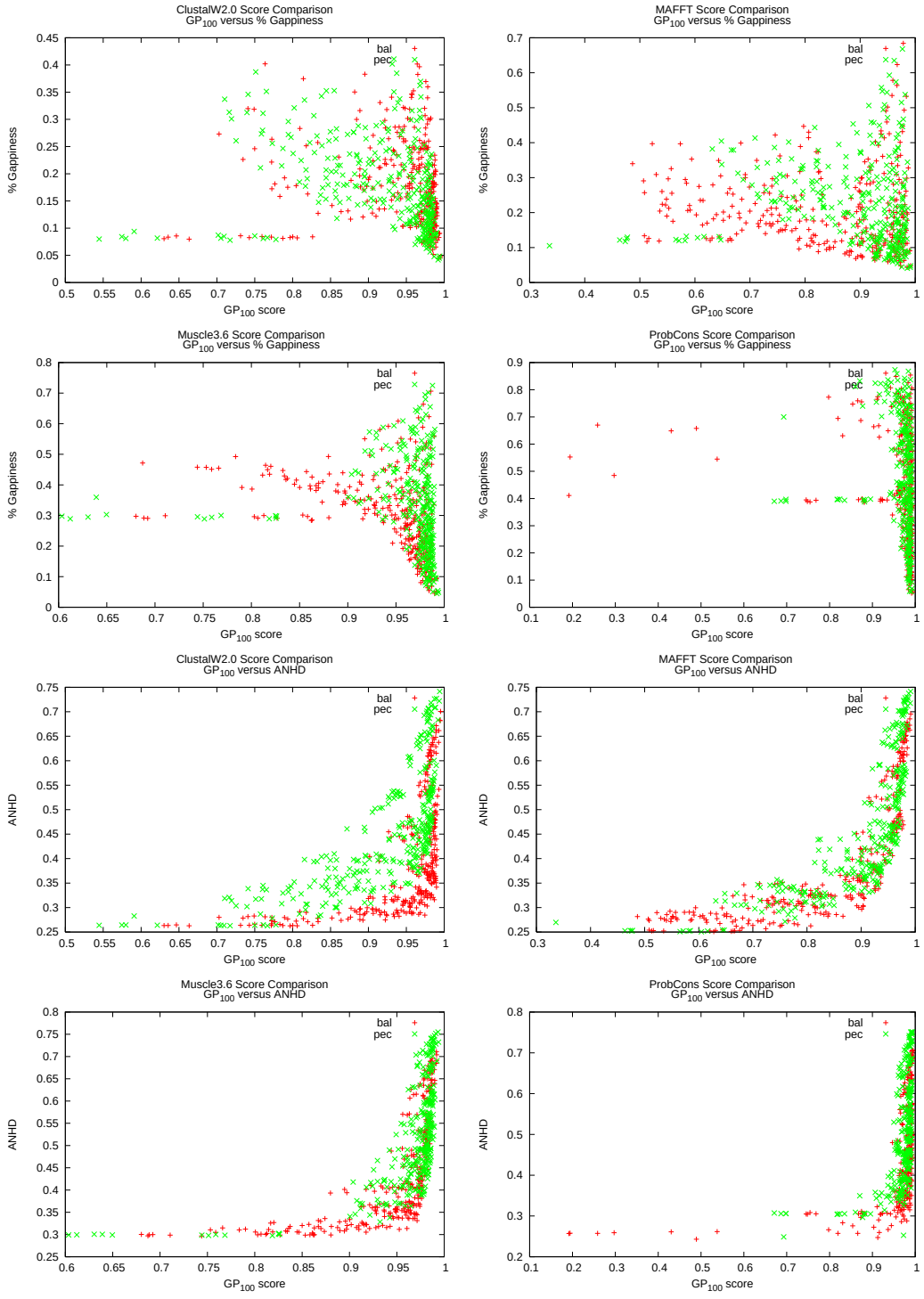

ClustalW2.0 Score Comparison
GP100 versus % Gappiness
MAFFT Score Comparison
GP100 versus % Gappiness
Muscle3.6 Score Comparison
GP100 versus % Gappiness
ProbCons Score Comparison
GP100 versus % Gappiness
ClustalW2.0 Score Comparison
GP100 versus ANHD
MAFFT Score Comparison
GP100 versus ANHD
Muscle3.6 Score Comparison
GP100 versus ANHD
ProbCons Score Comparison
GP100 versus ANHD
bal
pec
% Gappiness
ANHD
GP100 score

Figure B.1

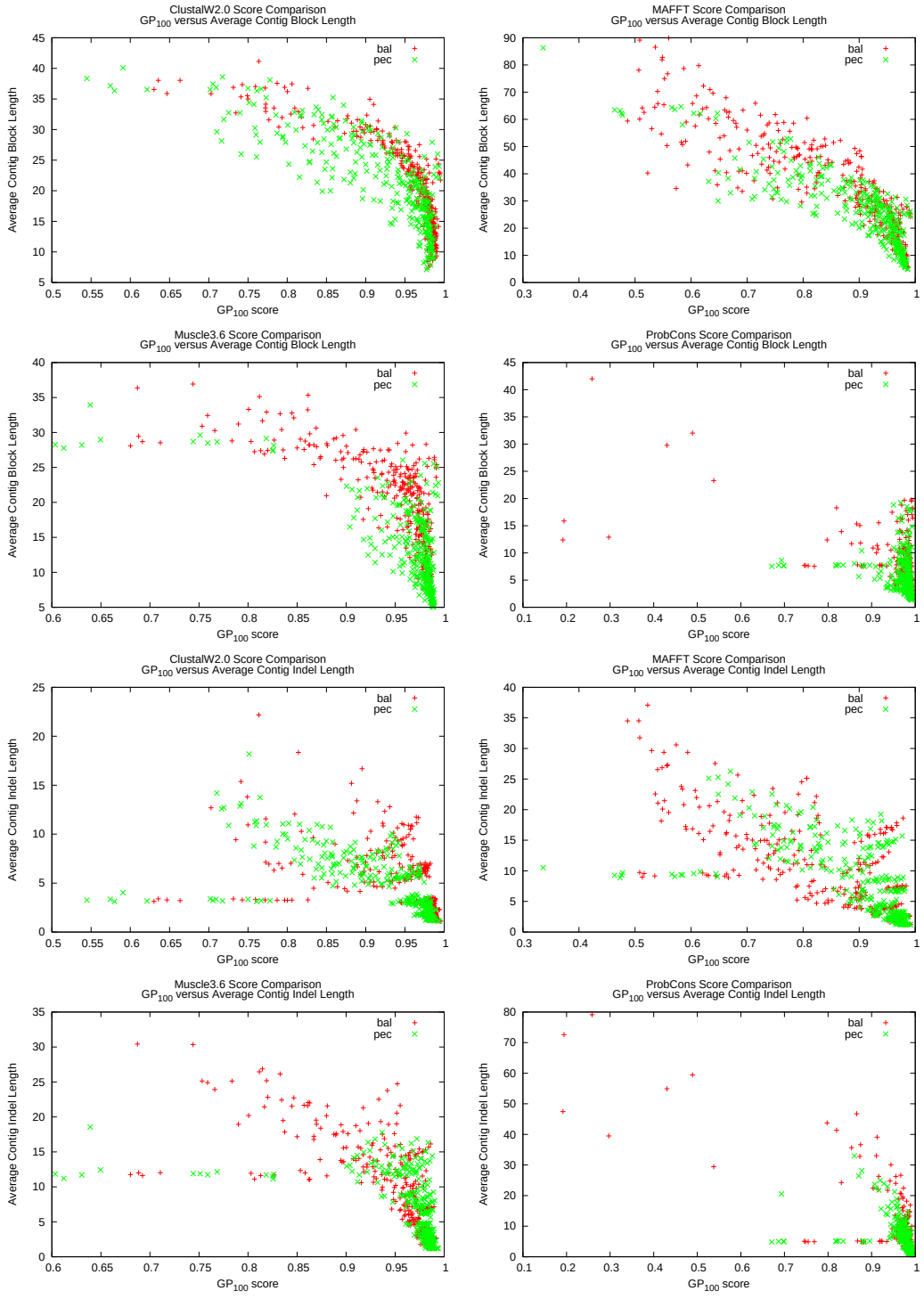

ClustalW2.0 Score Comparison
GP100 versus Average Contig Block Length
MAFFT Score Comparison
GP100 versus Average Contig Block Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Block Length
ProbCons Score Comparison
GP100 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP100 versus Average Contig Indel Length
MAFFT Score Comparison
GP100 versus Average Contig Indel Length
Muscle3.6 Score Comparison
GP100 versus Average Contig Indel Length
ProbCons Score Comparison
GP100 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP100 score

Figure B.1

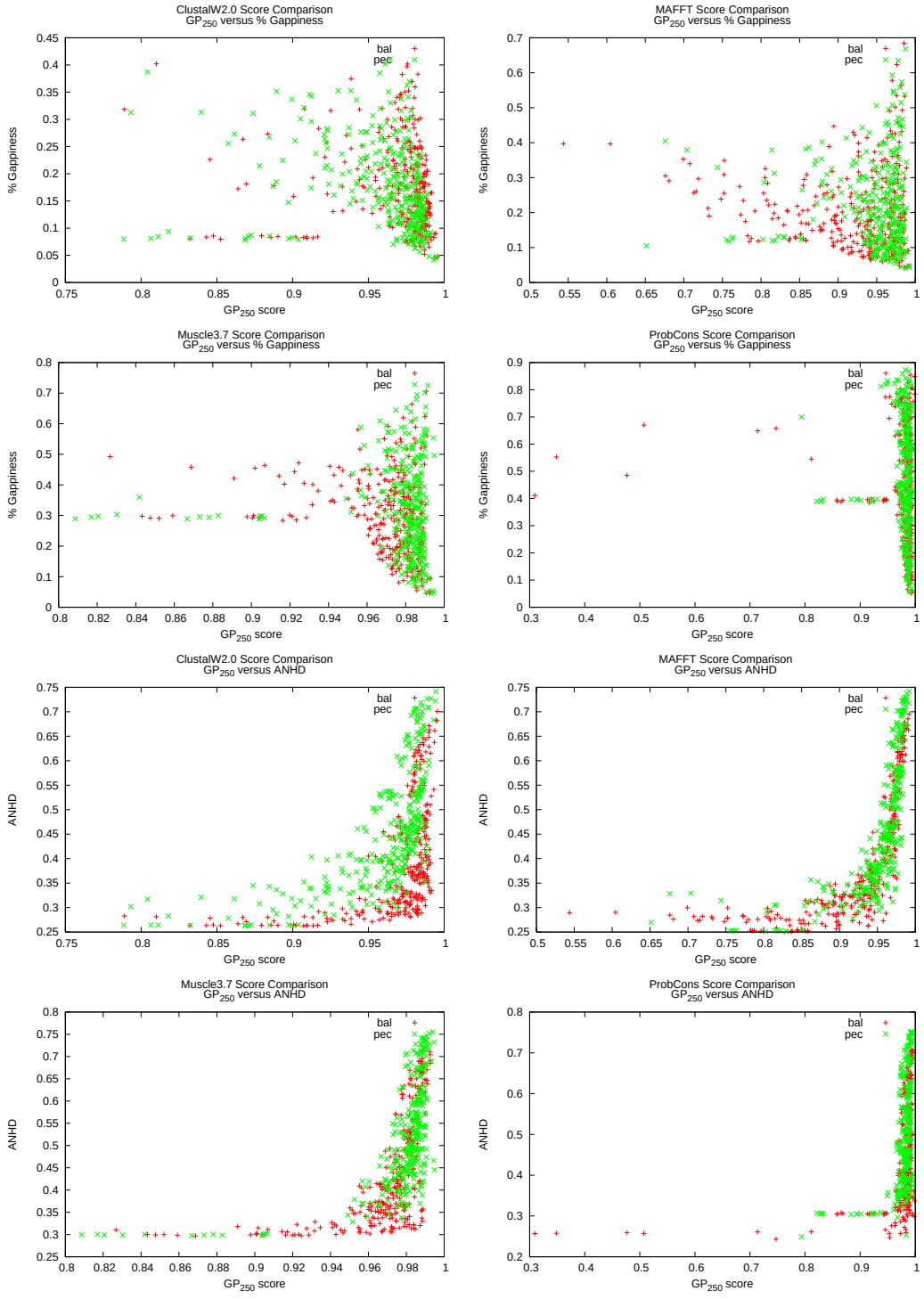

ClustalW2.0 Score Comparison
GP250 versus % Gappiness
MAFFT Score Comparison
GP250 versus % Gappiness
Muscle3.7 Score Comparison
GP250 versus % Gappiness
ProbCons Score Comparison
GP250 versus % Gappiness
ClustalW2.0 Score Comparison
GP250 versus ANHD
MAFFT Score Comparison
GP250 versus ANHD
Muscle3.7 Score Comparison
GP250 versus ANHD
ProbCons Score Comparison
GP250 versus ANHD
bal
pec
% Gappiness
ANHD
GP250 score

Figure B.1

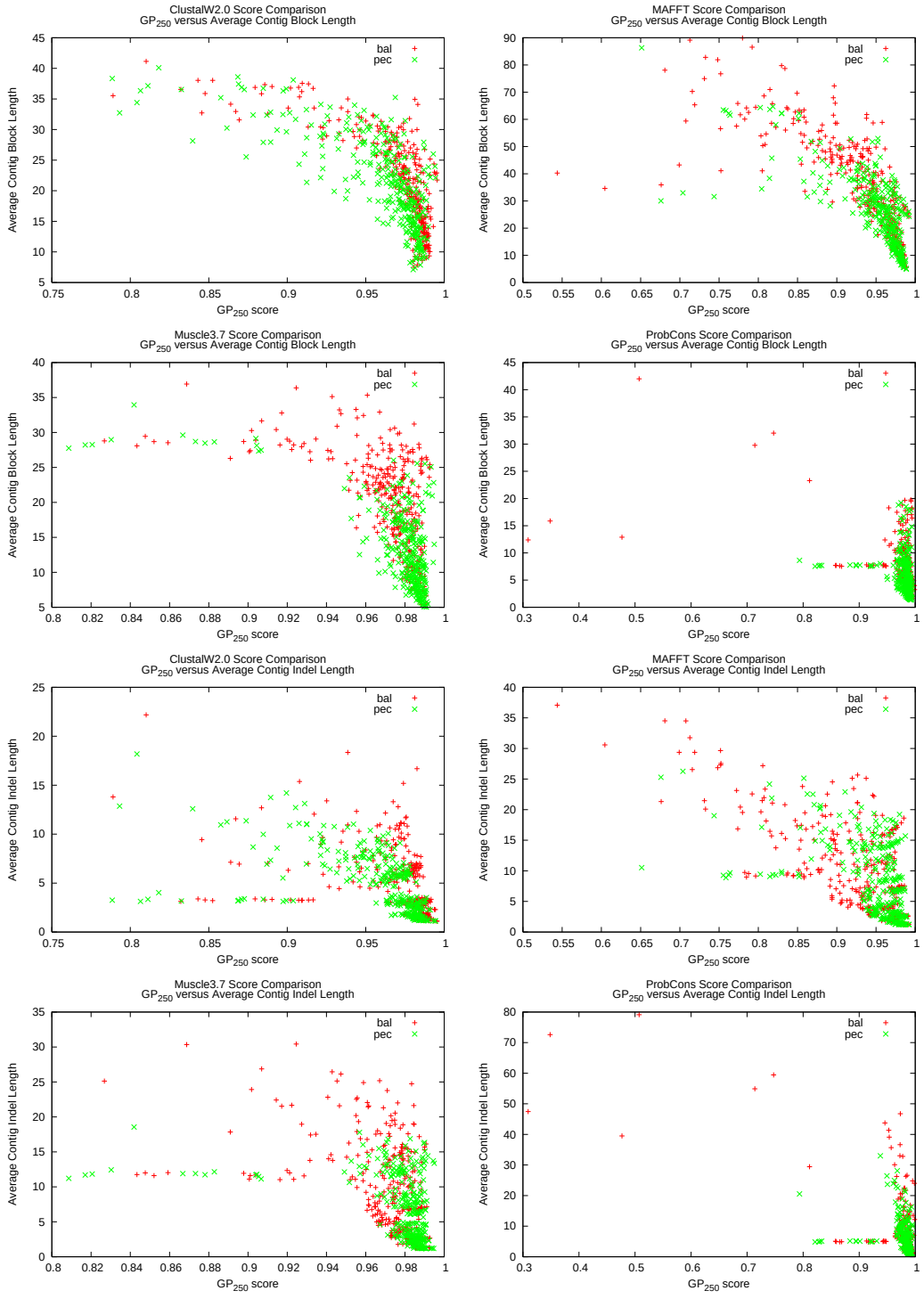

ClustalW2.0 Score Comparison
GP250 versus Average Contig Block Length
MAFFT Score Comparison
GP250 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Block Length
ProbCons Score Comparison
GP250 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP250 versus Average Contig Indel Length
MAFFT Score Comparison
GP250 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP250 versus Average Contig Indel Length
ProbCons Score Comparison
GP250 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP250 score

Figure B.1

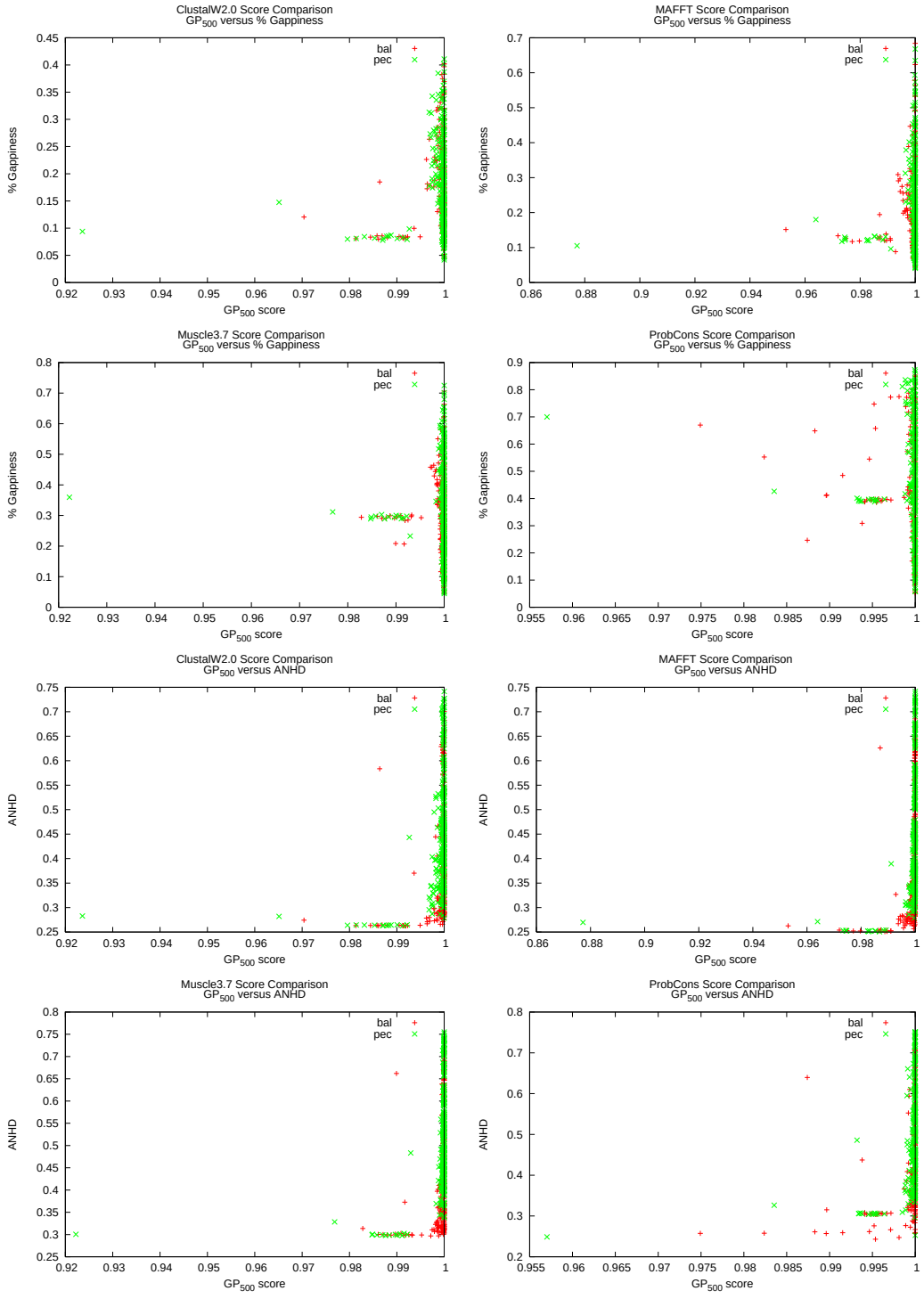

ClustalW2.0 Score Comparison
GP500 versus % Gappiness
MAFFT Score Comparison
GP500 versus % Gappiness
Muscle3.7 Score Comparison
GP500 versus % Gappiness
ProbCons Score Comparison
GP500 versus % Gappiness
ClustalW2.0 Score Comparison
GP500 versus ANHD
MAFFT Score Comparison
GP500 versus ANHD
Muscle3.7 Score Comparison
GP500 versus ANHD
ProbCons Score Comparison
GP500 versus ANHD
bal
pec
% Gappiness
ANHD
GP500 score

Figure B.1

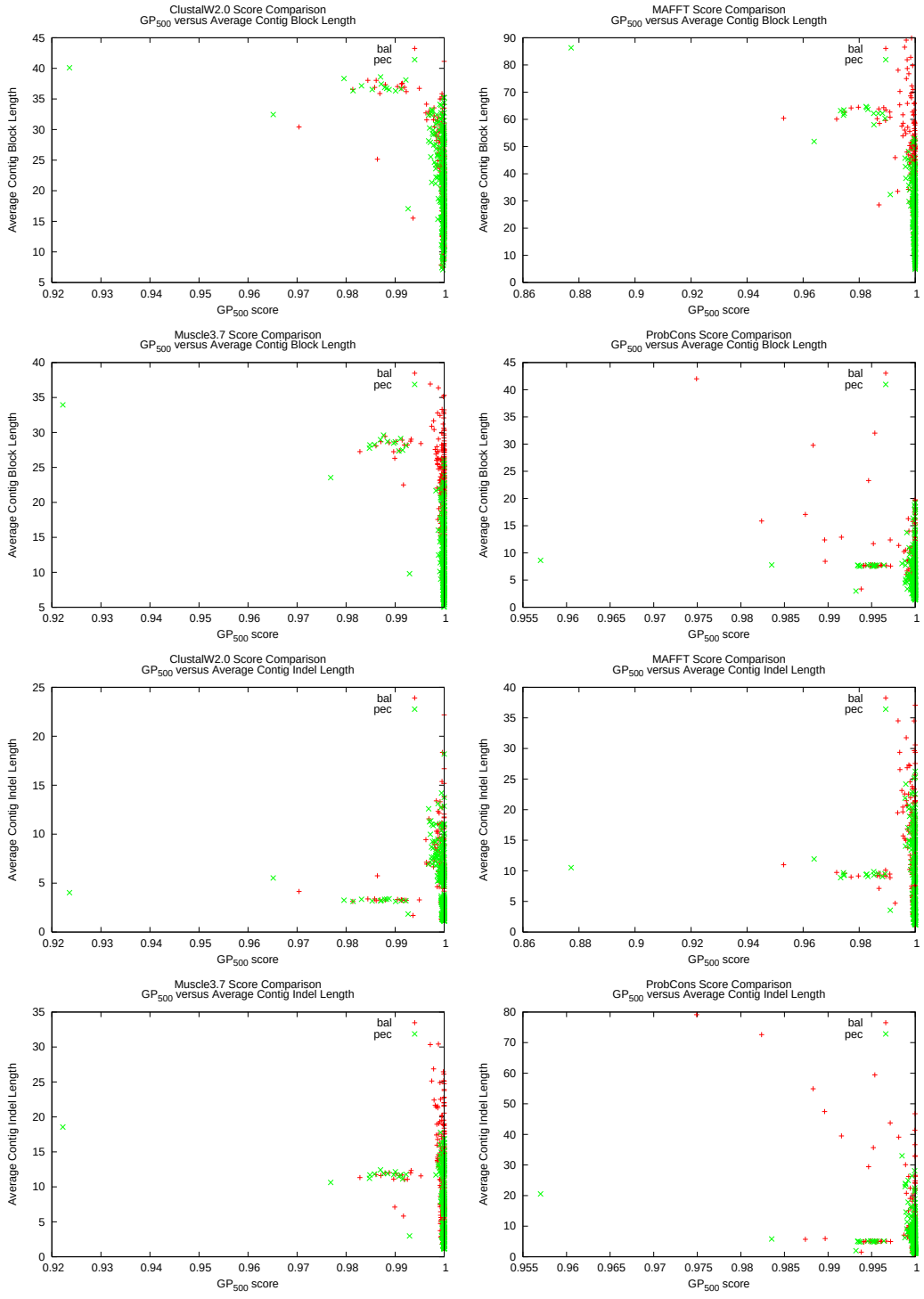

ClustalW2.0 Score Comparison
GP500 versus Average Contig Block Length
MAFFT Score Comparison
GP500 versus Average Contig Block Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Block Length
ProbCons Score Comparison
GP500 versus Average Contig Block Length
ClustalW2.0 Score Comparison
GP500 versus Average Contig Indel Length
MAFFT Score Comparison
GP500 versus Average Contig Indel Length
Muscle3.7 Score Comparison
GP500 versus Average Contig Indel Length
ProbCons Score Comparison
GP500 versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
GP500 score

Figure B.1

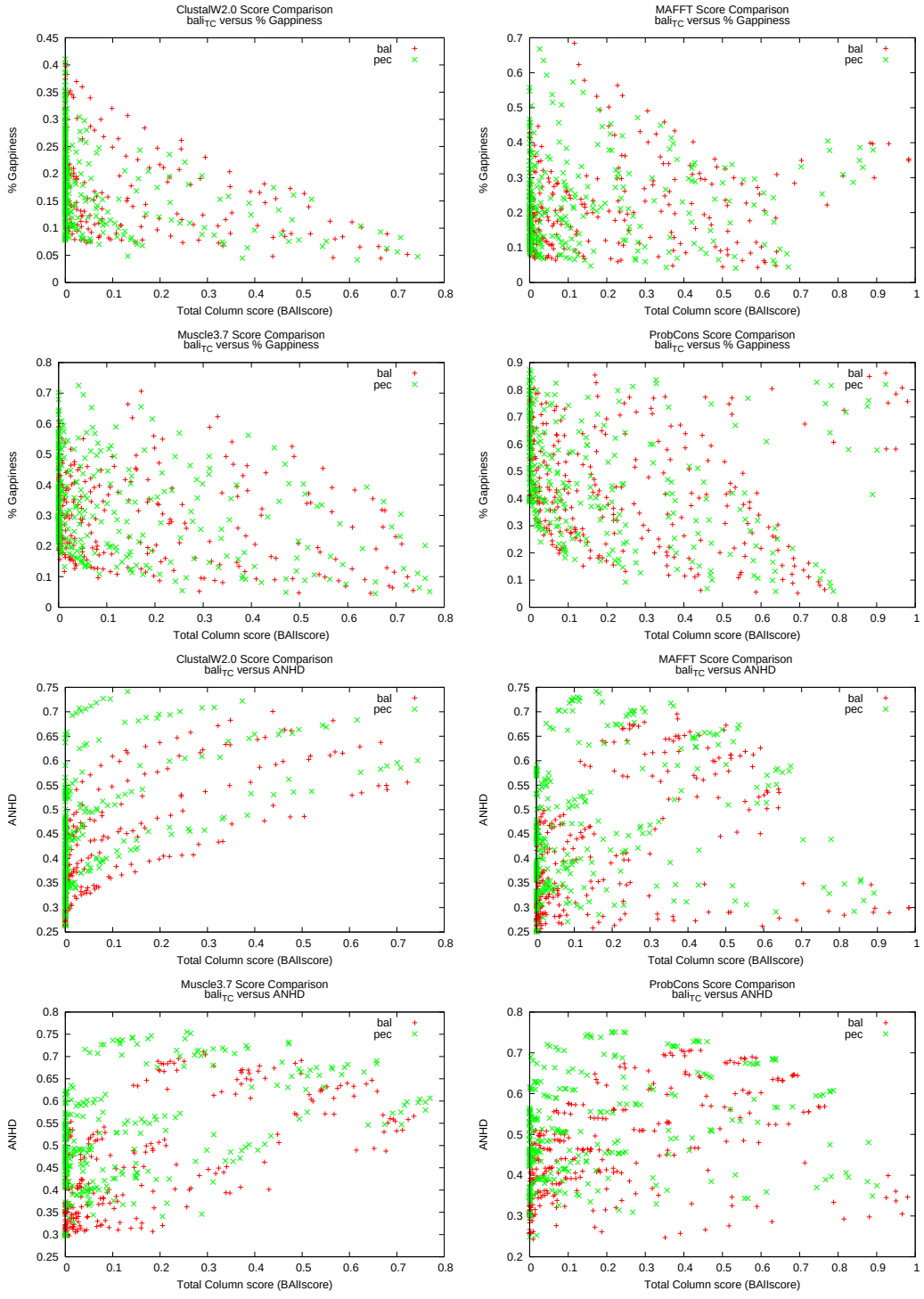

ClustalW2.0 Score Comparison
bali_TC versus % Gappiness
MAFFT Score Comparison
bali_TC versus % Gappiness
Muscle3.7 Score Comparison
bali_TC versus % Gappiness
ProbCons Score Comparison
bali_TC versus % Gappiness
ClustalW2.0 Score Comparison
bali_TC versus ANHD
MAFFT Score Comparison
bali_TC versus ANHD
Muscle3.7 Score Comparison
bali_TC versus ANHD
ProbCons Score Comparison
bali_TC versus ANHD
bal
pec
% Gappiness
ANHD
Total Column score (BAIIscore)

Figure B.1

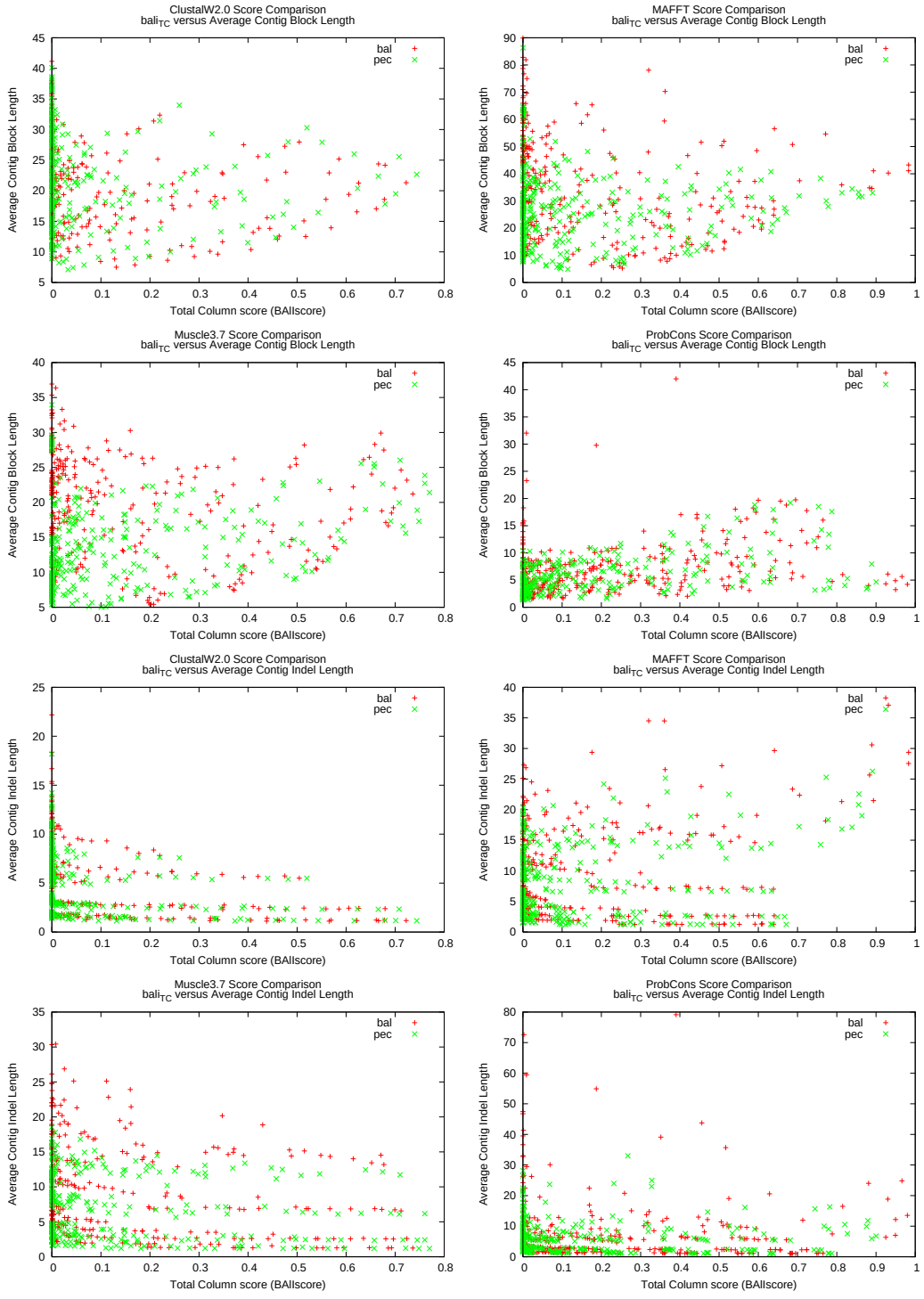

ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Block Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
MAFFT Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
ProbCons Score Comparison
bali$_{TC}$ versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
Total Column score (BAIIscore)

Figure B.1

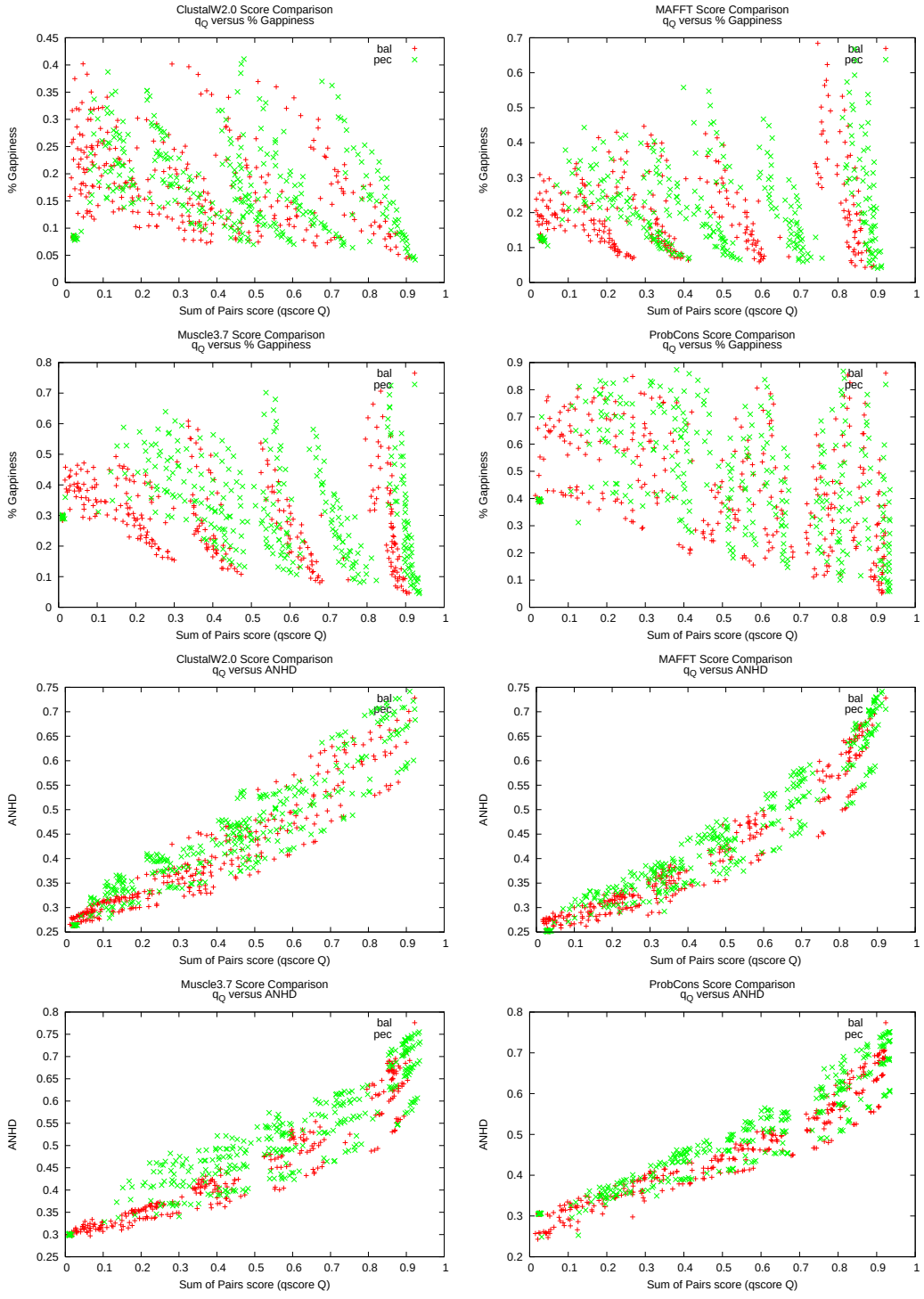

ClustalW2.0 Score Comparison
$q_Q$ versus % Gappiness
MAFFT Score Comparison
$q_Q$ versus % Gappiness
Muscle3.7 Score Comparison
$q_Q$ versus % Gappiness
ProbCons Score Comparison
$q_Q$ versus % Gappiness
ClustalW2.0 Score Comparison
$q_Q$ versus ANHD
MAFFT Score Comparison
$q_Q$ versus ANHD
Muscle3.7 Score Comparison
$q_Q$ versus ANHD
ProbCons Score Comparison
$q_Q$ versus ANHD
% Gappiness
ANHD
Sum of Pairs score (qscore Q)
bal
pec

Figure B.1

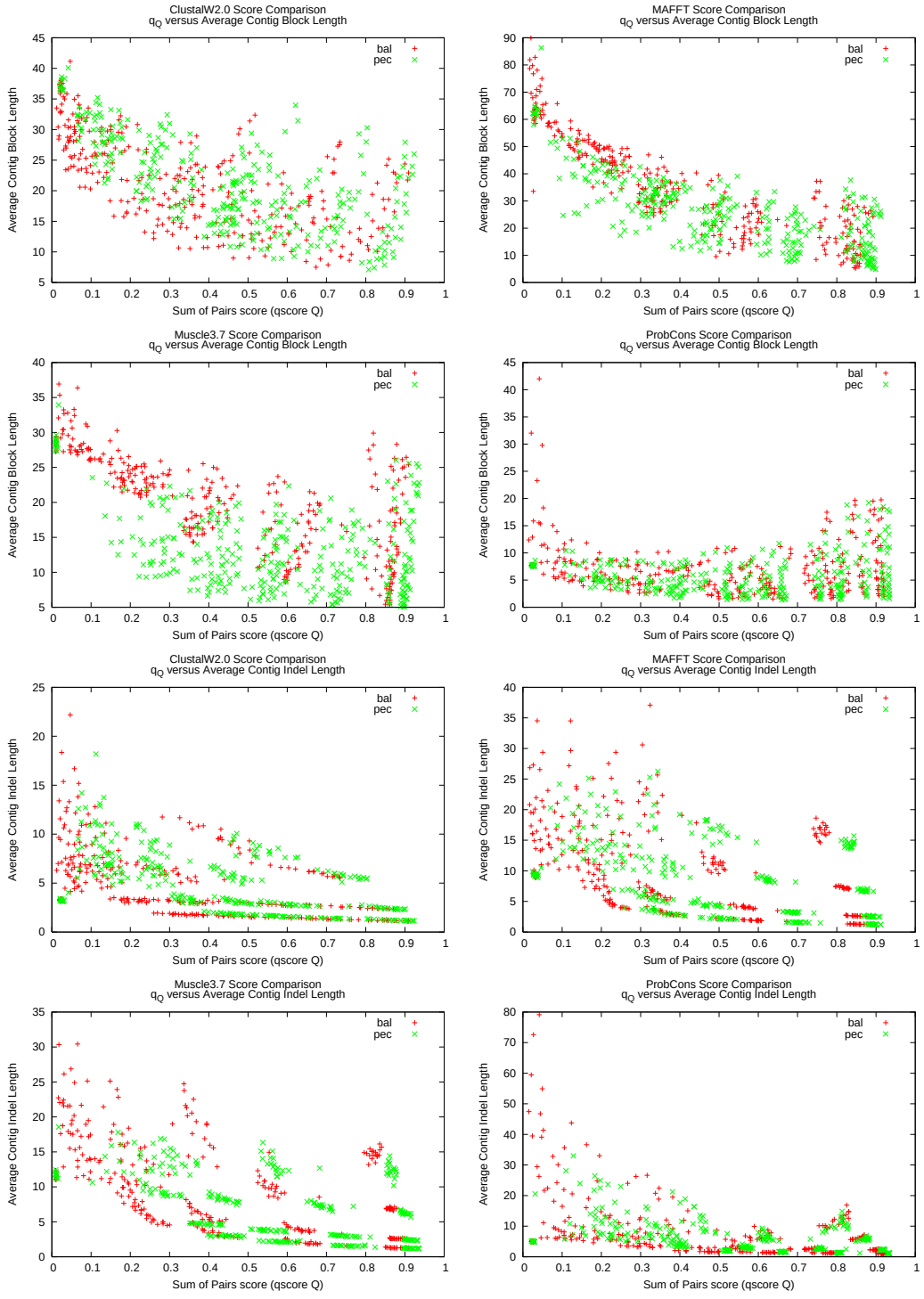

ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Block Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Block Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$q_Q$ versus Average Contig Indel Length
MAFFT Score Comparison
$q_Q$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$q_Q$ versus Average Contig Indel Length
ProbCons Score Comparison
$q_Q$ versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
Sum of Pairs score (qscore Q)

Figure B.1

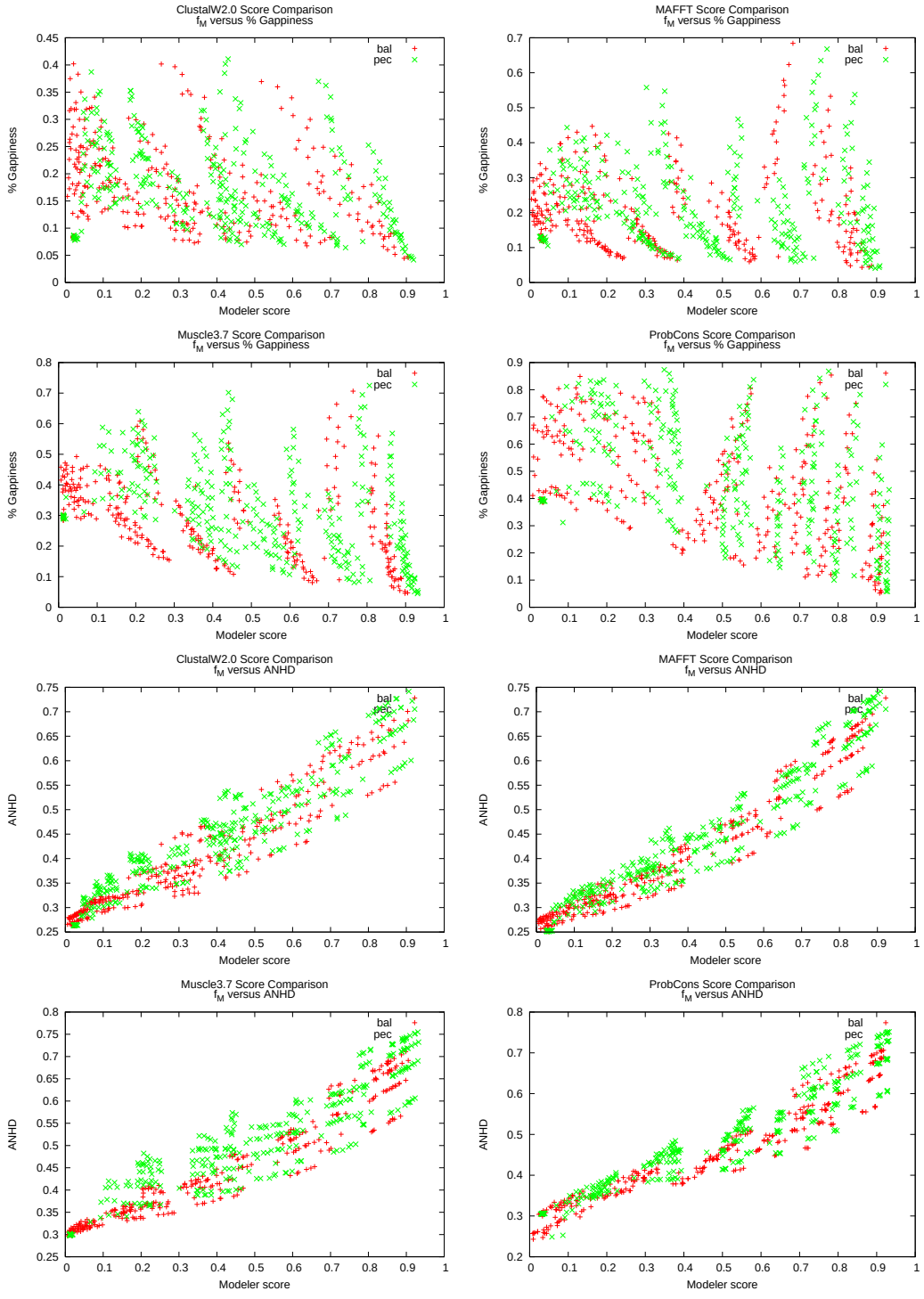

ClustalW2.0 Score Comparison
$f_M$ versus % Gappiness
MAFFT Score Comparison
$f_M$ versus % Gappiness
Muscle3.7 Score Comparison
$f_M$ versus % Gappiness
ProbCons Score Comparison
$f_M$ versus % Gappiness
ClustalW2.0 Score Comparison
$f_M$ versus ANHD
MAFFT Score Comparison
$f_M$ versus ANHD
Muscle3.7 Score Comparison
$f_M$ versus ANHD
ProbCons Score Comparison
$f_M$ versus ANHD
% Gappiness
ANHD
Modeler score
bal
pec

Figure B.1

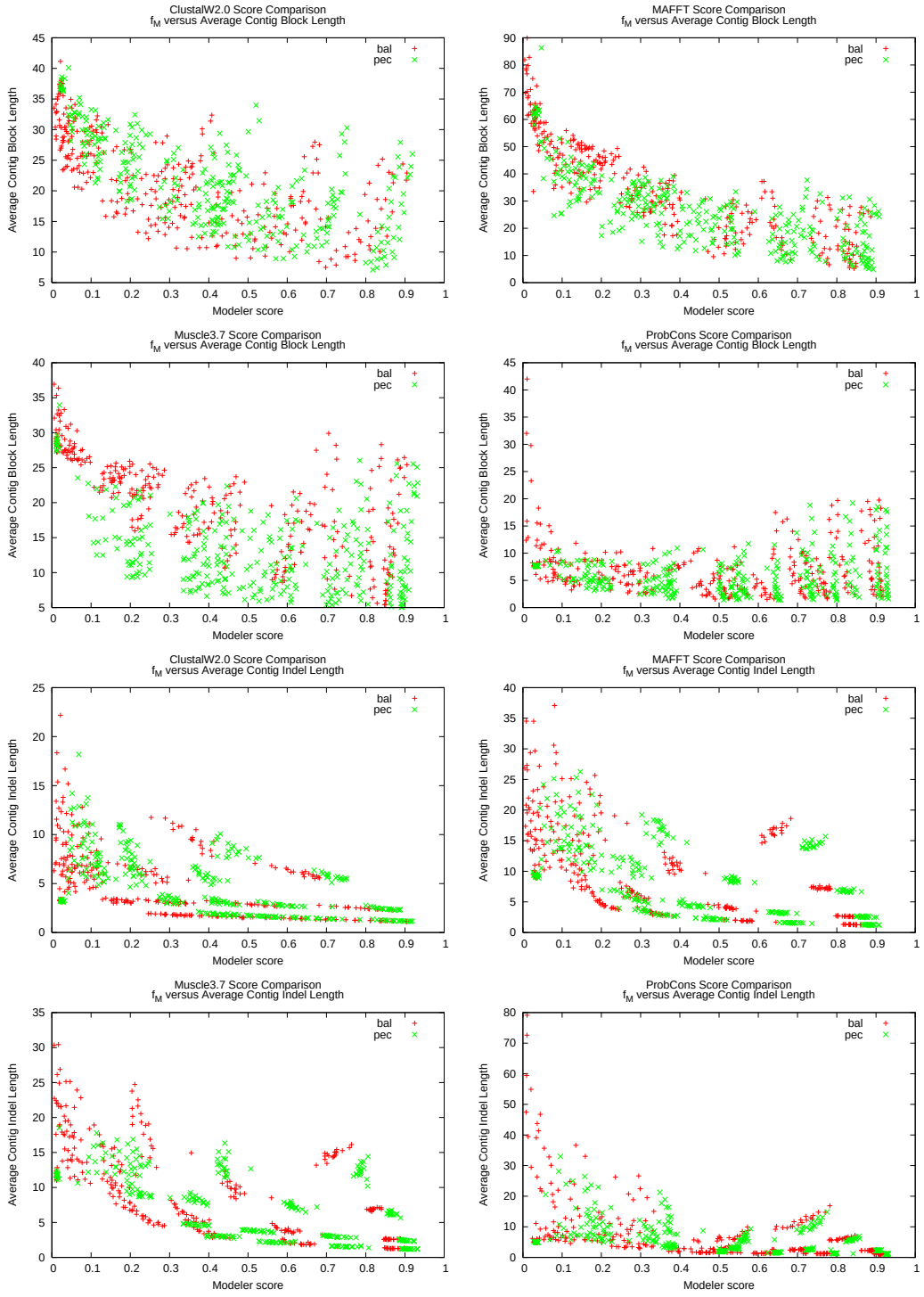

ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Block Length
MAFFT Score Comparison
$f_M$ versus Average Contig Block Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Block Length
ProbCons Score Comparison
$f_M$ versus Average Contig Block Length
ClustalW2.0 Score Comparison
$f_M$ versus Average Contig Indel Length
MAFFT Score Comparison
$f_M$ versus Average Contig Indel Length
Muscle3.7 Score Comparison
$f_M$ versus Average Contig Indel Length
ProbCons Score Comparison
$f_M$ versus Average Contig Indel Length
bal
pec
Average Contig Block Length
Average Contig Indel Length
Modeler score

Figure B.1

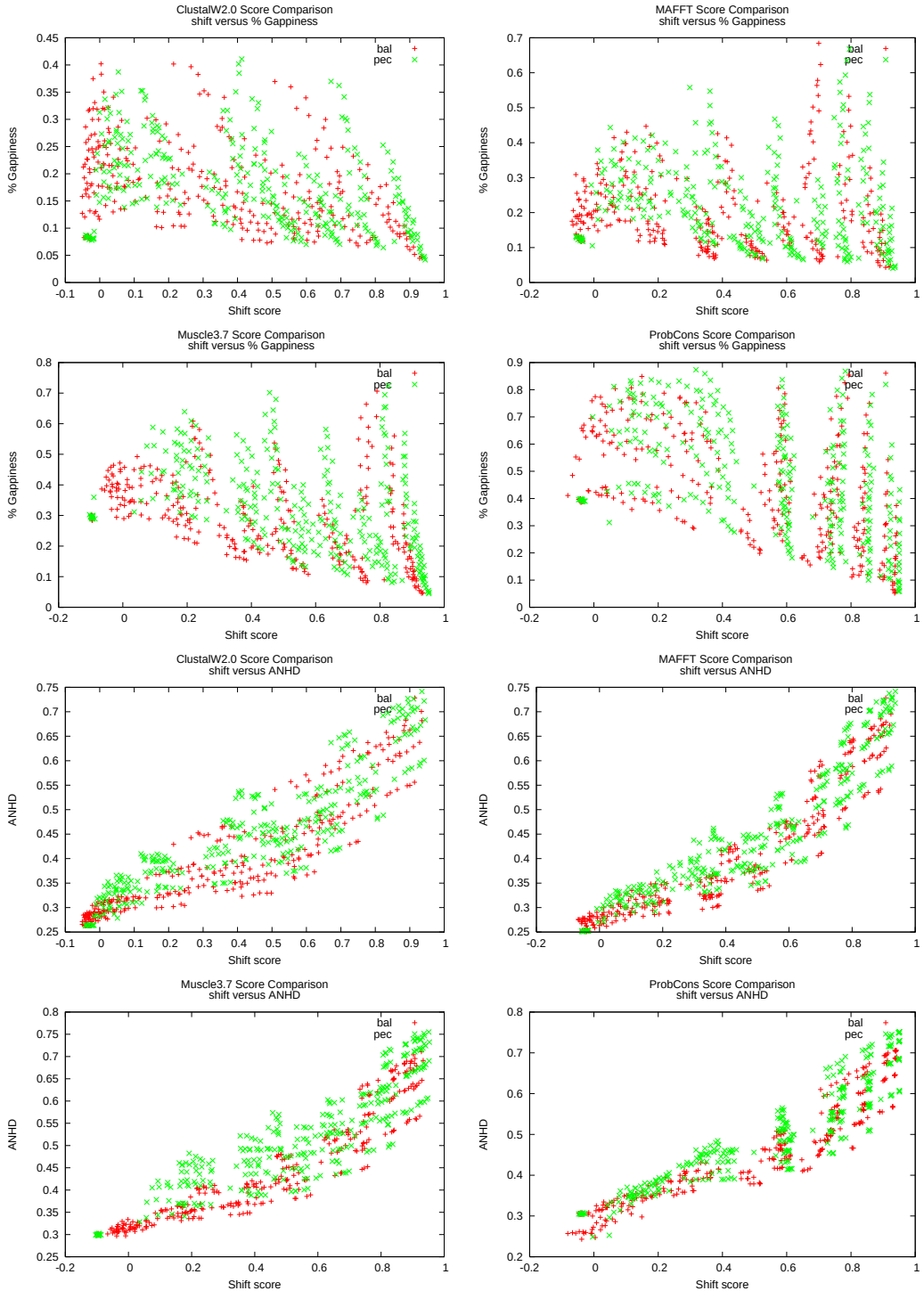

ClustalW2.0 Score Comparison
shift versus % Gappiness
MAFFT Score Comparison
shift versus % Gappiness
Muscle3.7 Score Comparison
shift versus % Gappiness
ProbCons Score Comparison
shift versus % Gappiness
ClustalW2.0 Score Comparison
shift versus ANHD
MAFFT Score Comparison
shift versus ANHD
Muscle3.7 Score Comparison
shift versus ANHD
ProbCons Score Comparison
shift versus ANHD
bal
pec
% Gappiness
ANHD
Shift score

Figure B.1

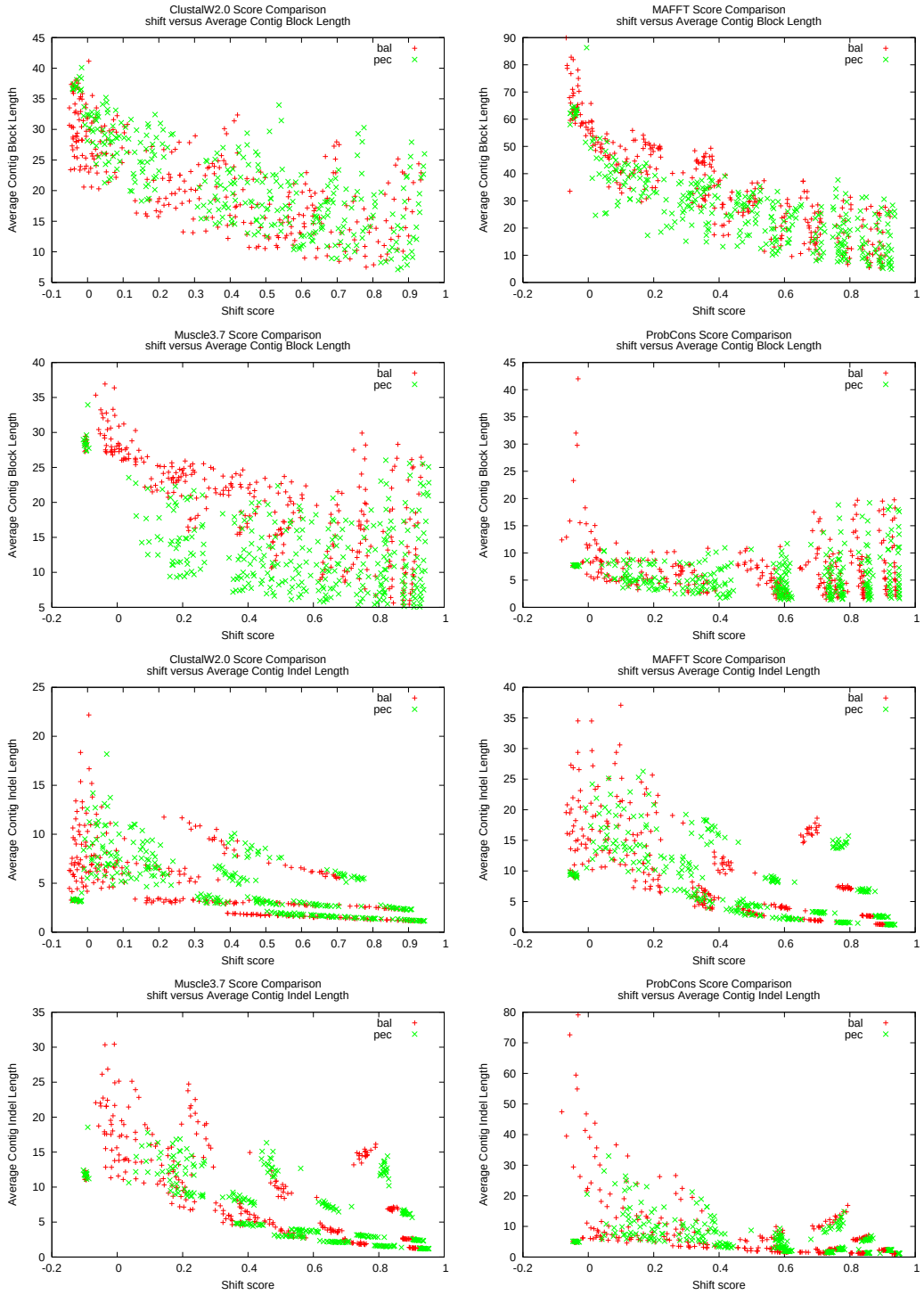



Figure B.1: Scoring metric scores versus benchmark statistic. Color of data points correspond to the benchmark parameter.

## B.2 Scoring Methods Side-by-side Comparison

This section is a comprehensive side-by-side comparison of MSA scoring metrics. For each of the MSA methods, the axes of the plot correspond to the two scoring metrics being compared, with the data points colored to correspond with the benchmark parameter used to create each data point.

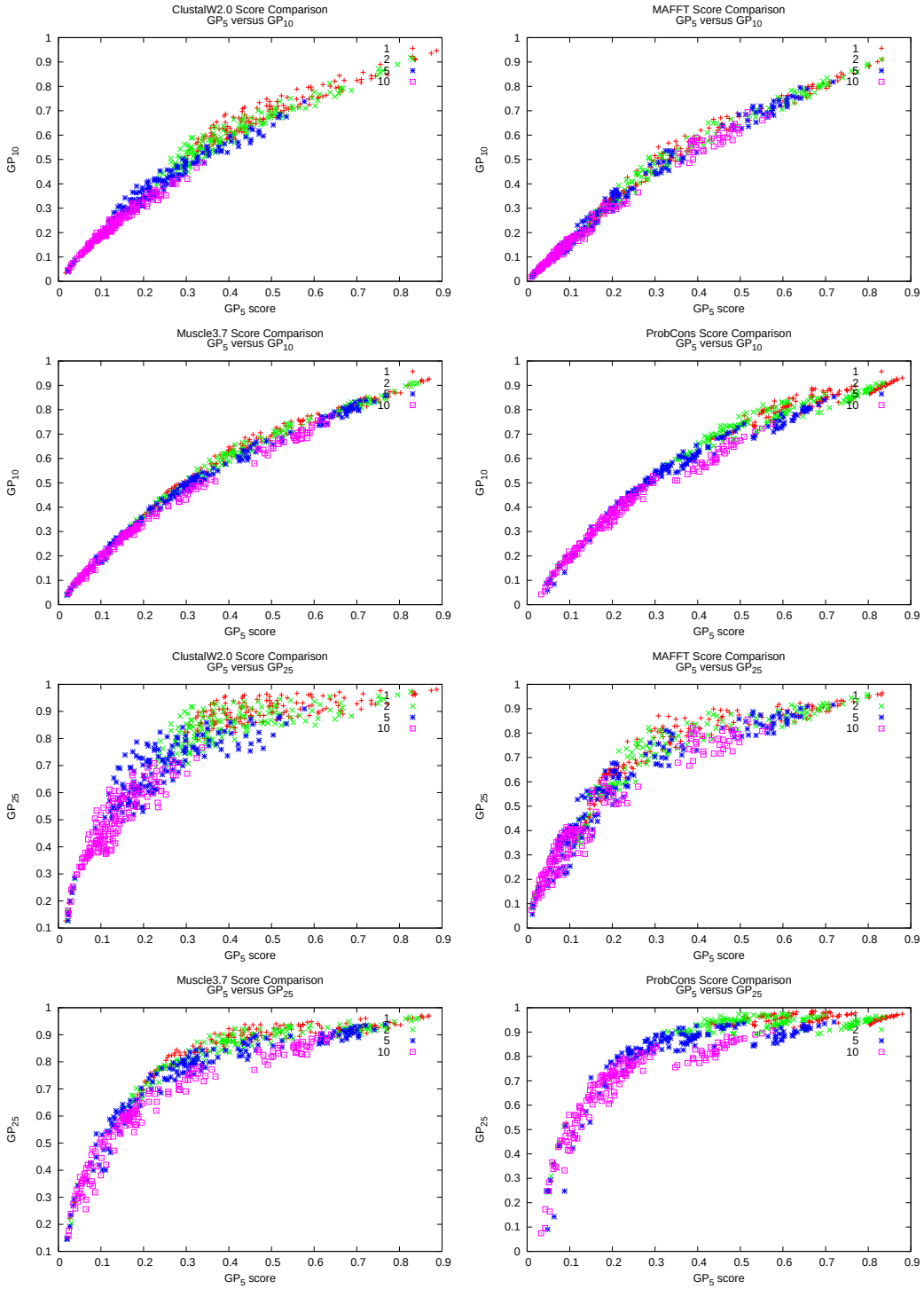

ClustalW2.0 Score Comparison
GP5 versus GP10
MAFFT Score Comparison
GP5 versus GP10
Muscle3.7 Score Comparison
GP5 versus GP10
ProbCons Score Comparison
GP5 versus GP10
ClustalW2.0 Score Comparison
GP5 versus GP25
MAFFT Score Comparison
GP5 versus GP25
Muscle3.7 Score Comparison
GP5 versus GP25
ProbCons Score Comparison
GP5 versus GP25
GP10
GP25
GP5 score
1
2
5
10

Figure B.2

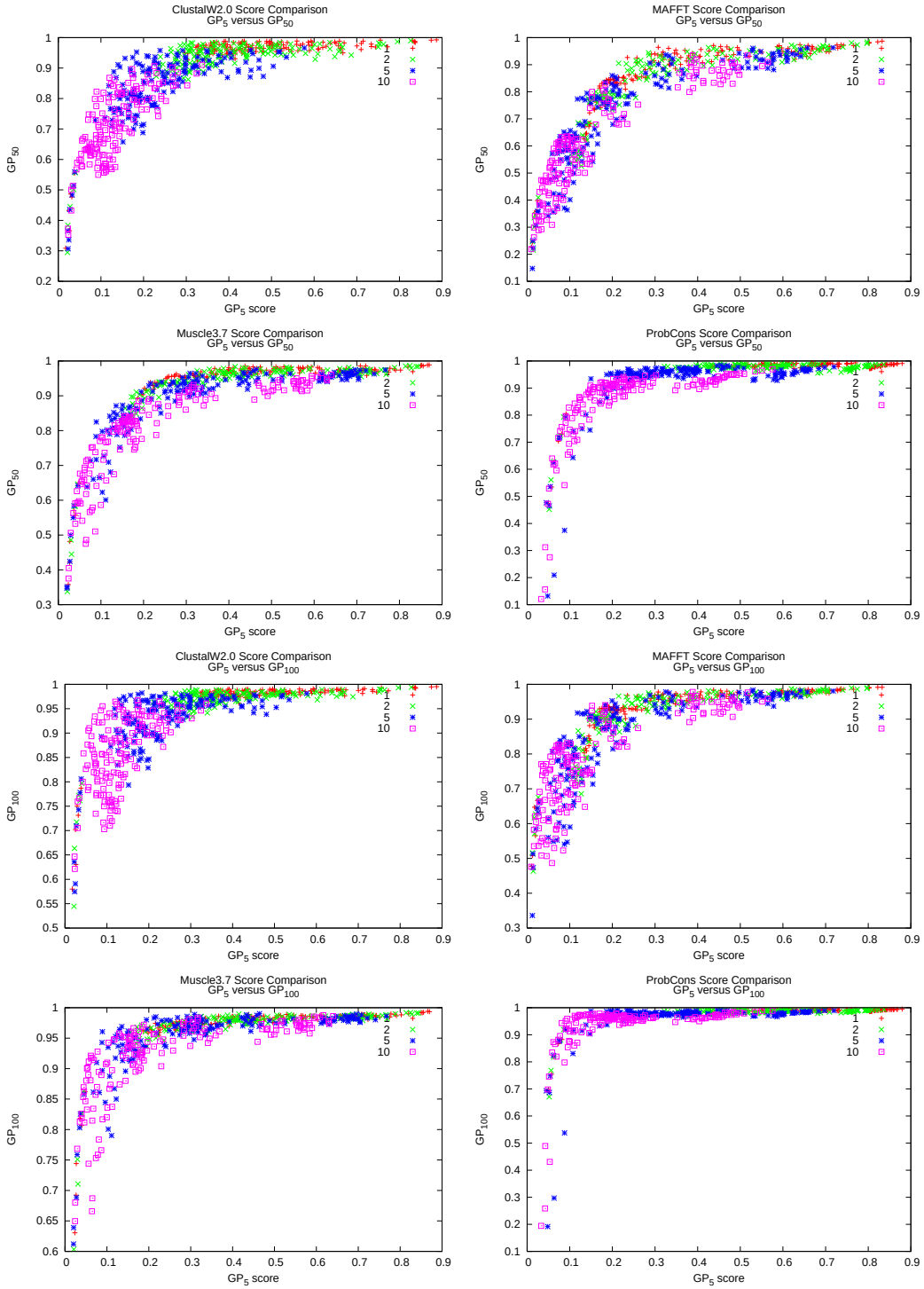

ClustalW2.0 Score Comparison
GP5 versus GP50
MAFFT Score Comparison
GP5 versus GP50
Muscle3.7 Score Comparison
GP5 versus GP50
ProbCons Score Comparison
GP5 versus GP50
ClustalW2.0 Score Comparison
GP5 versus GP100
MAFFT Score Comparison
GP5 versus GP100
Muscle3.7 Score Comparison
GP5 versus GP100
ProbCons Score Comparison
GP5 versus GP100
GP50
GP100
GP5 score
1
2
5
10

Figure B.2

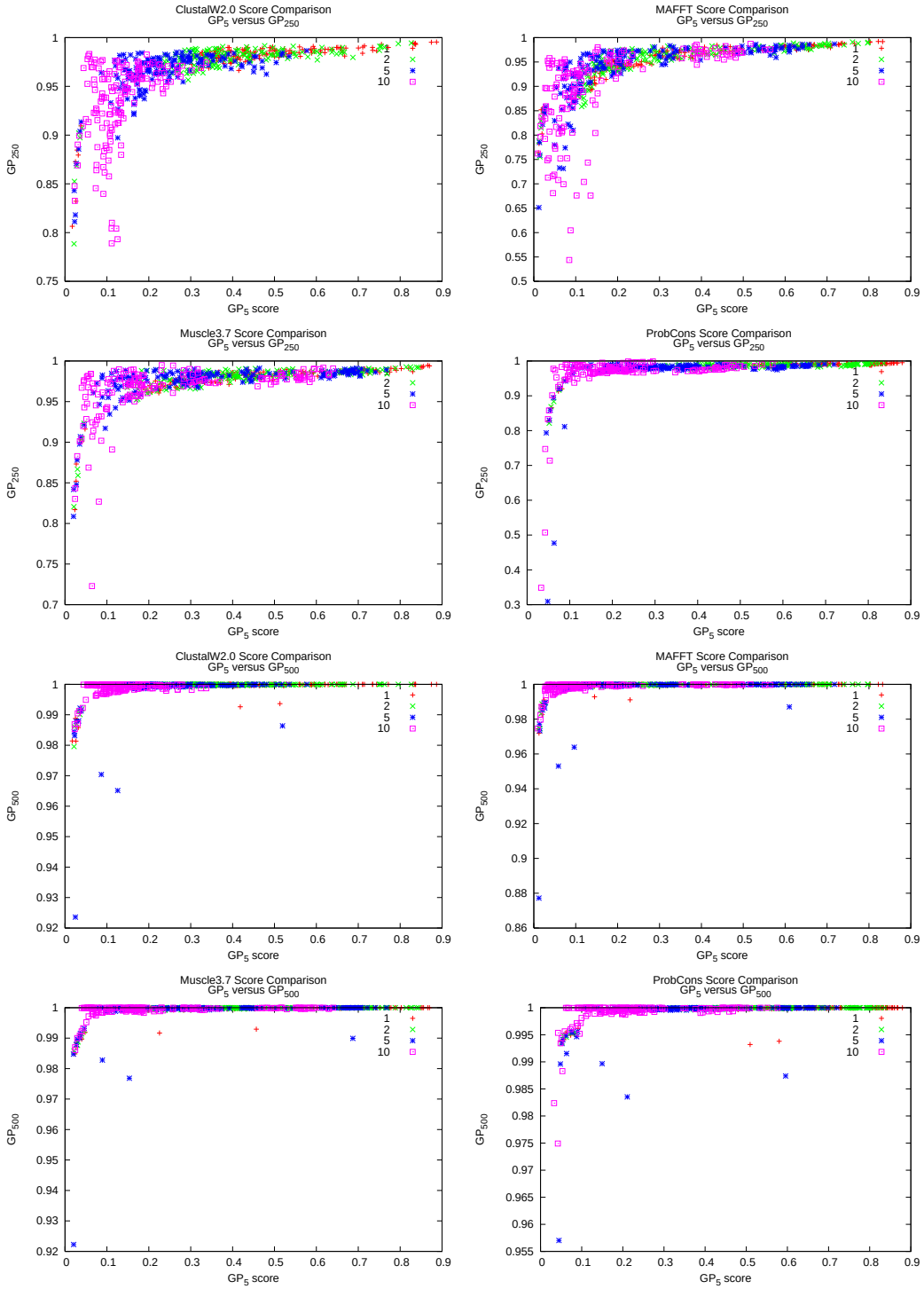

ClustalW2.0 Score Comparison
GP5 versus GP250
MAFFT Score Comparison
GP5 versus GP250
Muscle3.7 Score Comparison
GP5 versus GP250
ProbCons Score Comparison
GP5 versus GP250
ClustalW2.0 Score Comparison
GP5 versus GP500
MAFFT Score Comparison
GP5 versus GP500
Muscle3.7 Score Comparison
GP5 versus GP500
ProbCons Score Comparison
GP5 versus GP500
GP250
GP500
GP5 score
1
2
5
10

Figure B.2

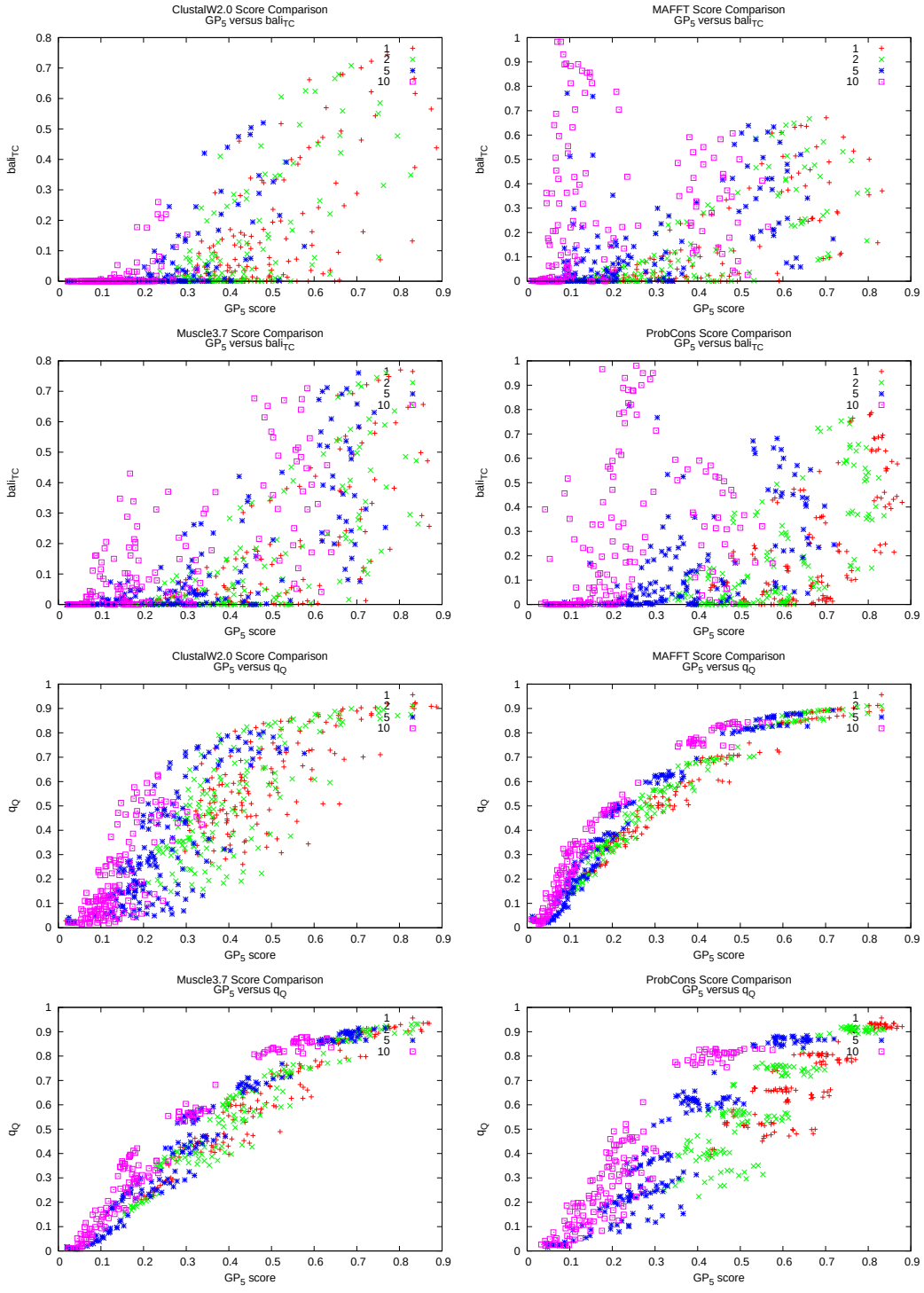

ClustalW2.0 Score Comparison
GP5 versus balitc
MAFFT Score Comparison
GP5 versus balitc
Muscle3.7 Score Comparison
GP5 versus balitc
ProbCons Score Comparison
GP5 versus balitc
ClustalW2.0 Score Comparison
GP5 versus qQ
MAFFT Score Comparison
GP5 versus qQ
Muscle3.7 Score Comparison
GP5 versus qQ
ProbCons Score Comparison
GP5 versus qQ
balitc
qQ
GP5 score
1
2
5
10

Figure B.2

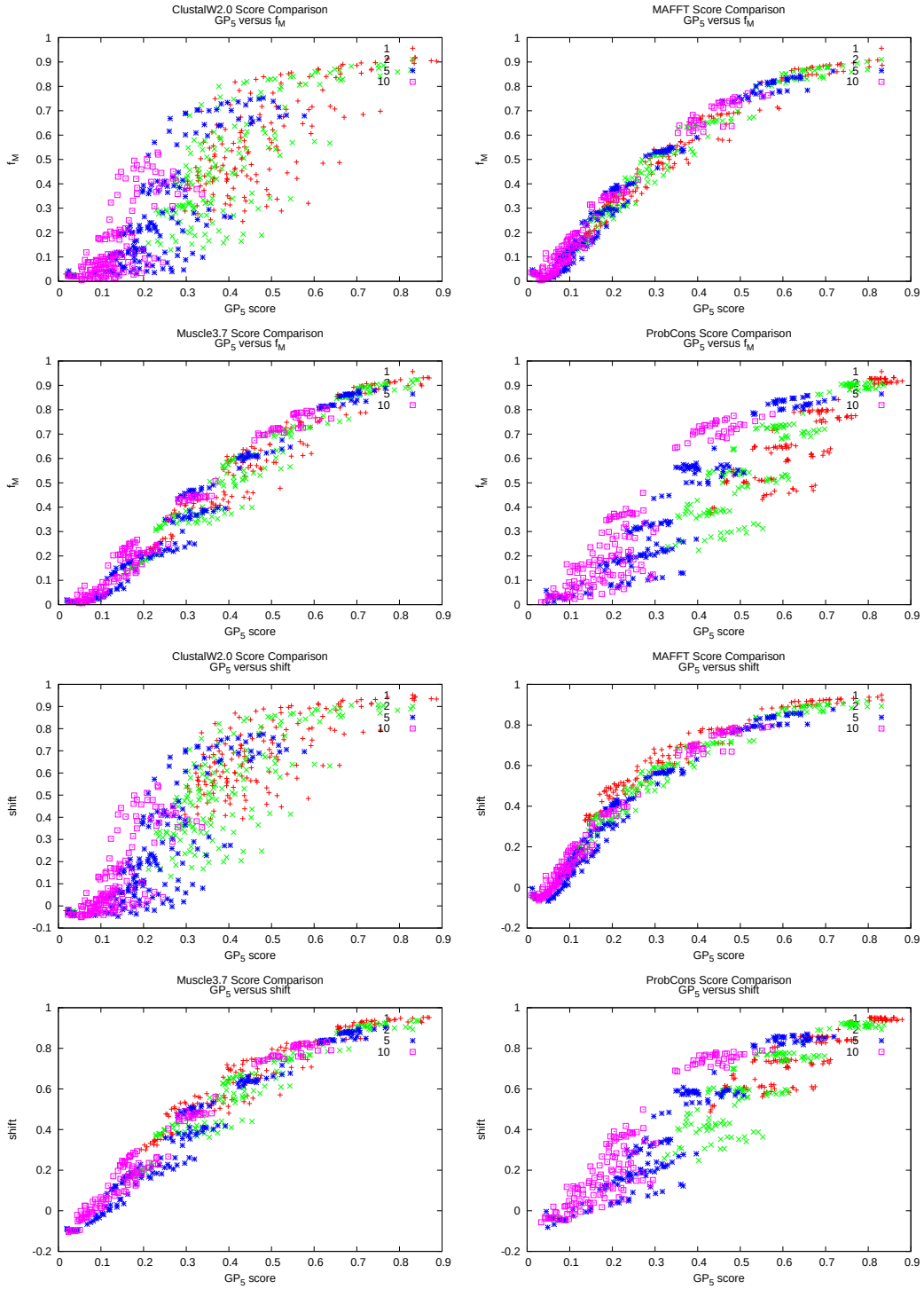

ClustalW2.0 Score Comparison
GP5 versus fM
MAFFT Score Comparison
GP5 versus fM
Muscle3.7 Score Comparison
GP5 versus fM
ProbCons Score Comparison
GP5 versus fM
ClustalW2.0 Score Comparison
GP5 versus shift
MAFFT Score Comparison
GP5 versus shift
Muscle3.7 Score Comparison
GP5 versus shift
ProbCons Score Comparison
GP5 versus shift
fM
shift
GP5 score
1
2
5
10

Figure B.2

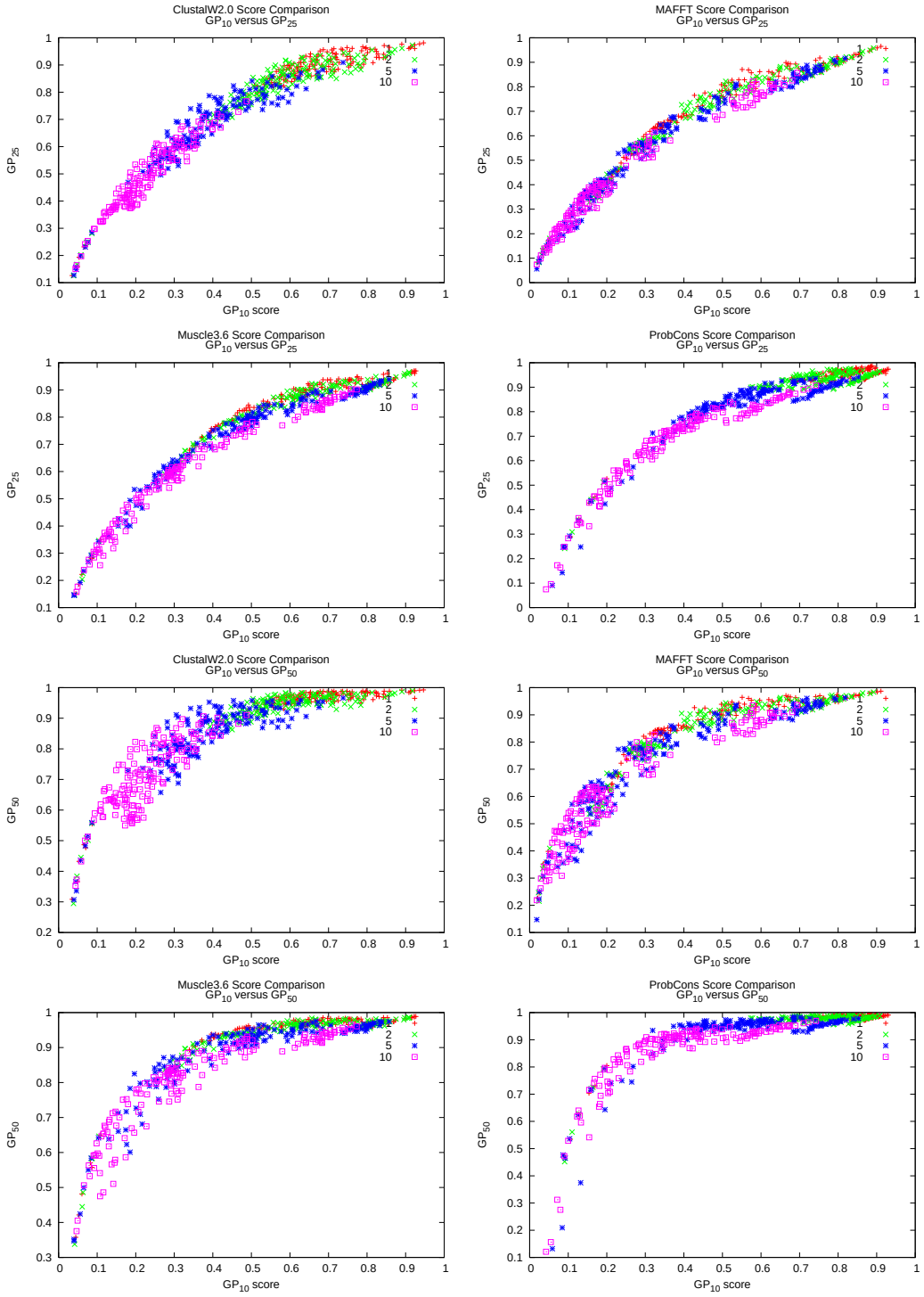

ClustalW2.0 Score Comparison
GP10 versus GP25
MAFFT Score Comparison
GP10 versus GP25
Muscle3.6 Score Comparison
GP10 versus GP25
ProbCons Score Comparison
GP10 versus GP25
ClustalW2.0 Score Comparison
GP10 versus GP50
MAFFT Score Comparison
GP10 versus GP50
Muscle3.6 Score Comparison
GP10 versus GP50
ProbCons Score Comparison
GP10 versus GP50
GP25
GP50
GP10 score
1
2
5
10

Figure B.2

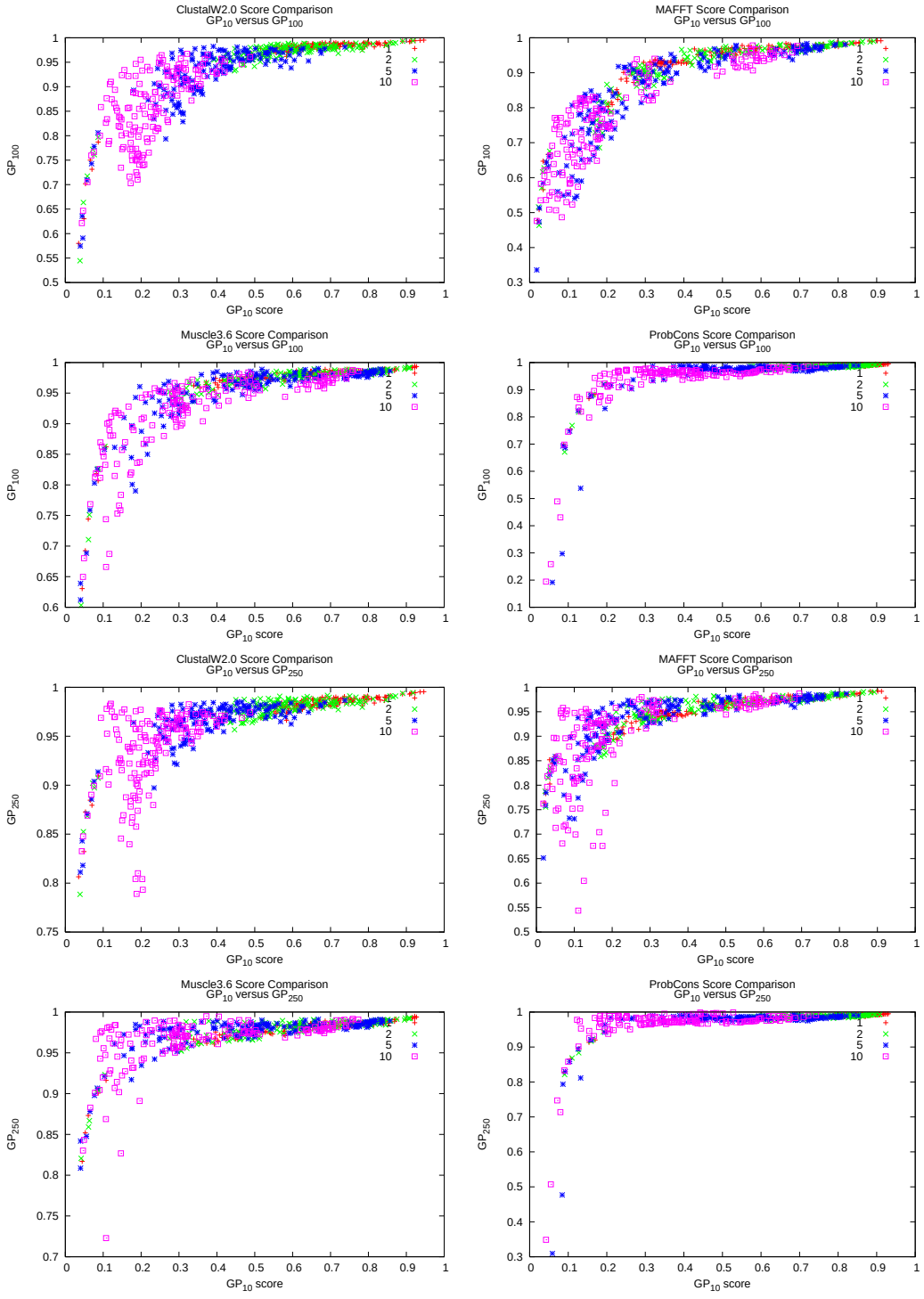

ClustalW2.0 Score Comparison
GP10 versus GP100
MAFFT Score Comparison
GP10 versus GP100
Muscle3.6 Score Comparison
GP10 versus GP100
ProbCons Score Comparison
GP10 versus GP100
ClustalW2.0 Score Comparison
GP10 versus GP250
MAFFT Score Comparison
GP10 versus GP250
Muscle3.6 Score Comparison
GP10 versus GP250
ProbCons Score Comparison
GP10 versus GP250
GP100
GP250
GP10 score
1
2
5
10

Figure B.2

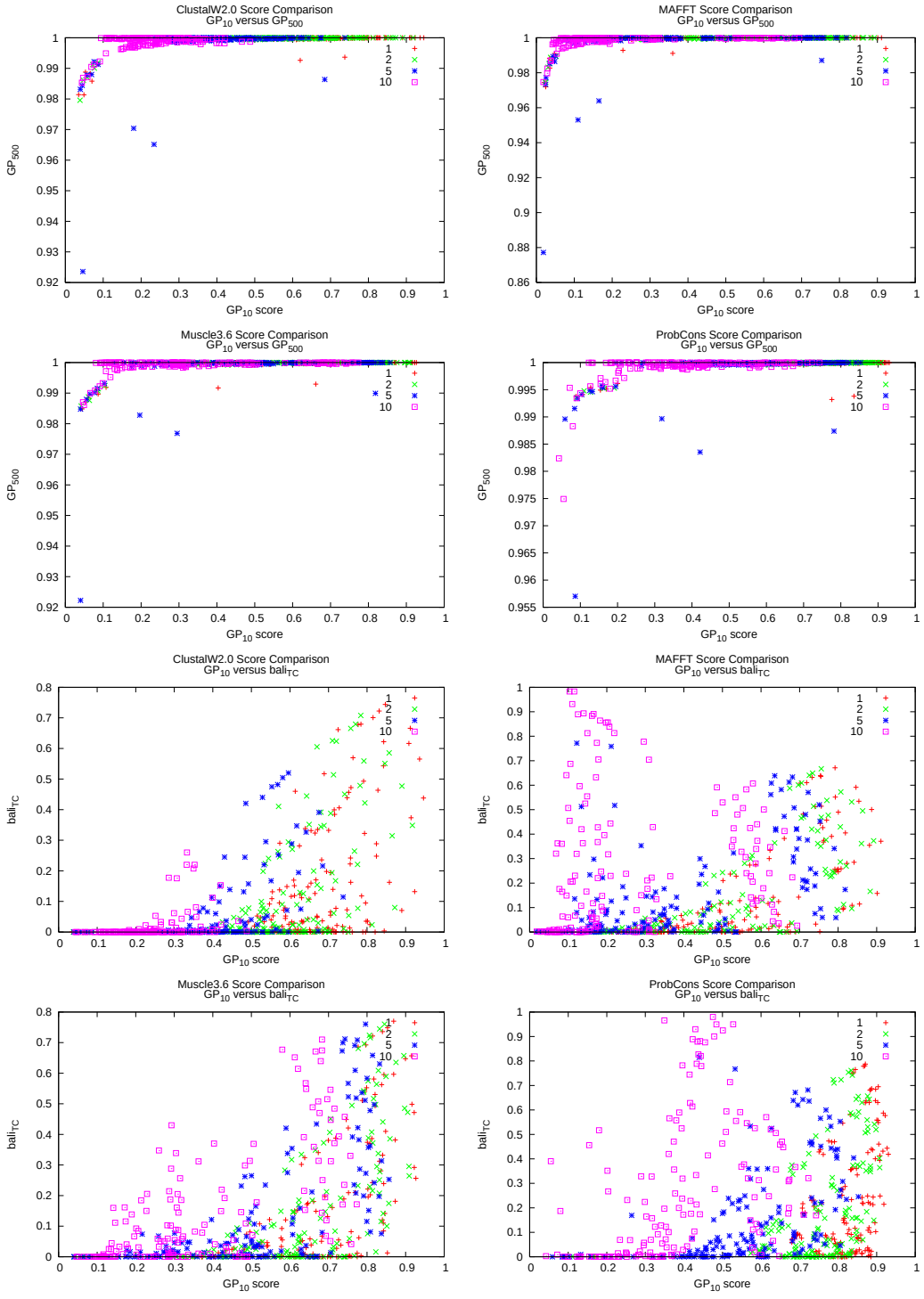

ClustalW2.0 Score Comparison
GP10 versus GP500
MAFFT Score Comparison
GP10 versus GP500
Muscle3.6 Score Comparison
GP10 versus GP500
ProbCons Score Comparison
GP10 versus GP500
ClustalW2.0 Score Comparison
GP10 versus balitc
MAFFT Score Comparison
GP10 versus balitc
Muscle3.6 Score Comparison
GP10 versus balitc
ProbCons Score Comparison
GP10 versus balitc
GP500
balitc
GP10 score
1
2
5
10

Figure B.2

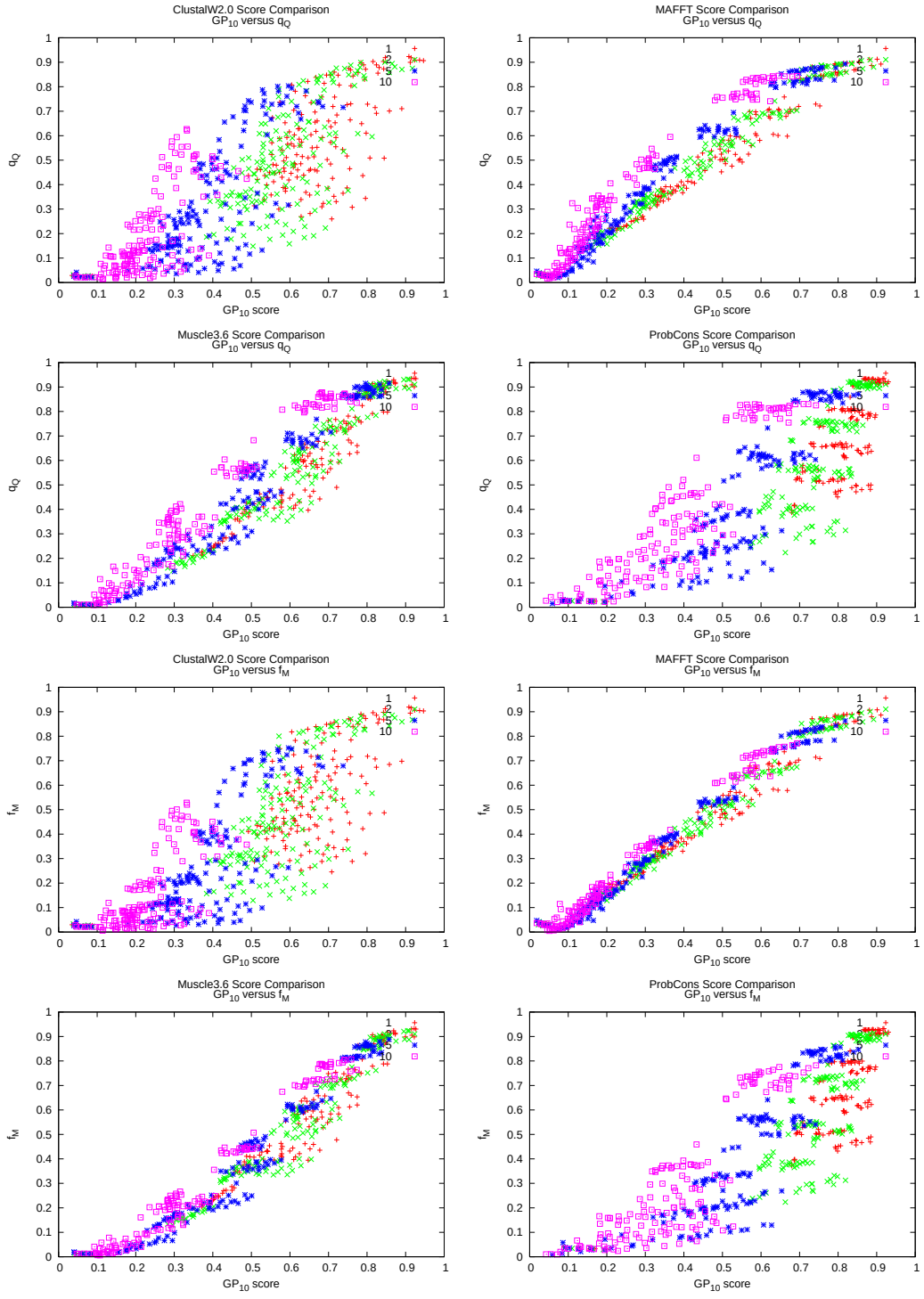

ClustalW2.0 Score Comparison
GP10 versus qQ
MAFFT Score Comparison
GP10 versus qQ
Muscle3.6 Score Comparison
GP10 versus qQ
ProbCons Score Comparison
GP10 versus qQ
ClustalW2.0 Score Comparison
GP10 versus fM
MAFFT Score Comparison
GP10 versus fM
Muscle3.6 Score Comparison
GP10 versus fM
ProbCons Score Comparison
GP10 versus fM
qQ
fM
GP10 score
1
2
5
10

Figure B.2

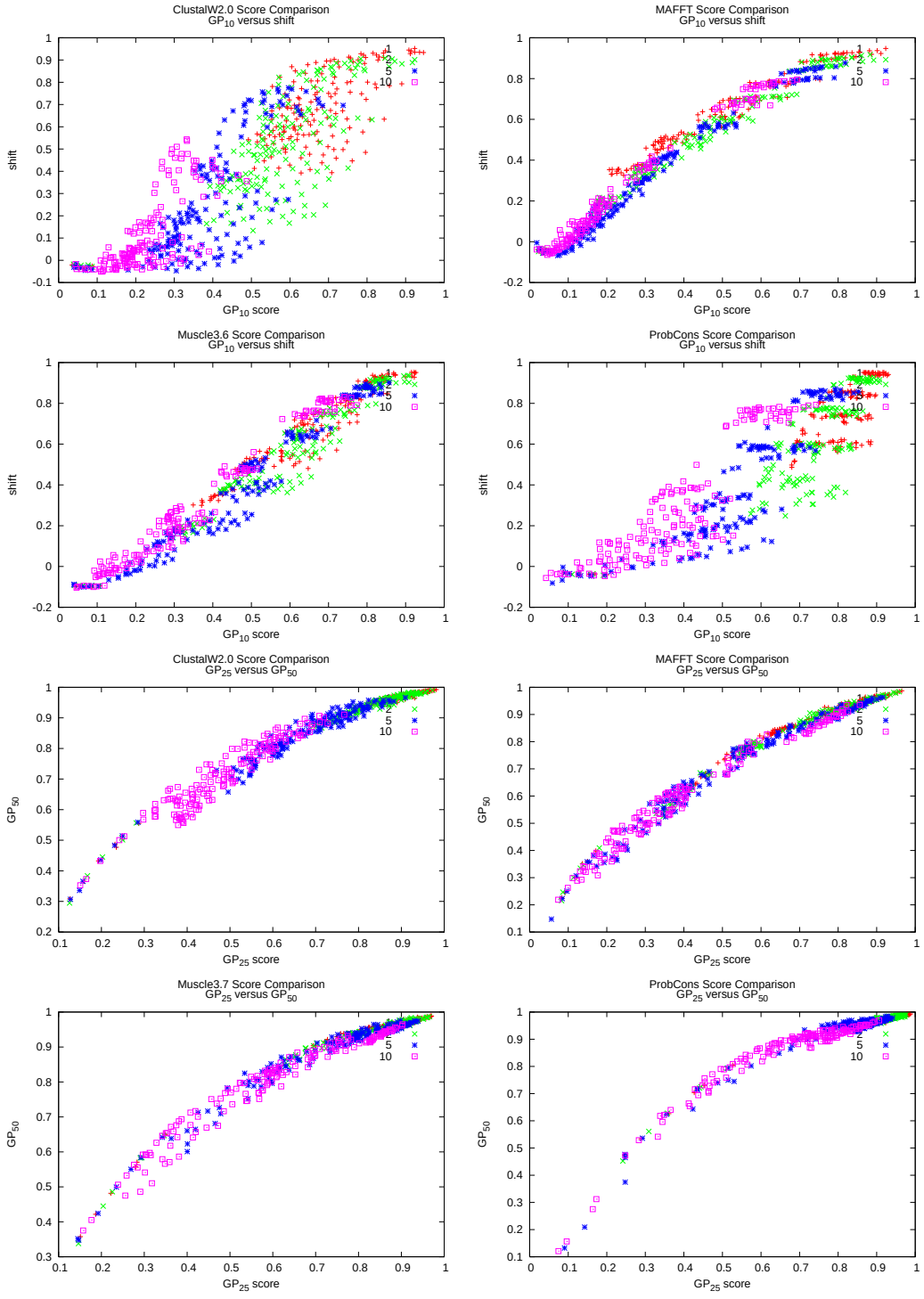

ClustalW2.0 Score Comparison
GP10 versus shift
MAFFT Score Comparison
GP10 versus shift
Muscle3.6 Score Comparison
GP10 versus shift
ProbCons Score Comparison
GP10 versus shift
ClustalW2.0 Score Comparison
GP25 versus GP50
MAFFT Score Comparison
GP25 versus GP50
Muscle3.7 Score Comparison
GP25 versus GP50
ProbCons Score Comparison
GP25 versus GP50
shift
GP50
GP10 score
GP25 score
1
2
5
10

Figure B.2

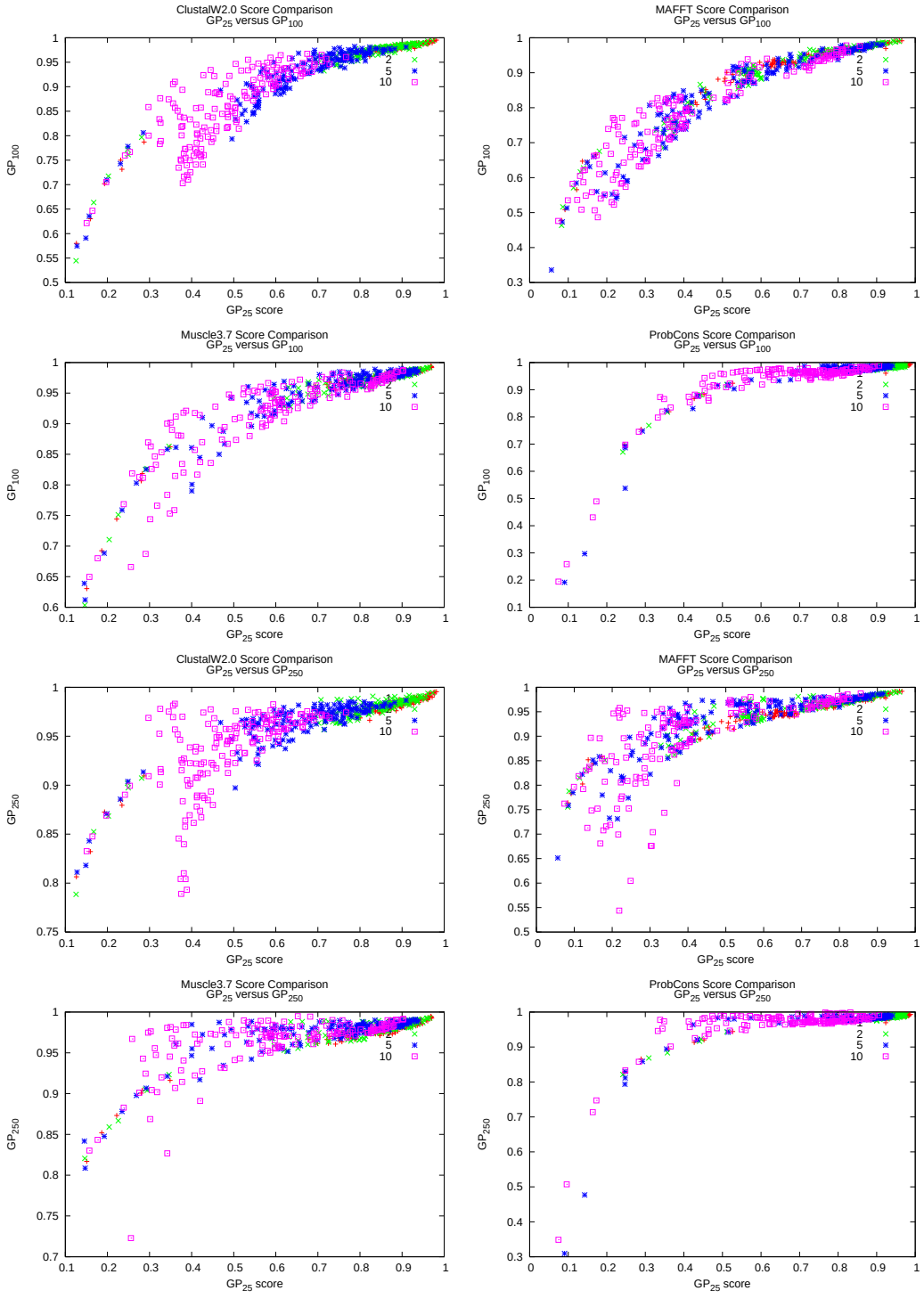

ClustalW2.0 Score Comparison
GP25 versus GP100
MAFFT Score Comparison
GP25 versus GP100
Muscle3.7 Score Comparison
GP25 versus GP100
ProbCons Score Comparison
GP25 versus GP100
ClustalW2.0 Score Comparison
GP25 versus GP250
MAFFT Score Comparison
GP25 versus GP250
Muscle3.7 Score Comparison
GP25 versus GP250
ProbCons Score Comparison
GP25 versus GP250
GP100
GP250
GP25 score
1
2
5
10

Figure B.2

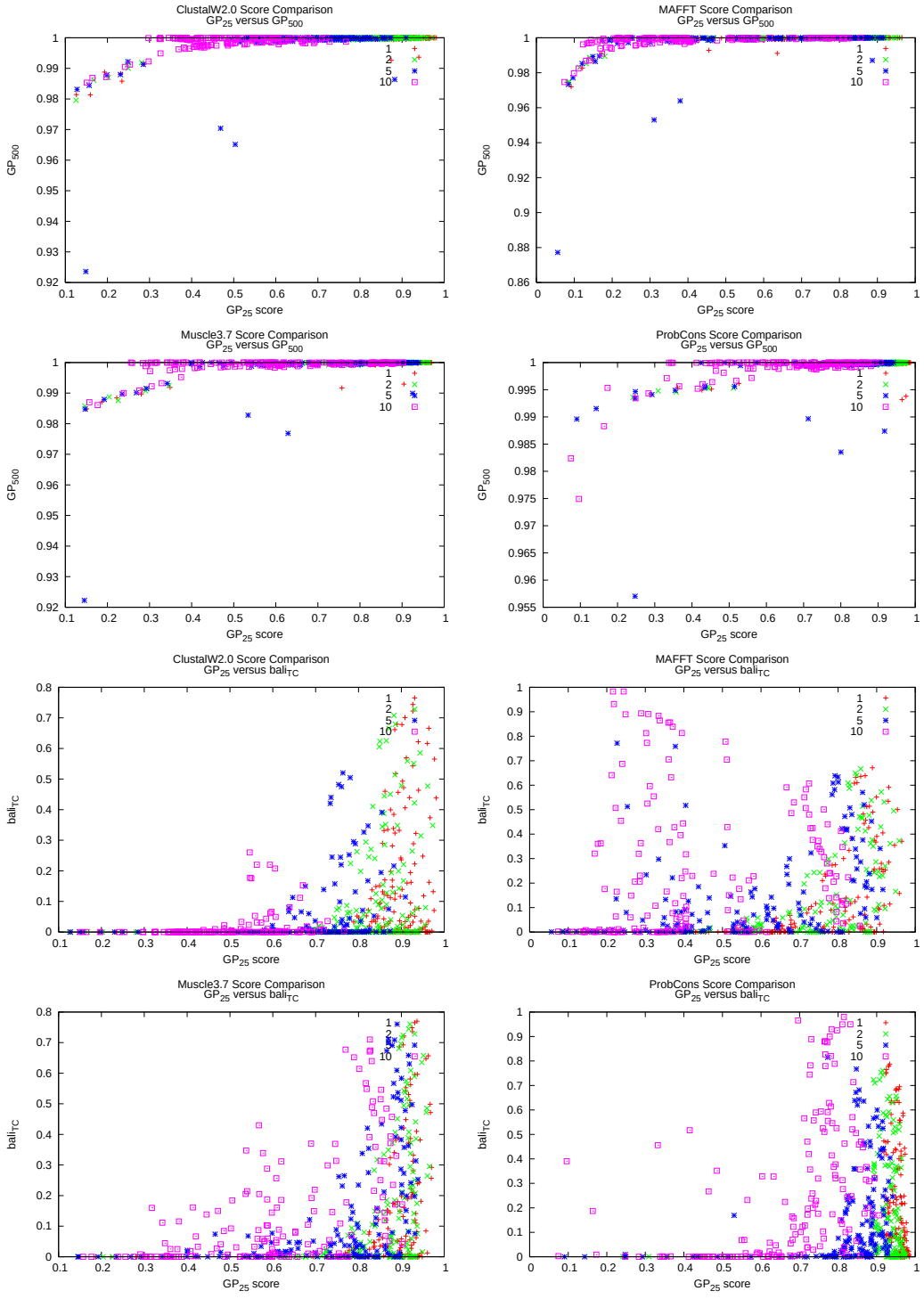

ClustalW2.0 Score Comparison
GP25 versus GP500
MAFFT Score Comparison
GP25 versus GP500
Muscle3.7 Score Comparison
GP25 versus GP500
ProbCons Score Comparison
GP25 versus GP500
ClustalW2.0 Score Comparison
GP25 versus balitc
MAFFT Score Comparison
GP25 versus balitc
Muscle3.7 Score Comparison
GP25 versus balitc
ProbCons Score Comparison
GP25 versus balitc
GP500
balitc
GP25 score
1
2
5
10

Figure B.2

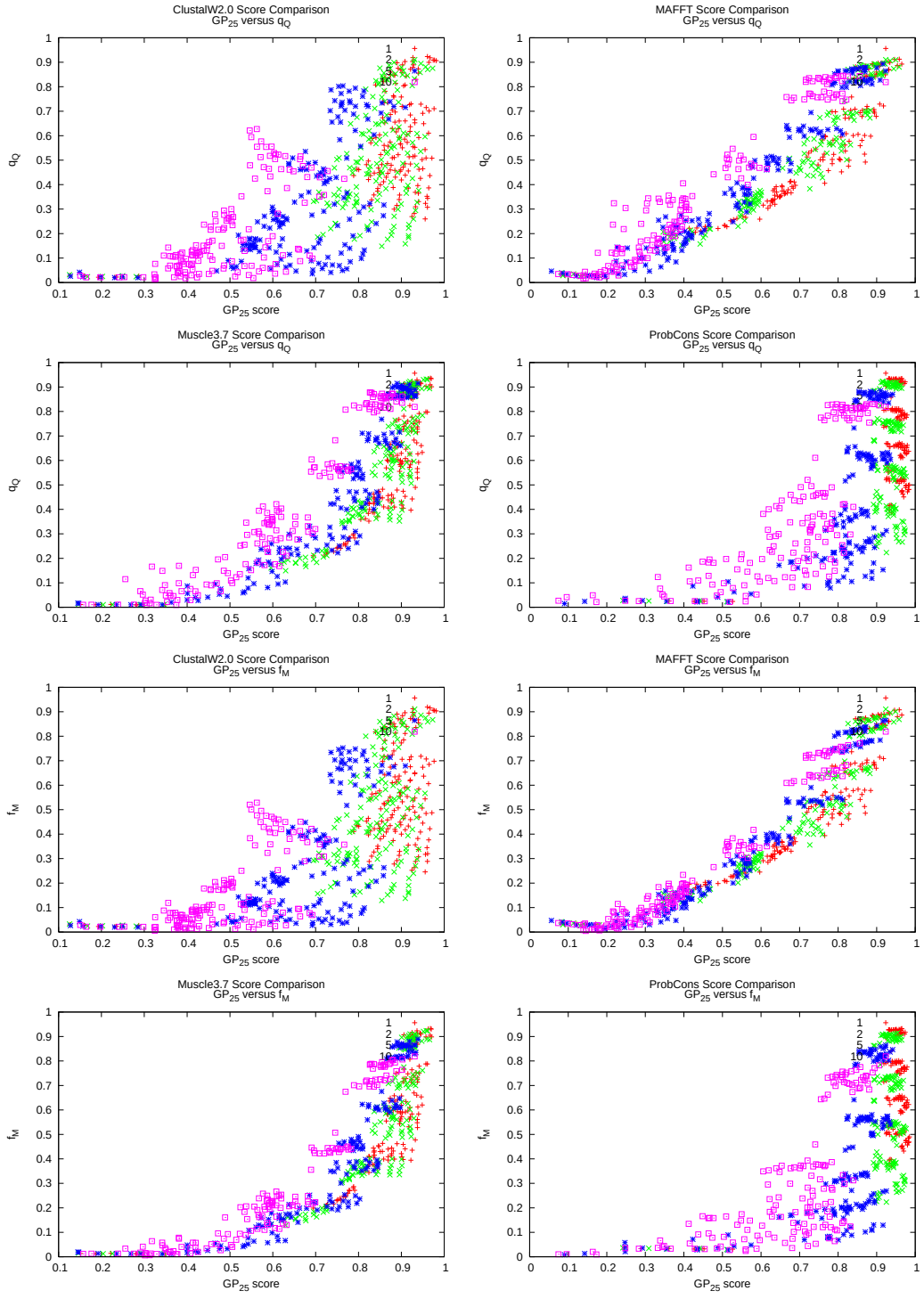

ClustalW2.0 Score Comparison
GP25 versus qQ
MAFFT Score Comparison
GP25 versus qQ
Muscle3.7 Score Comparison
GP25 versus qQ
ProbCons Score Comparison
GP25 versus qQ
ClustalW2.0 Score Comparison
GP25 versus fM
MAFFT Score Comparison
GP25 versus fM
Muscle3.7 Score Comparison
GP25 versus fM
ProbCons Score Comparison
GP25 versus fM
qQ
fM
GP25 score
1
2
5
10

Figure B.2

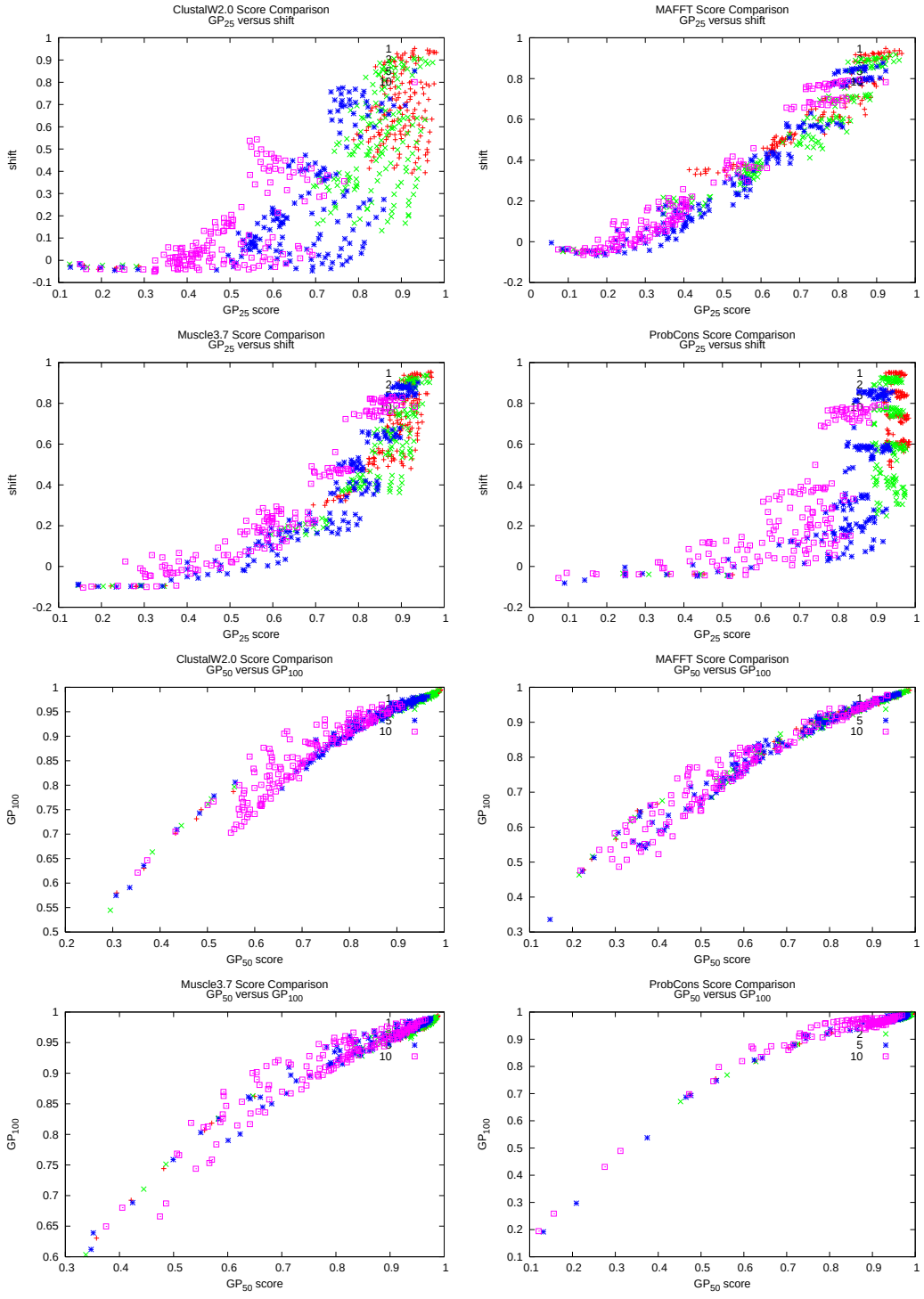

ClustalW2.0 Score Comparison
GP25 versus shift
MAFFT Score Comparison
GP25 versus shift
Muscle3.7 Score Comparison
GP25 versus shift
ProbCons Score Comparison
GP25 versus shift
ClustalW2.0 Score Comparison
GP50 versus GP100
MAFFT Score Comparison
GP50 versus GP100
Muscle3.7 Score Comparison
GP50 versus GP100
ProbCons Score Comparison
GP50 versus GP100
shift
GP25 score
GP100
GP50 score
1
2
5
10

Figure B.2

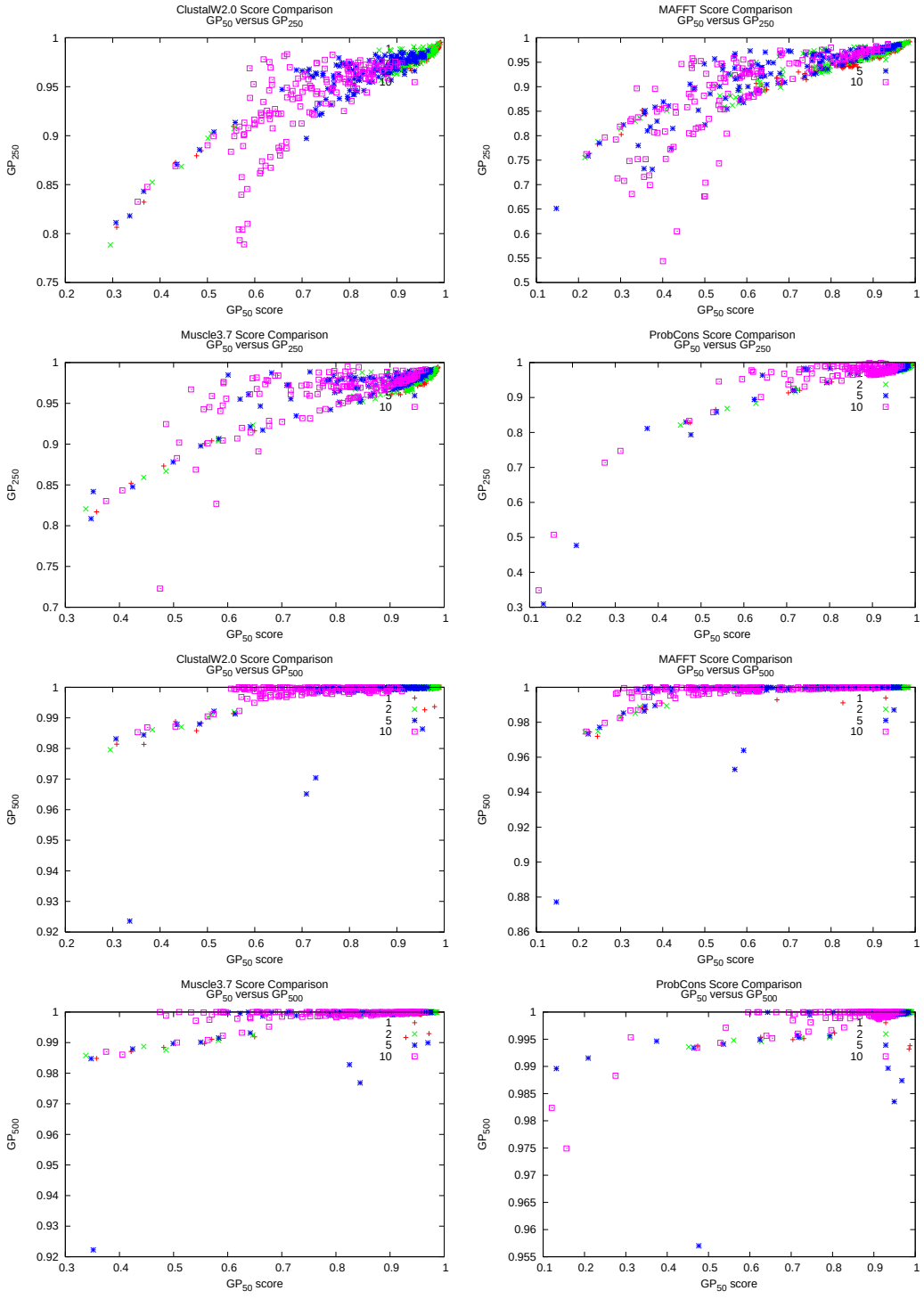

ClustalW2.0 Score Comparison
GP50 versus GP250
MAFFT Score Comparison
GP50 versus GP250
Muscle3.7 Score Comparison
GP50 versus GP250
ProbCons Score Comparison
GP50 versus GP250
ClustalW2.0 Score Comparison
GP50 versus GP500
MAFFT Score Comparison
GP50 versus GP500
Muscle3.7 Score Comparison
GP50 versus GP500
ProbCons Score Comparison
GP50 versus GP500
GP250
GP500
GP50 score
1
2
5
10

Figure B.2

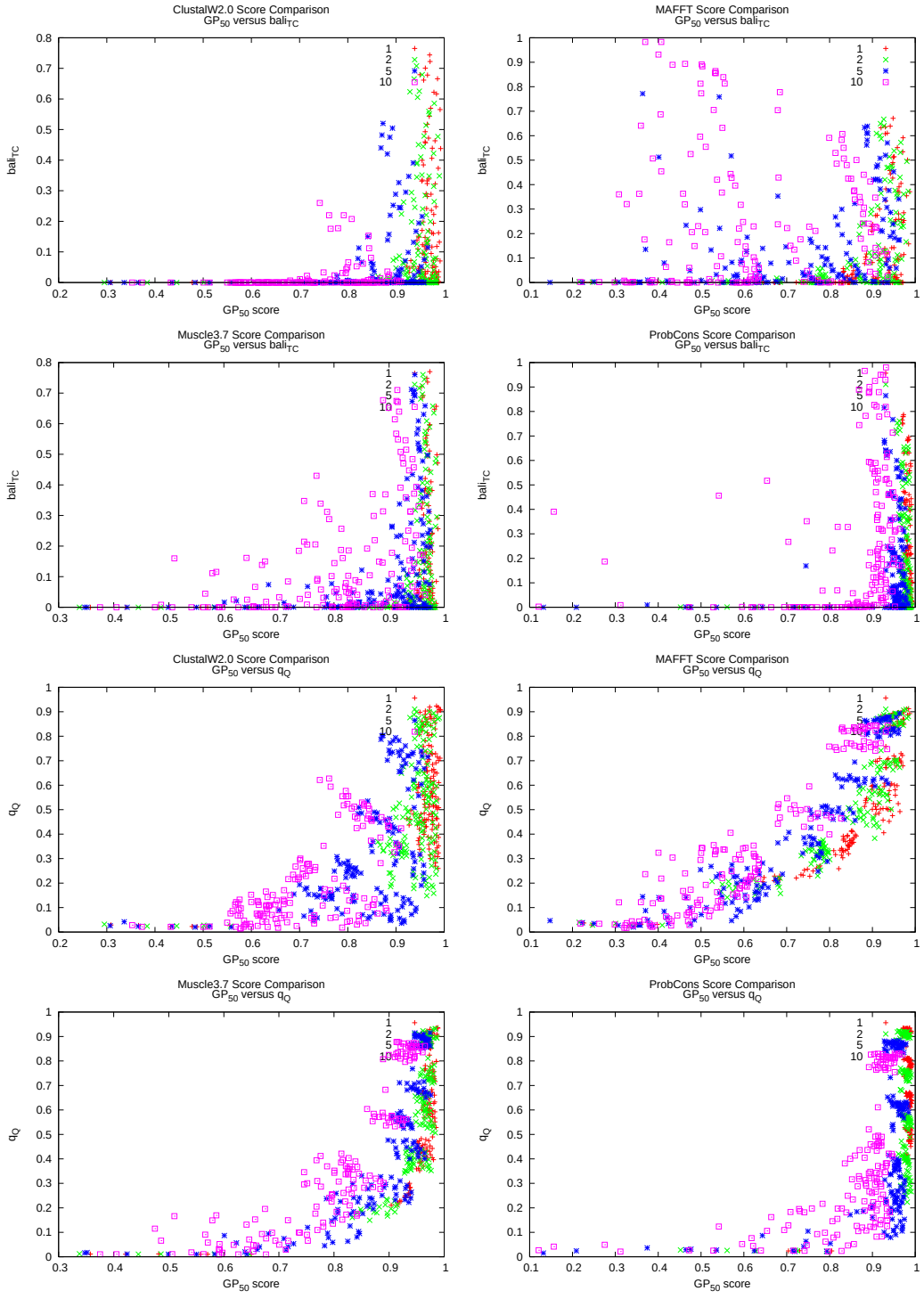

ClustalW2.0 Score Comparison
GP50 versus baliTC
MAFFT Score Comparison
GP50 versus baliTC
Muscle3.7 Score Comparison
GP50 versus baliTC
ProbCons Score Comparison
GP50 versus baliTC
ClustalW2.0 Score Comparison
GP50 versus qQ
MAFFT Score Comparison
GP50 versus qQ
Muscle3.7 Score Comparison
GP50 versus qQ
ProbCons Score Comparison
GP50 versus qQ
baliTC
qQ
GP50 score
1
2
5
10

Figure B.2

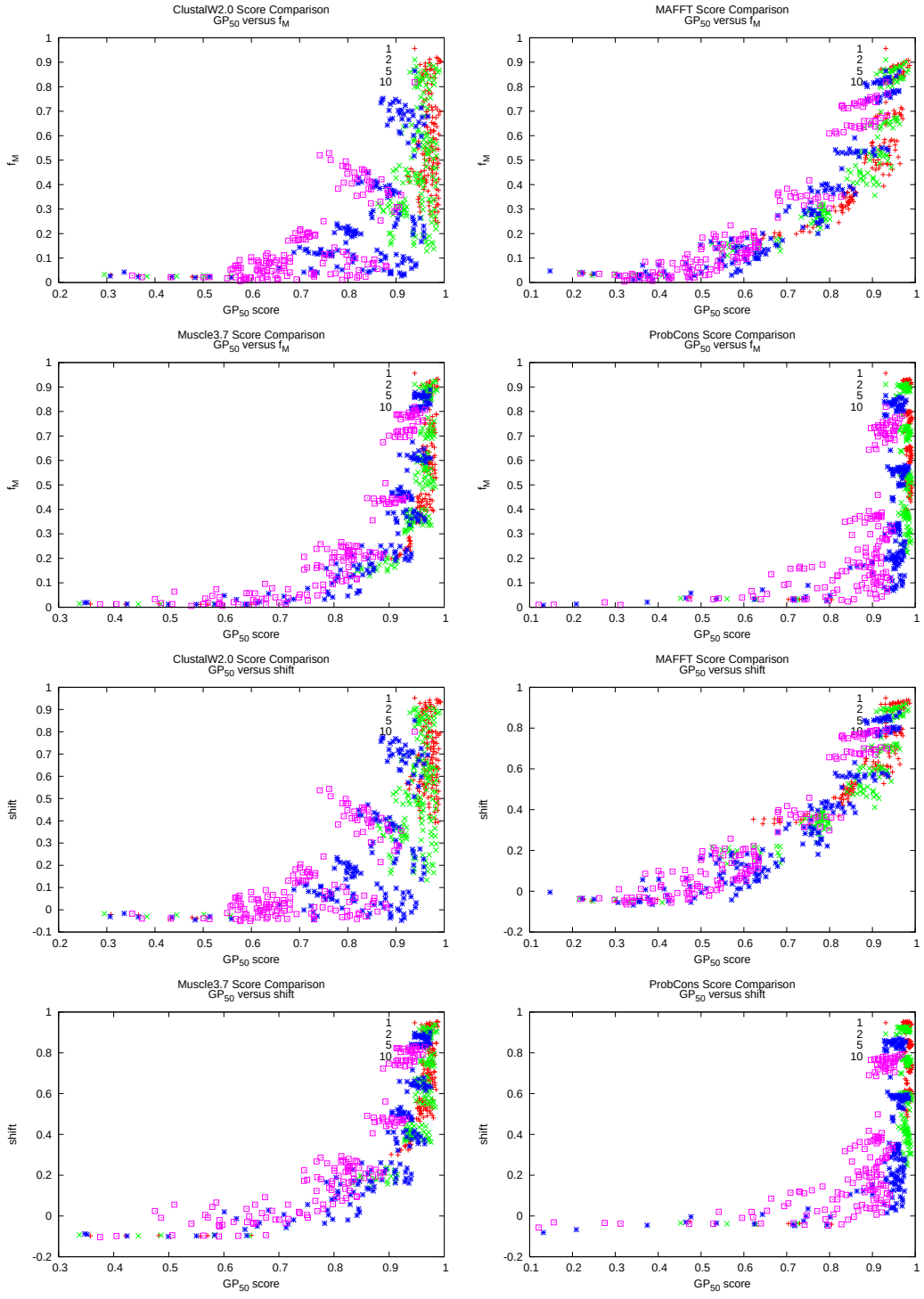

ClustalW2.0 Score Comparison
GP50 versus fM
MAFFT Score Comparison
GP50 versus fM
Muscle3.7 Score Comparison
GP50 versus fM
ProbCons Score Comparison
GP50 versus fM
ClustalW2.0 Score Comparison
GP50 versus shift
MAFFT Score Comparison
GP50 versus shift
Muscle3.7 Score Comparison
GP50 versus shift
ProbCons Score Comparison
GP50 versus shift
fM
shift
GP50 score
1
2
5
10

Figure B.2

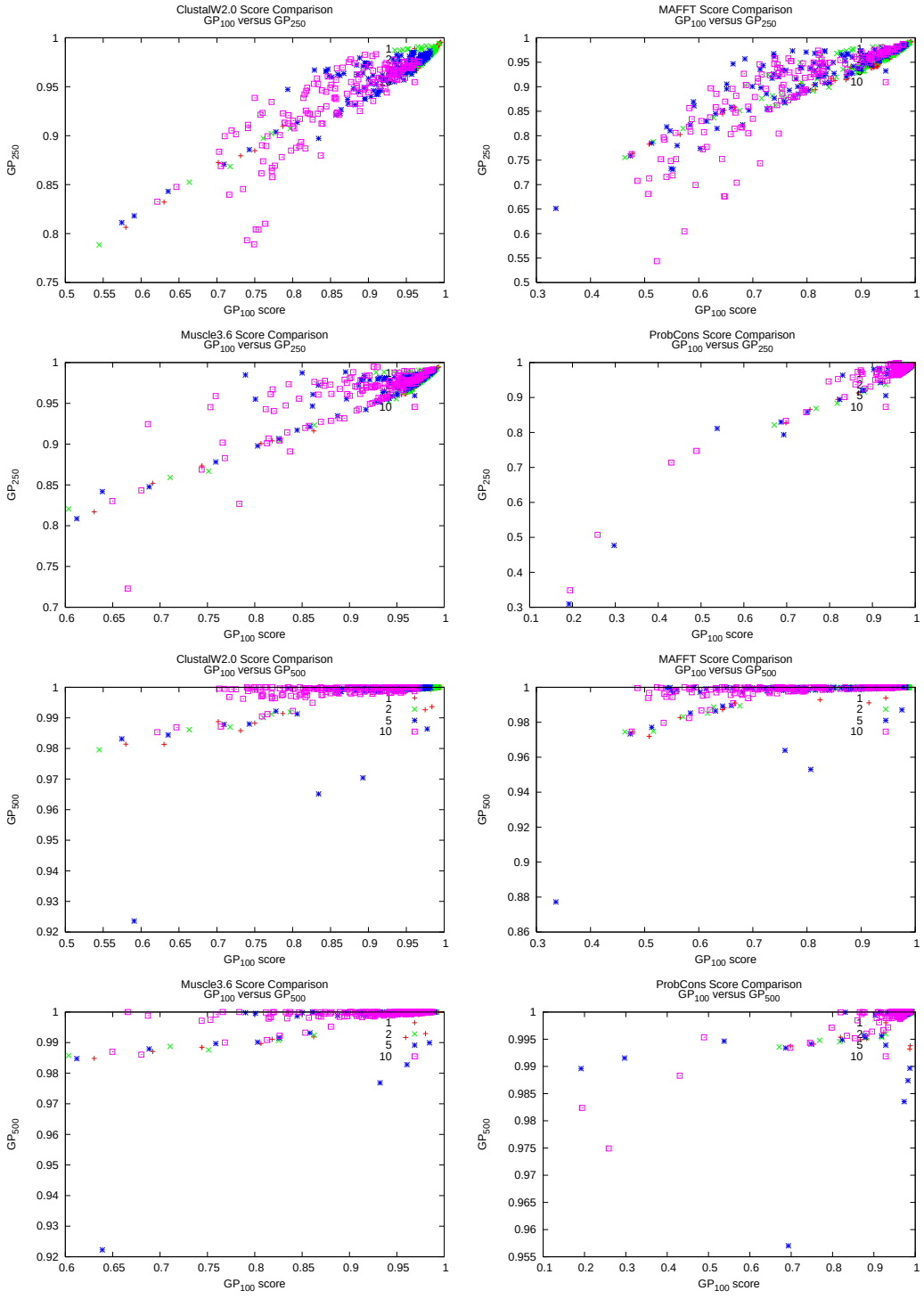

ClustalW2.0 Score Comparison
GP100 versus GP250
MAFFT Score Comparison
GP100 versus GP250
Muscle3.6 Score Comparison
GP100 versus GP250
ProbCons Score Comparison
GP100 versus GP250
ClustalW2.0 Score Comparison
GP100 versus GP500
MAFFT Score Comparison
GP100 versus GP500
Muscle3.6 Score Comparison
GP100 versus GP500
ProbCons Score Comparison
GP100 versus GP500
GP100 score
GP250
GP500
1
2
5
10

Figure B.2

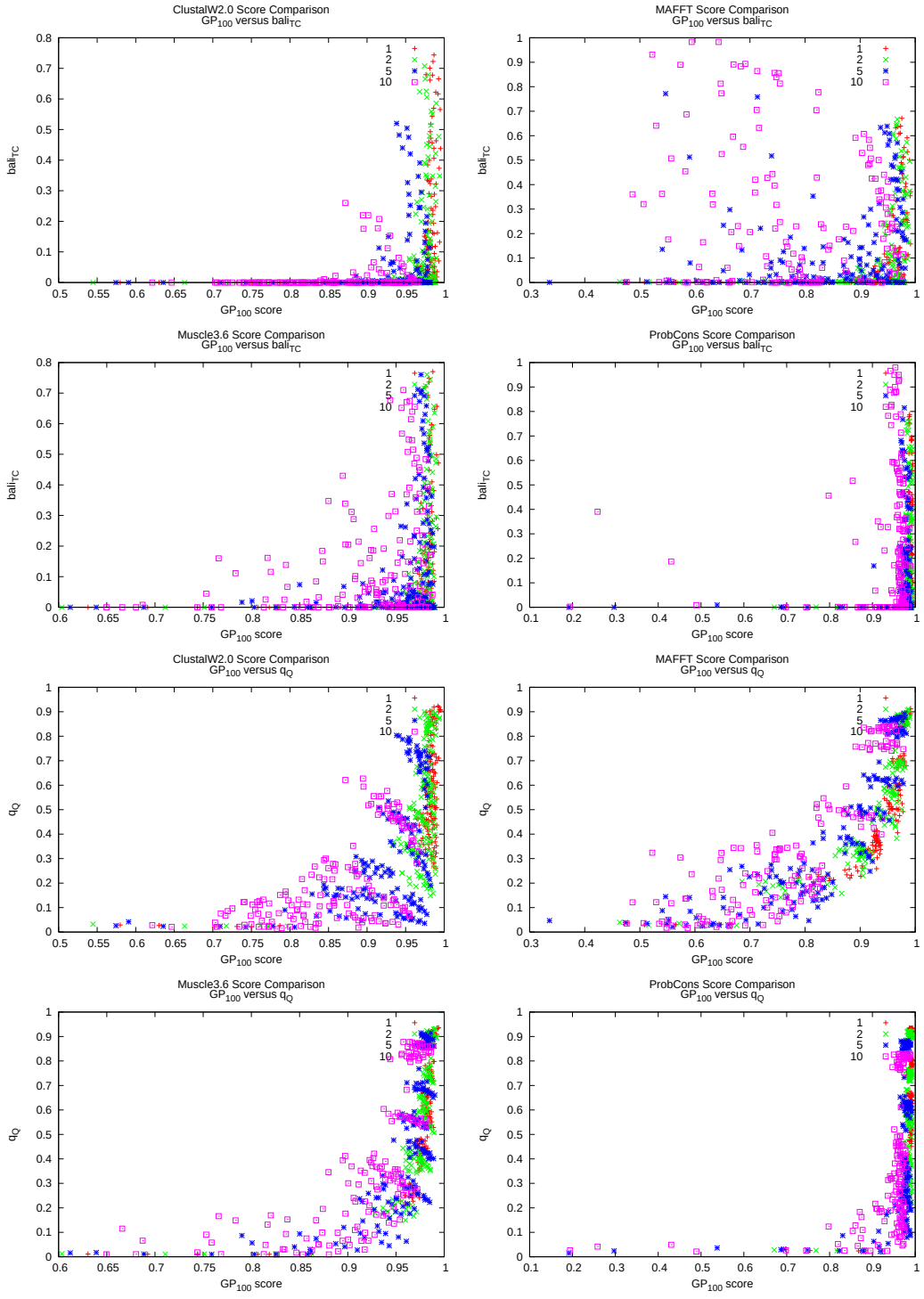

ClustalW2.0 Score Comparison
GP100 versus bali_TC
MAFFT Score Comparison
GP100 versus bali_TC
Muscle3.6 Score Comparison
GP100 versus bali_TC
ProbCons Score Comparison
GP100 versus bali_TC
ClustalW2.0 Score Comparison
GP100 versus q_Q
MAFFT Score Comparison
GP100 versus q_Q
Muscle3.6 Score Comparison
GP100 versus q_Q
ProbCons Score Comparison
GP100 versus q_Q
bali_TC
q_Q
GP100 score
1
2
5
10

Figure B.2

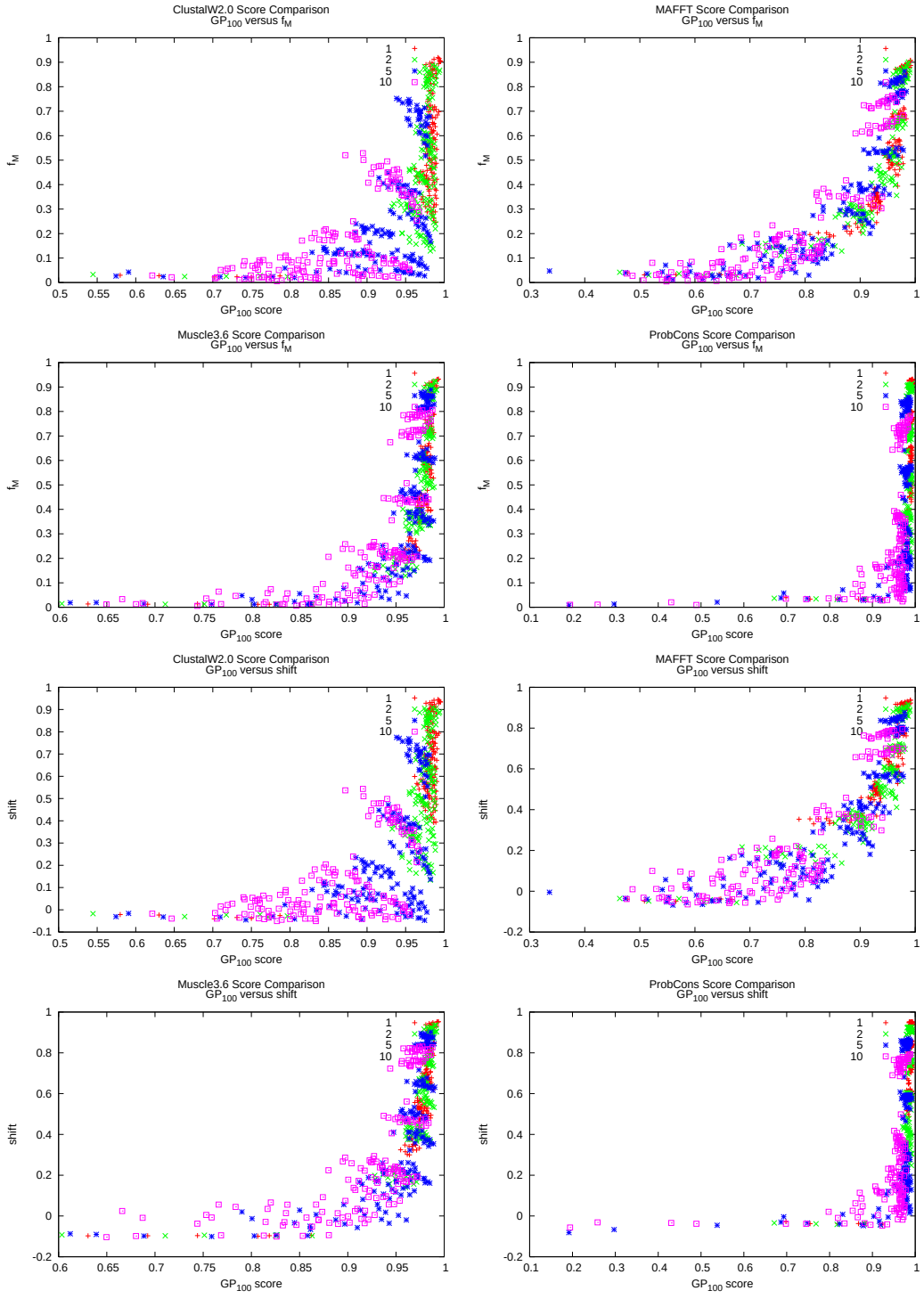

ClustalW2.0 Score Comparison
GP100 versus fM
MAFFT Score Comparison
GP100 versus fM
Muscle3.6 Score Comparison
GP100 versus fM
ProbCons Score Comparison
GP100 versus fM
ClustalW2.0 Score Comparison
GP100 versus shift
MAFFT Score Comparison
GP100 versus shift
Muscle3.6 Score Comparison
GP100 versus shift
ProbCons Score Comparison
GP100 versus shift
fM
shift
GP100 score
1
2
5
10

Figure B.2

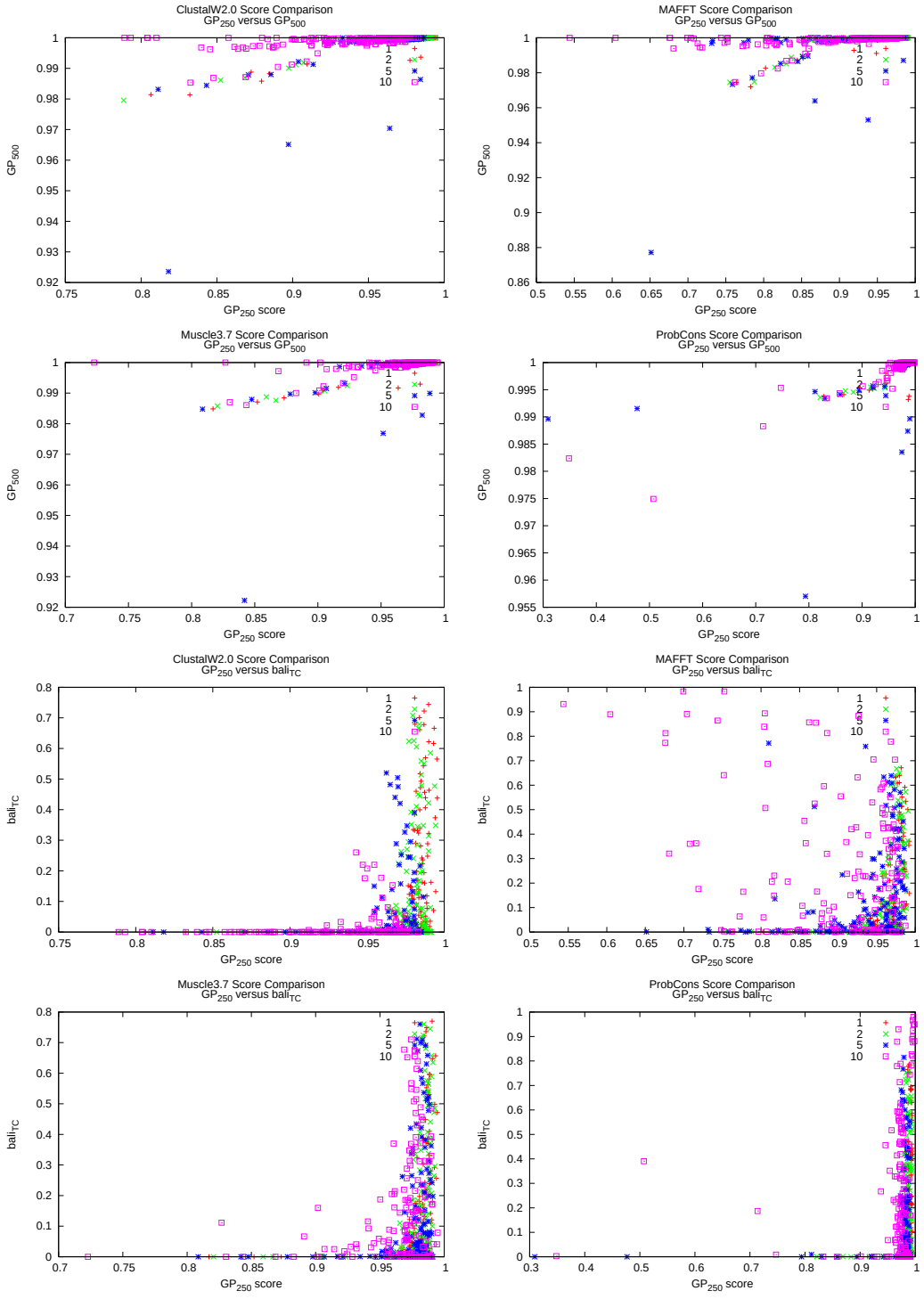

ClustalW2.0 Score Comparison
GP250 versus GP500
MAFFT Score Comparison
GP250 versus GP500
Muscle3.7 Score Comparison
GP250 versus GP500
ProbCons Score Comparison
GP250 versus GP500
ClustalW2.0 Score Comparison
GP250 versus balitc
MAFFT Score Comparison
GP250 versus balitc
Muscle3.7 Score Comparison
GP250 versus balitc
ProbCons Score Comparison
GP250 versus balitc
GP500
balitc
GP250 score
1
2
5
10

Figure B.2

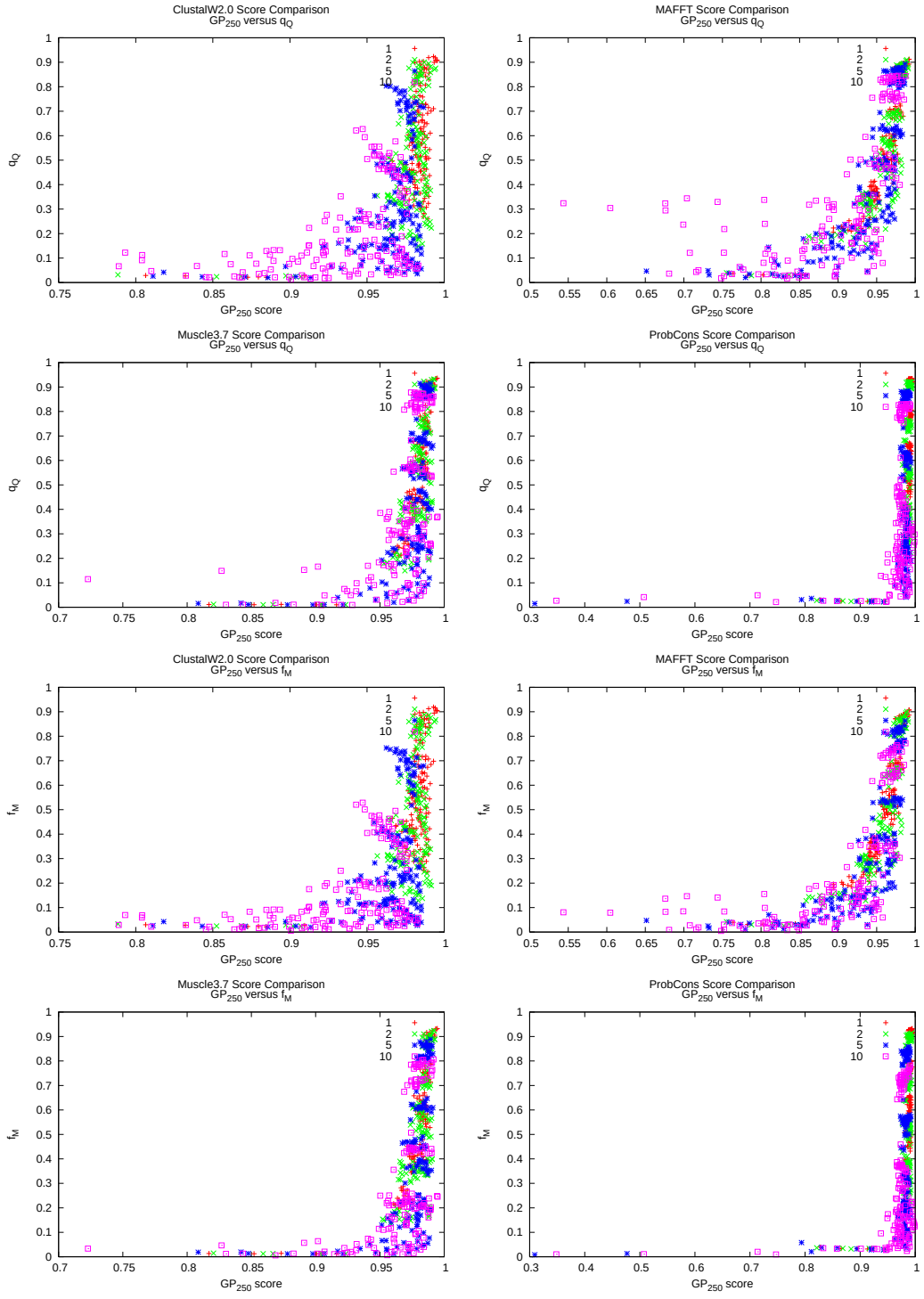

ClustalW2.0 Score Comparison
GP250 versus qQ
MAFFT Score Comparison
GP250 versus qQ
Muscle3.7 Score Comparison
GP250 versus qQ
ProbCons Score Comparison
GP250 versus qQ
ClustalW2.0 Score Comparison
GP250 versus fM
MAFFT Score Comparison
GP250 versus fM
Muscle3.7 Score Comparison
GP250 versus fM
ProbCons Score Comparison
GP250 versus fM
qQ
fM
GP250 score
1
2
5
10

Figure B.2

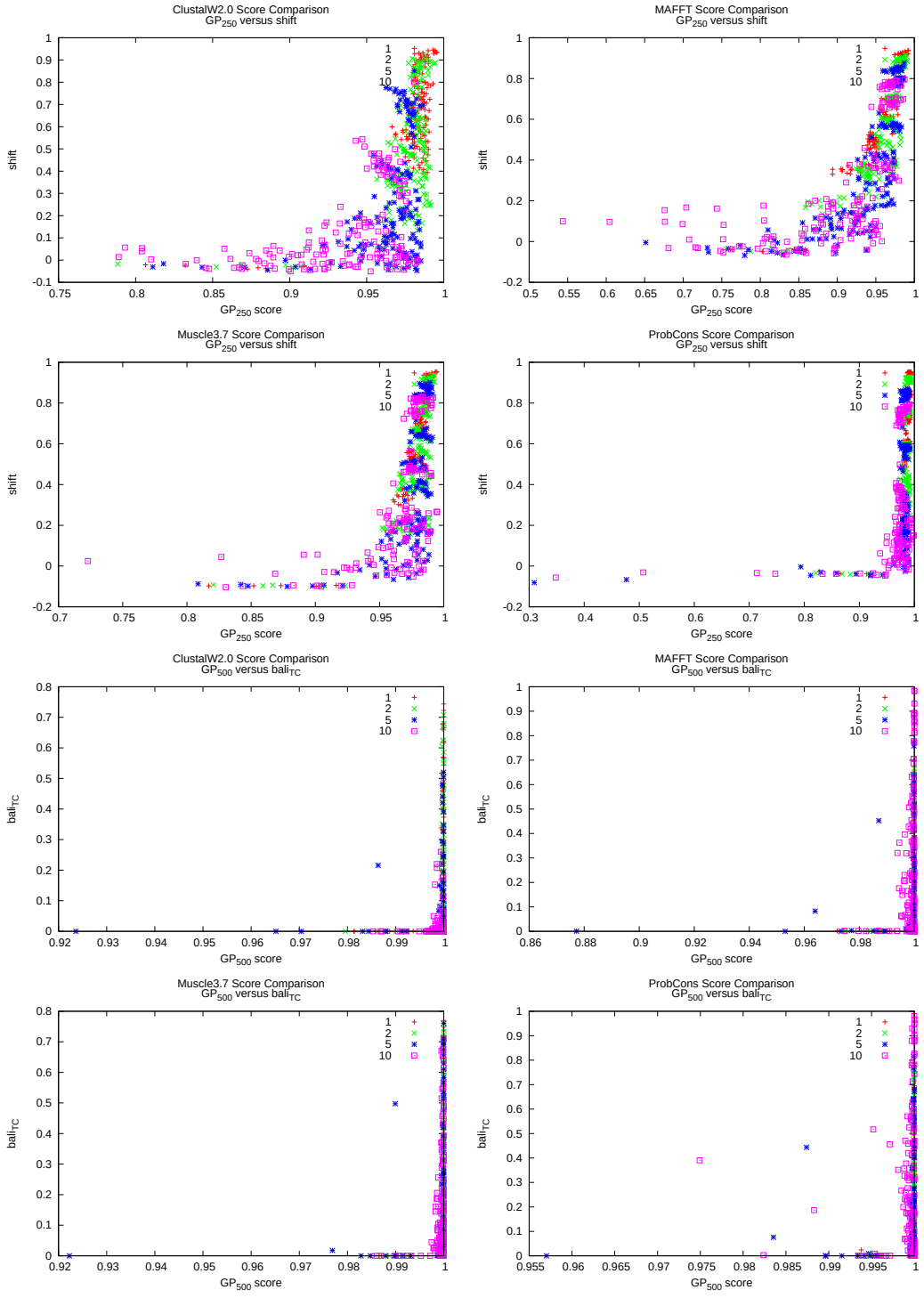

ClustalW2.0 Score Comparison
GP250 versus shift
MAFFT Score Comparison
GP250 versus shift
Muscle3.7 Score Comparison
GP250 versus shift
ProbCons Score Comparison
GP250 versus shift
ClustalW2.0 Score Comparison
GP500 versus balitc
MAFFT Score Comparison
GP500 versus balitc
Muscle3.7 Score Comparison
GP500 versus balitc
ProbCons Score Comparison
GP500 versus balitc
shift
balitc
GP250 score
GP500 score
1
2
5
10

Figure B.2

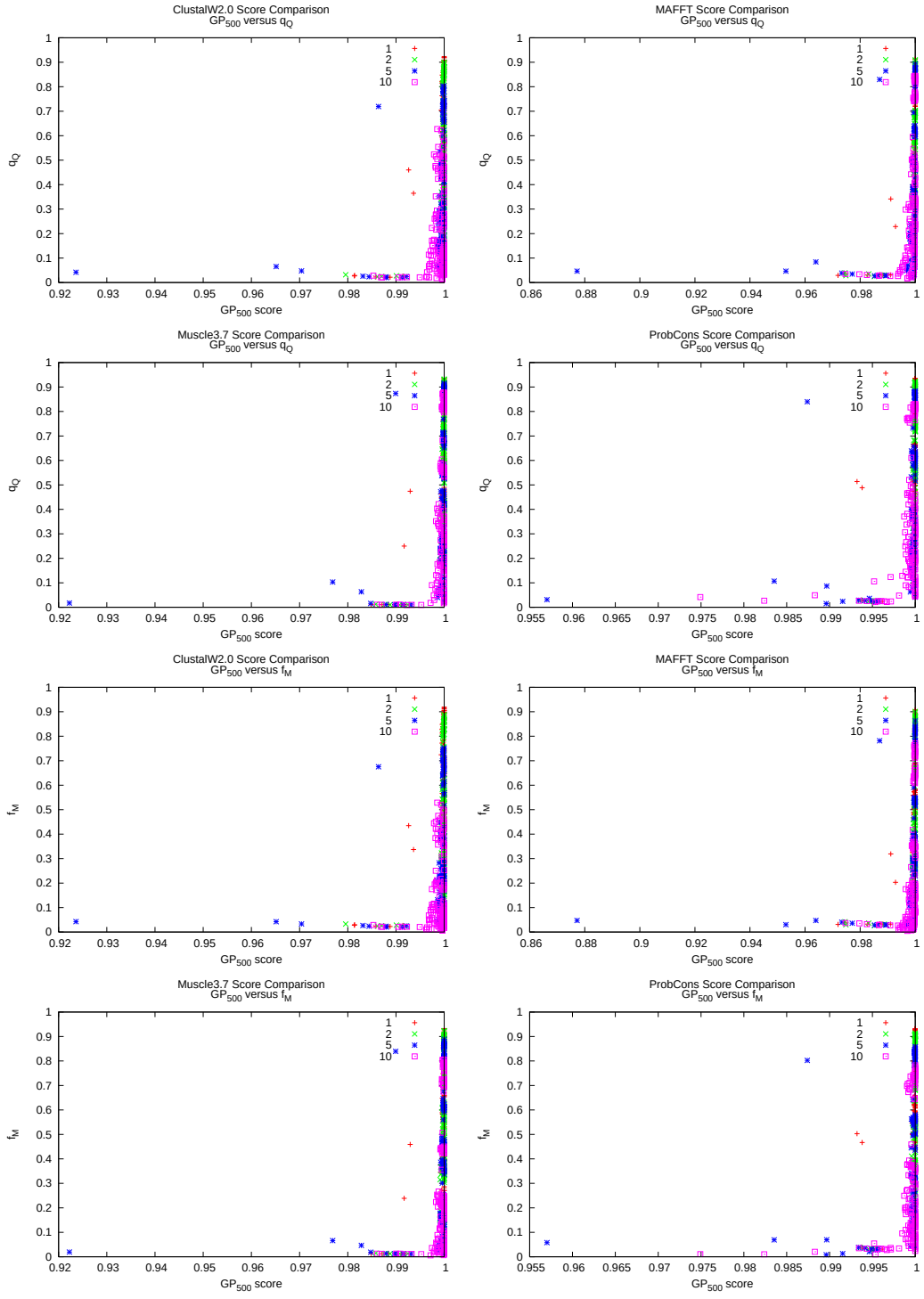

ClustalW2.0 Score Comparison
GP500 versus qQ
MAFFT Score Comparison
GP500 versus qQ
Muscle3.7 Score Comparison
GP500 versus qQ
ProbCons Score Comparison
GP500 versus qQ
ClustalW2.0 Score Comparison
GP500 versus fM
MAFFT Score Comparison
GP500 versus fM
Muscle3.7 Score Comparison
GP500 versus fM
ProbCons Score Comparison
GP500 versus fM
qQ
fM
GP500 score
1
2
5
10

Figure B.2

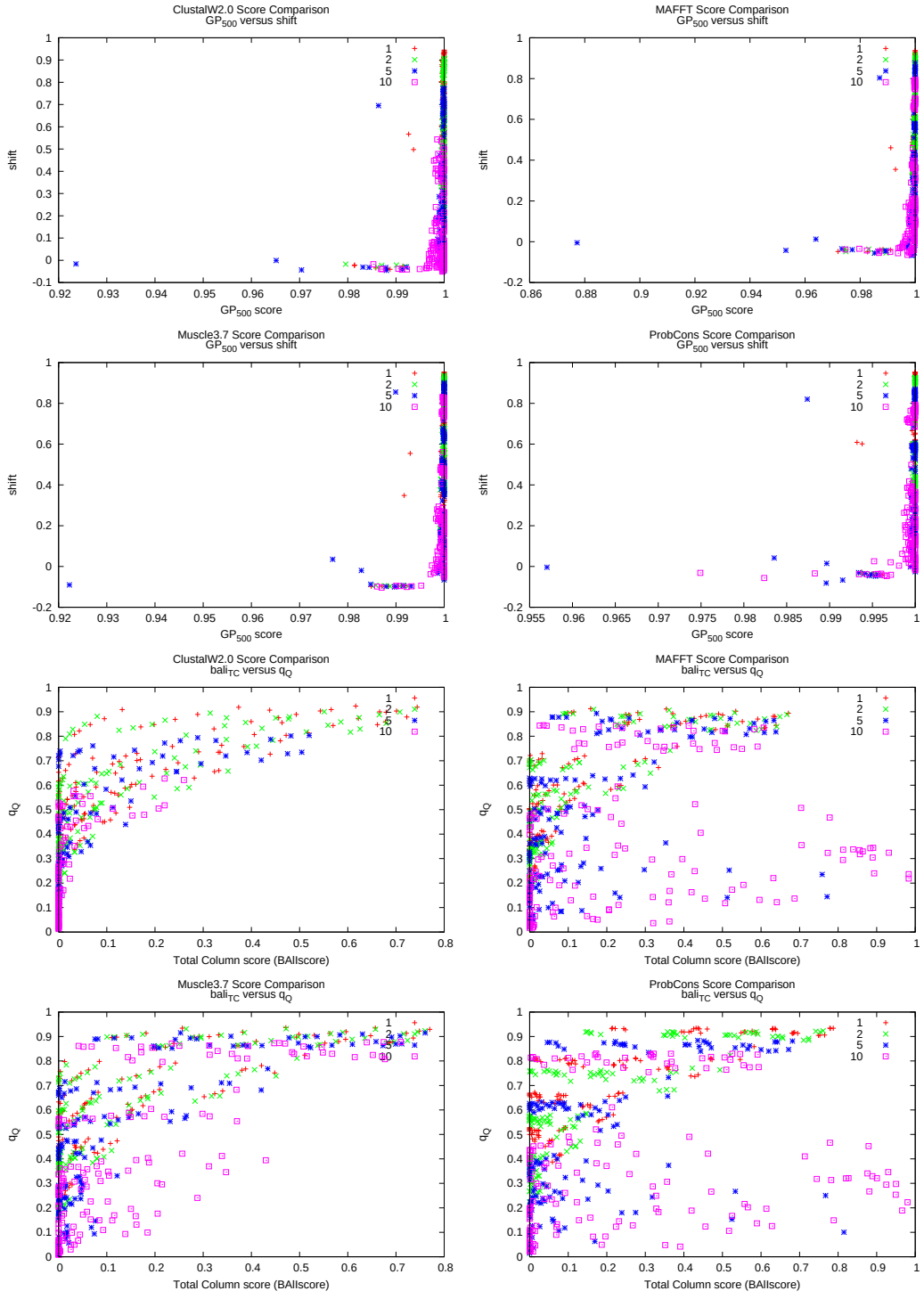

ClustalW2.0 Score Comparison
GP500 versus shift
MAFFT Score Comparison
GP500 versus shift
Muscle3.7 Score Comparison
GP500 versus shift
ProbCons Score Comparison
GP500 versus shift
ClustalW2.0 Score Comparison
balitc versus qQ
MAFFT Score Comparison
balitc versus qQ
Muscle3.7 Score Comparison
balitc versus qQ
ProbCons Score Comparison
balitc versus qQ
shift
GP500 score
qQ
Total Column score (BAIIscore)
1
2
5
10

Figure B.2

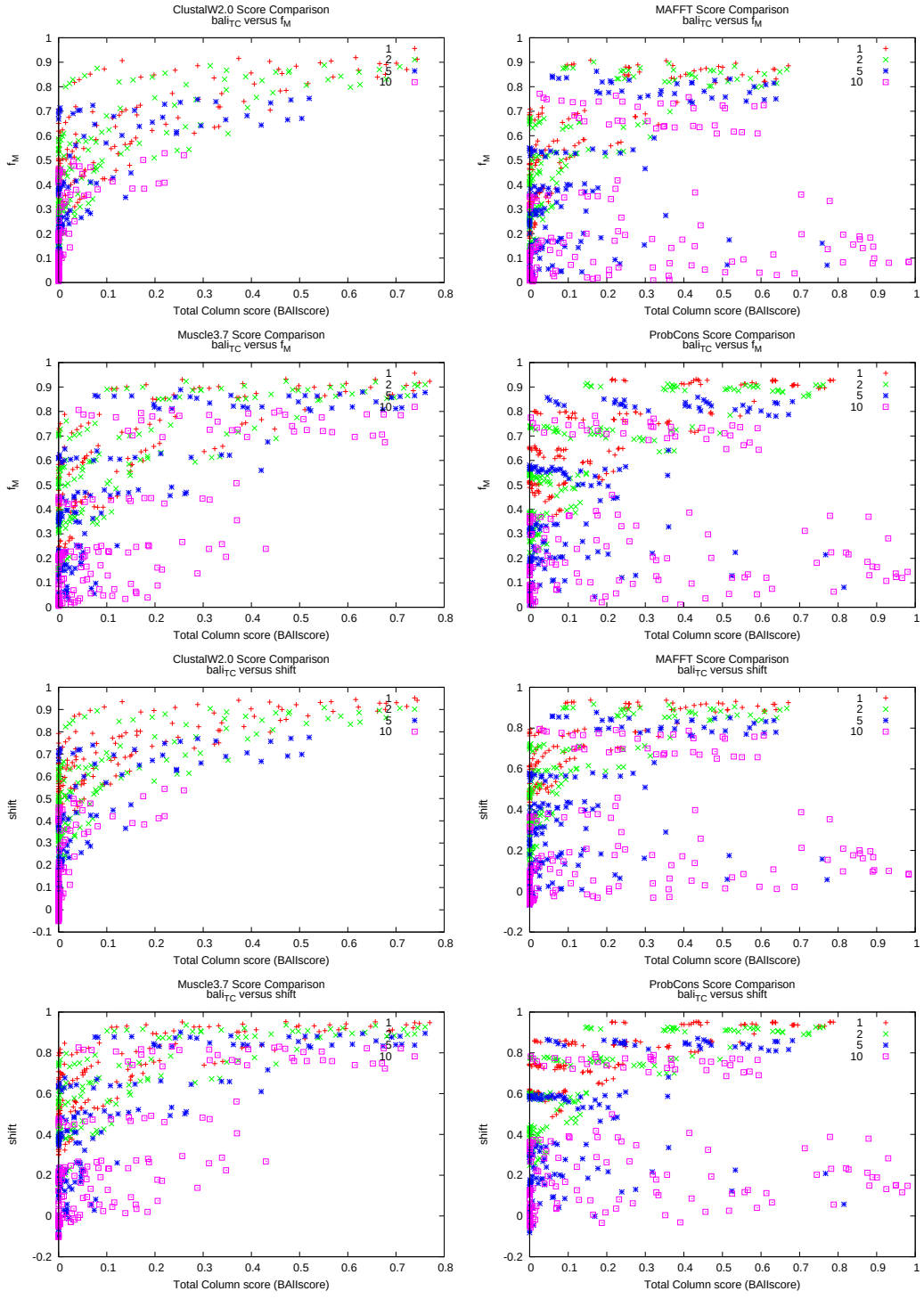

ClustalW2.0 Score Comparison
bali_TC versus f_M
MAFFT Score Comparison
bali_TC versus f_M
Muscle3.7 Score Comparison
bali_TC versus f_M
ProbCons Score Comparison
bali_TC versus f_M
ClustalW2.0 Score Comparison
bali_TC versus shift
MAFFT Score Comparison
bali_TC versus shift
Muscle3.7 Score Comparison
bali_TC versus shift
ProbCons Score Comparison
bali_TC versus shift
f_M
shift
Total Column score (BAIIscore)
1
2
5
10

Figure B.2

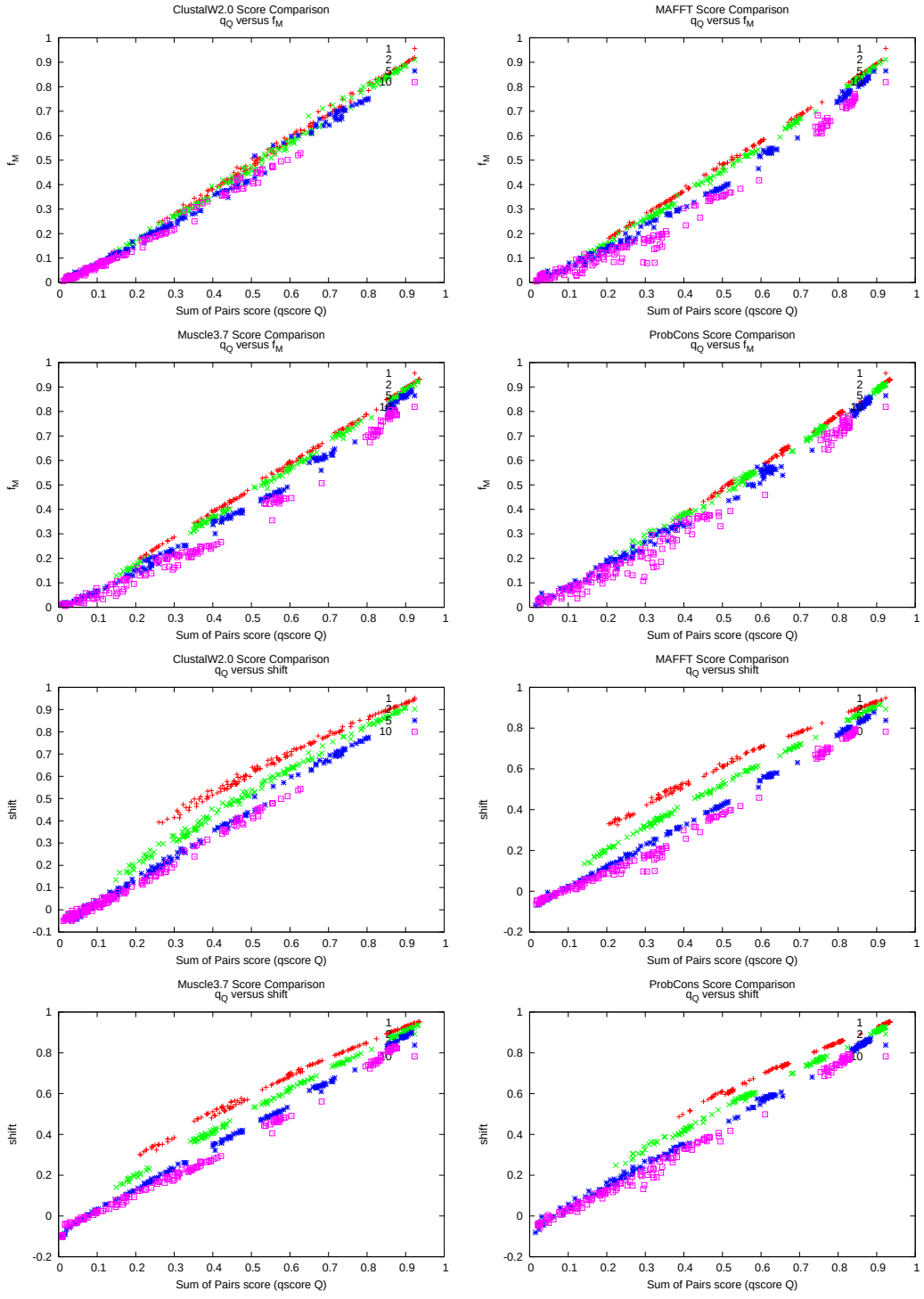

ClustalW2.0 Score Comparison
$q_Q$ versus $f_M$
MAFFT Score Comparison
$q_Q$ versus $f_M$
Muscle3.7 Score Comparison
$q_Q$ versus $f_M$
ProbCons Score Comparison
$q_Q$ versus $f_M$
ClustalW2.0 Score Comparison
$q_Q$ versus shift
MAFFT Score Comparison
$q_Q$ versus shift
Muscle3.7 Score Comparison
$q_Q$ versus shift
ProbCons Score Comparison
$q_Q$ versus shift
$f_M$
shift
Sum of Pairs score (qscore Q)
1
2
5
10

Figure B.2

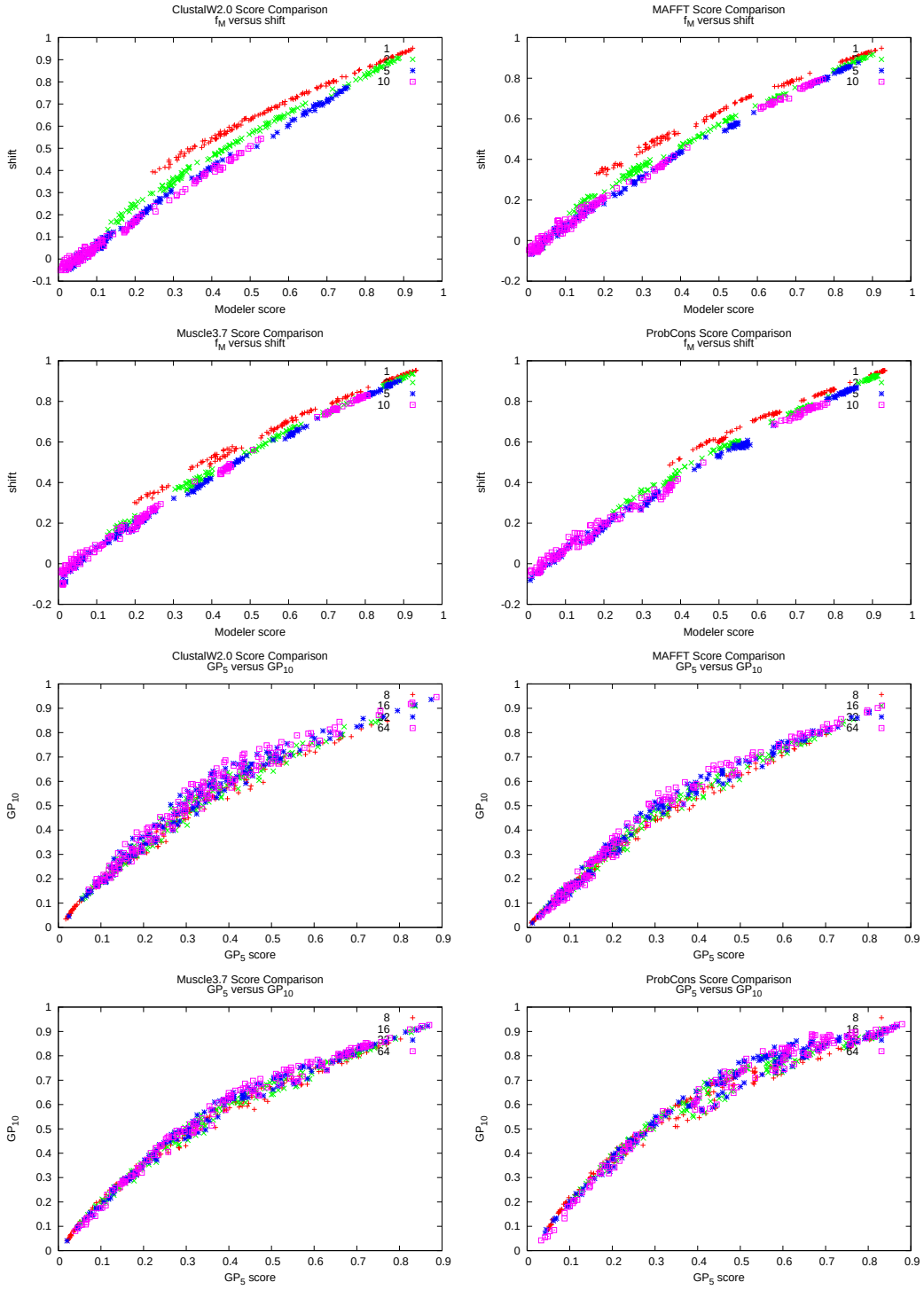

ClustalW2.0 Score Comparison
$f_M$ versus shift
MAFFT Score Comparison
$f_M$ versus shift
Muscle3.7 Score Comparison
$f_M$ versus shift
ProbCons Score Comparison
$f_M$ versus shift
ClustalW2.0 Score Comparison
$GP_5$ versus $GP_{10}$
MAFFT Score Comparison
$GP_5$ versus $GP_{10}$
Muscle3.7 Score Comparison
$GP_5$ versus $GP_{10}$
ProbCons Score Comparison
$GP_5$ versus $GP_{10}$
shift
Modeler score
$GP_{10}$
$GP_5$ score
1
2
5
10
8
16
32
64

Figure B.2

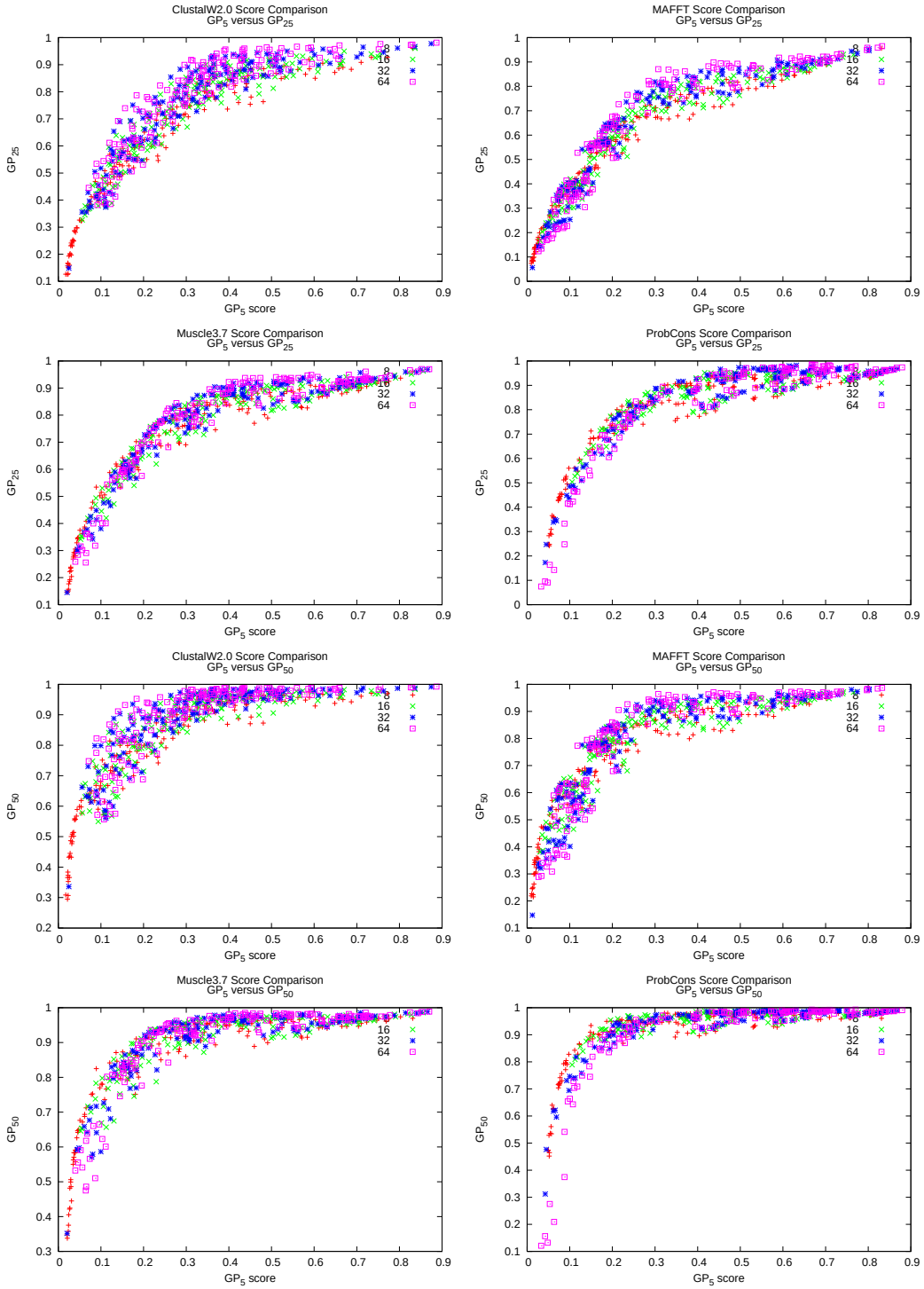

ClustalW2.0 Score Comparison
GP5 versus GP25
MAFFT Score Comparison
GP5 versus GP25
Muscle3.7 Score Comparison
GP5 versus GP25
ProbCons Score Comparison
GP5 versus GP25
ClustalW2.0 Score Comparison
GP5 versus GP50
MAFFT Score Comparison
GP5 versus GP50
Muscle3.7 Score Comparison
GP5 versus GP50
ProbCons Score Comparison
GP5 versus GP50
GP25
GP50
GP5 score
8
16
32
64

Figure B.2

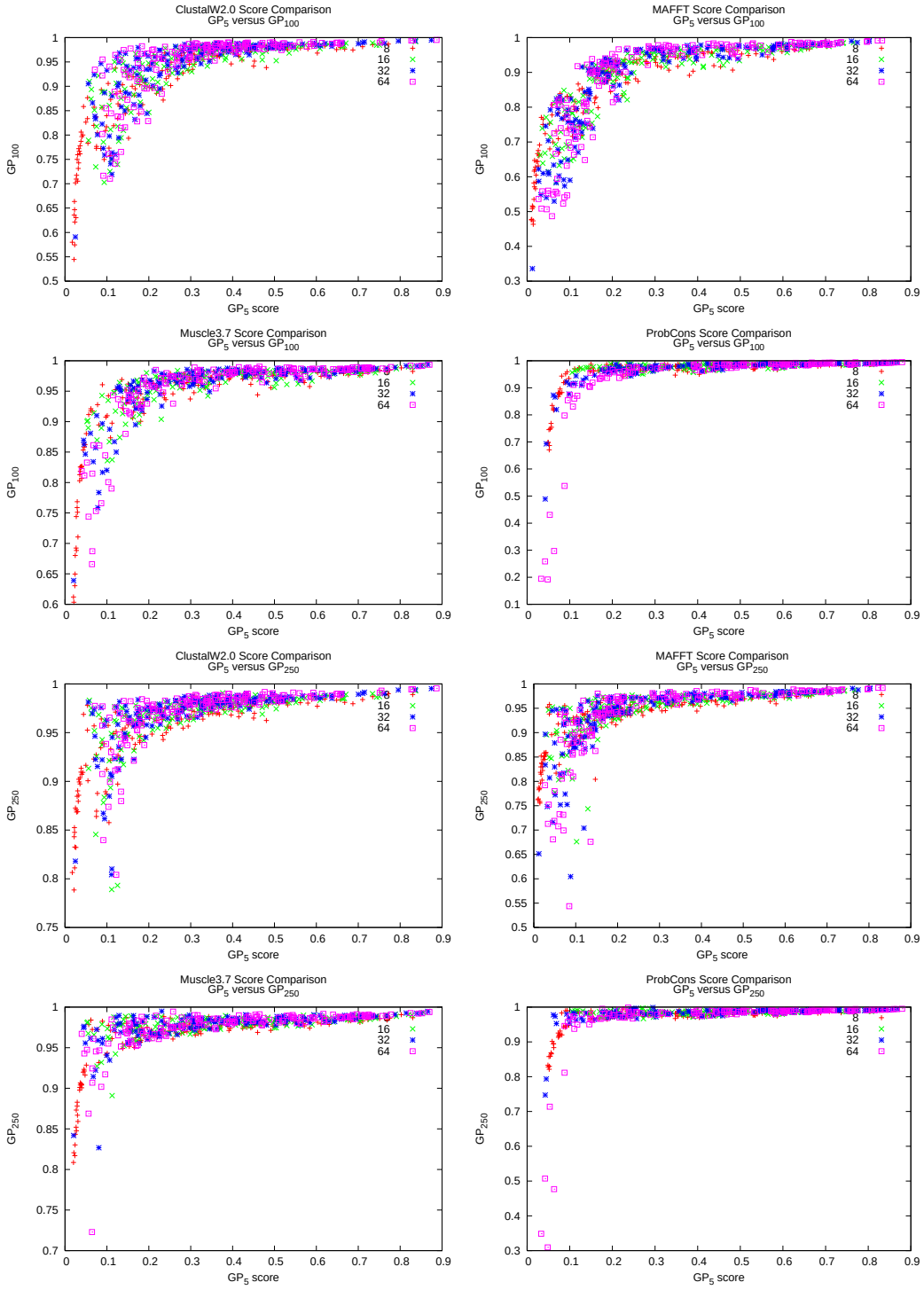

ClustalW2.0 Score Comparison
GP5 versus GP100
MAFFT Score Comparison
GP5 versus GP100
Muscle3.7 Score Comparison
GP5 versus GP100
ProbCons Score Comparison
GP5 versus GP100
ClustalW2.0 Score Comparison
GP5 versus GP250
MAFFT Score Comparison
GP5 versus GP250
Muscle3.7 Score Comparison
GP5 versus GP250
ProbCons Score Comparison
GP5 versus GP250
GP100
GP250
GP5 score
8
16
32
64

Figure B.2

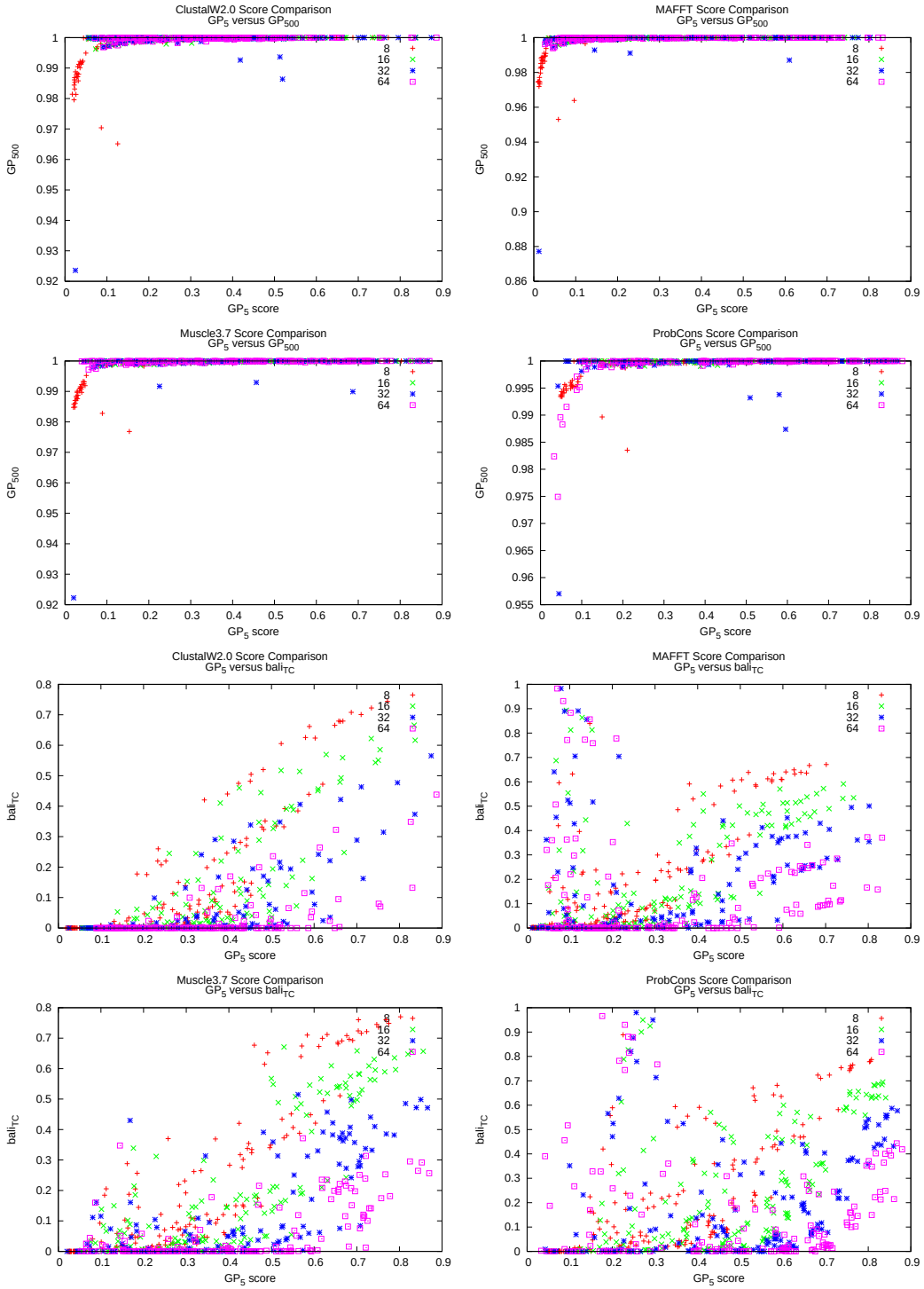

ClustalW2.0 Score Comparison
GP5 versus GP500
MAFFT Score Comparison
GP5 versus GP500
Muscle3.7 Score Comparison
GP5 versus GP500
ProbCons Score Comparison
GP5 versus GP500
ClustalW2.0 Score Comparison
GP5 versus balitc
MAFFT Score Comparison
GP5 versus balitc
Muscle3.7 Score Comparison
GP5 versus balitc
ProbCons Score Comparison
GP5 versus balitc
GP500
balitc
GP5 score
8
16
32
64

Figure B.2

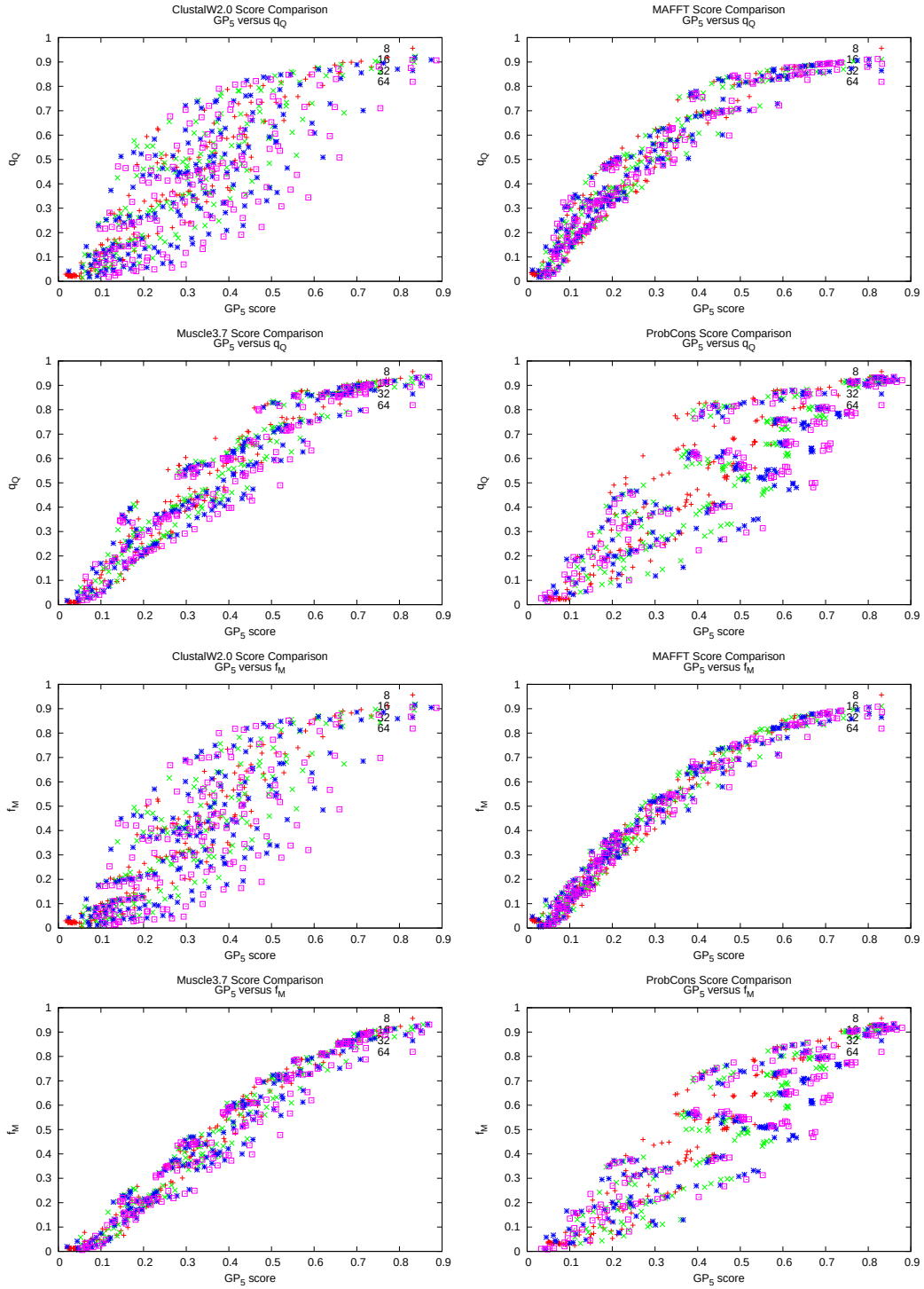

ClustalW2.0 Score Comparison
GP5 versus qQ
MAFFT Score Comparison
GP5 versus qQ
Muscle3.7 Score Comparison
GP5 versus qQ
ProbCons Score Comparison
GP5 versus qQ
ClustalW2.0 Score Comparison
GP5 versus fM
MAFFT Score Comparison
GP5 versus fM
Muscle3.7 Score Comparison
GP5 versus fM
ProbCons Score Comparison
GP5 versus fM
qQ
fM
GP5 score
8
16
32
64

Figure B.2

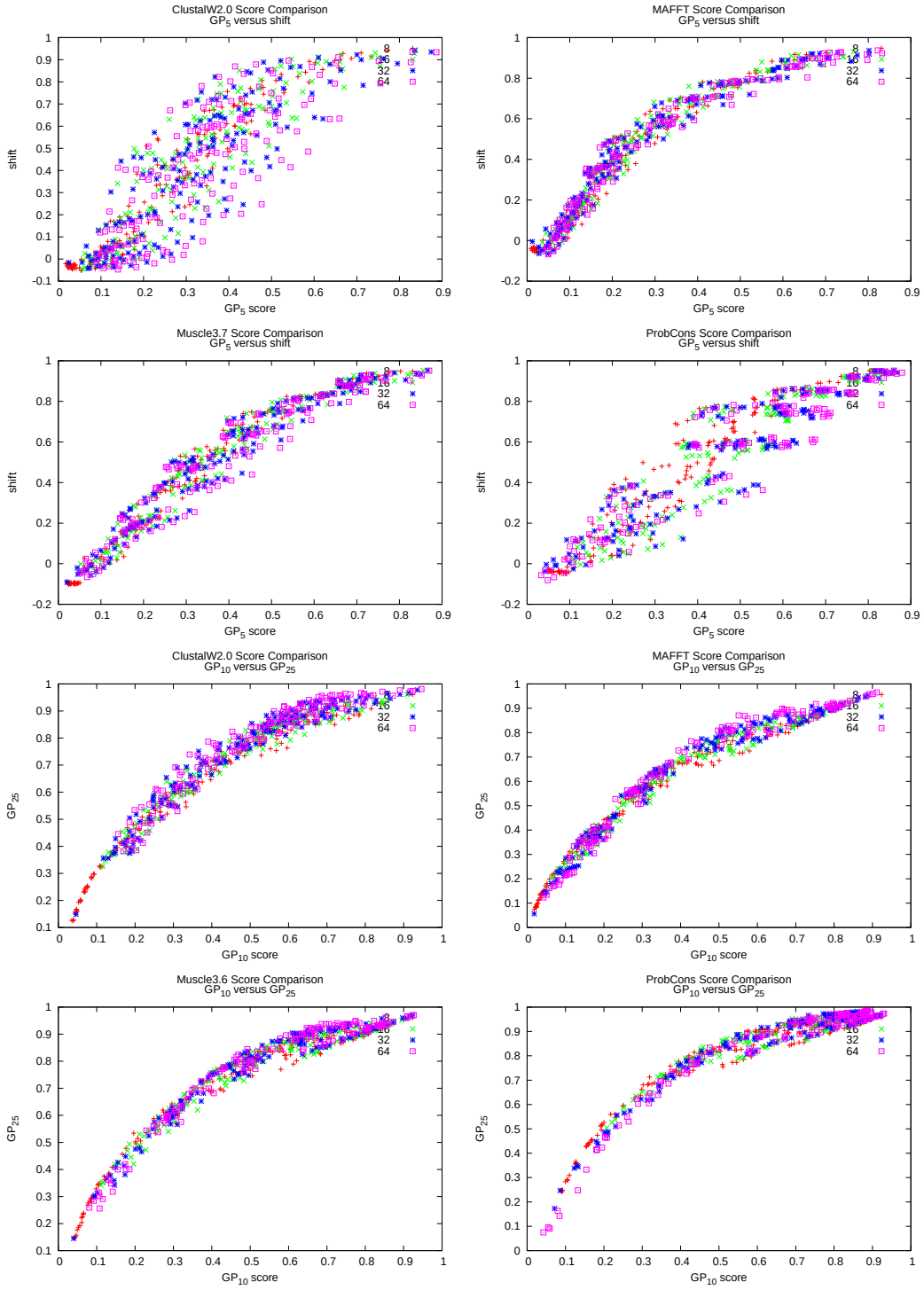

ClustalW2.0 Score Comparison
GP5 versus shift
MAFFT Score Comparison
GP5 versus shift
Muscle3.7 Score Comparison
GP5 versus shift
ProbCons Score Comparison
GP5 versus shift
ClustalW2.0 Score Comparison
GP10 versus GP25
MAFFT Score Comparison
GP10 versus GP25
Muscle3.6 Score Comparison
GP10 versus GP25
ProbCons Score Comparison
GP10 versus GP25
shift
GP25
GP5 score
GP10 score
8
16
32
64

Figure B.2

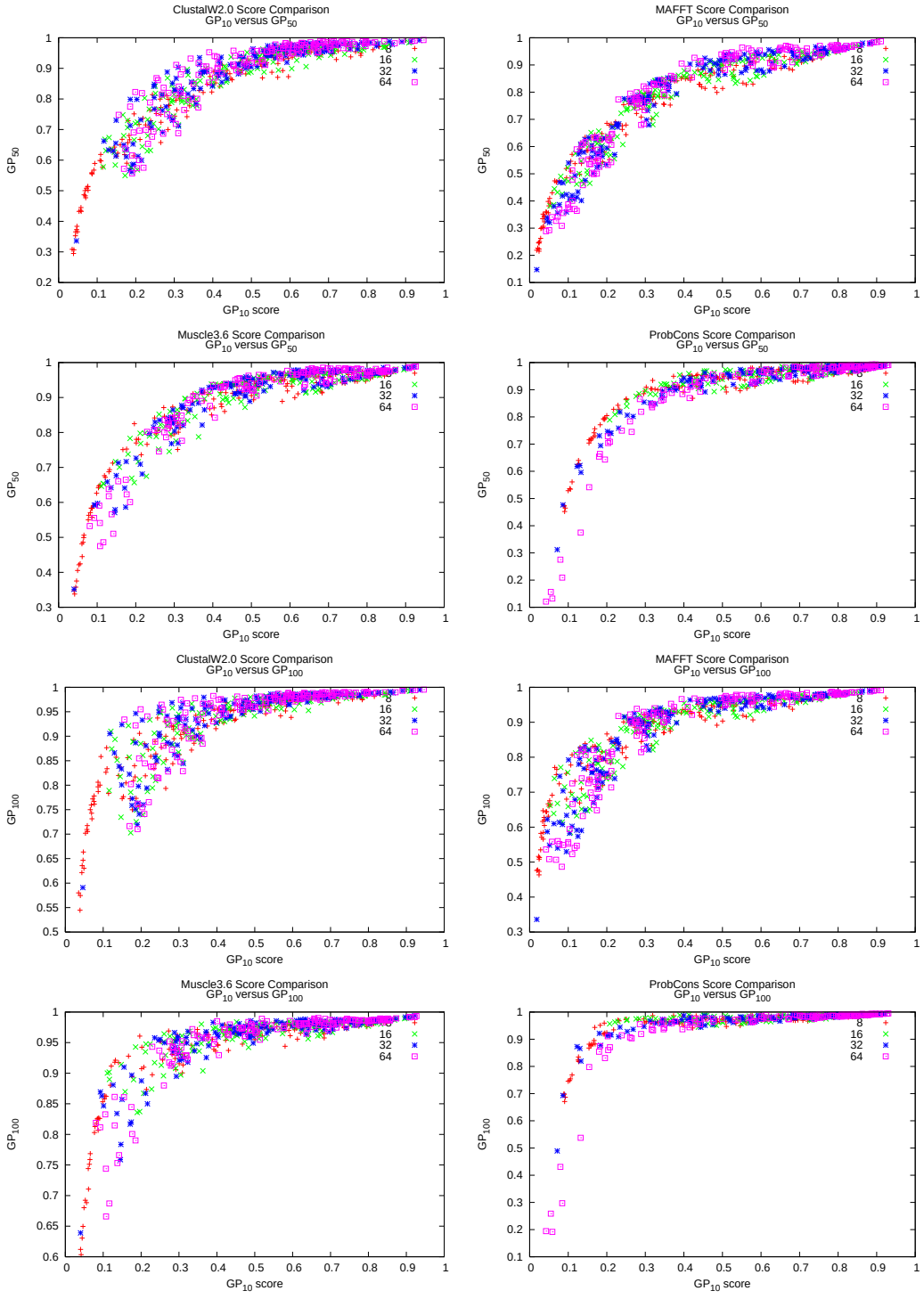

ClustalW2.0 Score Comparison
GP10 versus GP50
MAFFT Score Comparison
GP10 versus GP50
Muscle3.6 Score Comparison
GP10 versus GP50
ProbCons Score Comparison
GP10 versus GP50
ClustalW2.0 Score Comparison
GP10 versus GP100
MAFFT Score Comparison
GP10 versus GP100
Muscle3.6 Score Comparison
GP10 versus GP100
ProbCons Score Comparison
GP10 versus GP100
GP50
GP100
GP10 score
8
16
32
64

Figure B.2

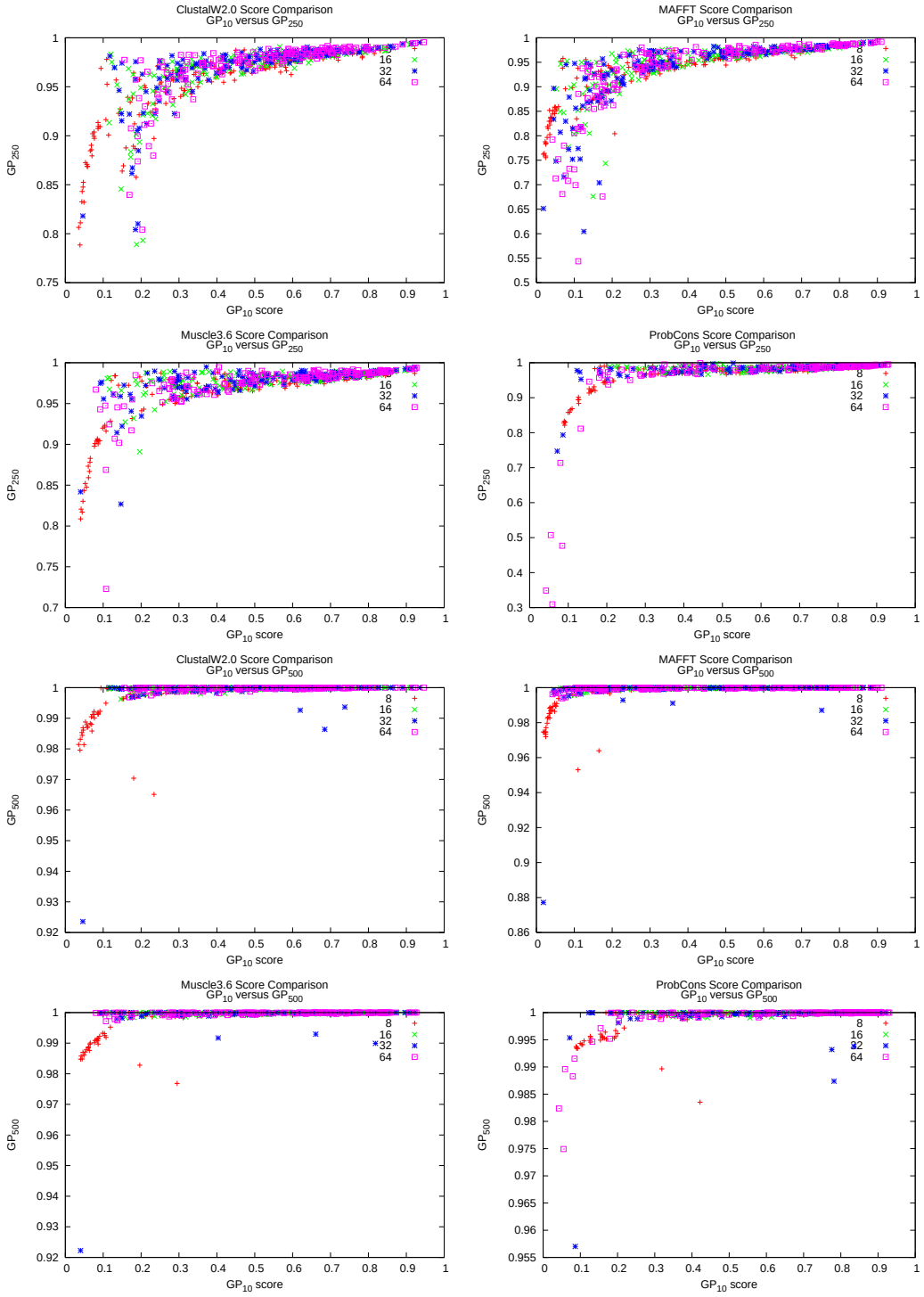

ClustalW2.0 Score Comparison
GP10 versus GP250
MAFFT Score Comparison
GP10 versus GP250
Muscle3.6 Score Comparison
GP10 versus GP250
ProbCons Score Comparison
GP10 versus GP250
ClustalW2.0 Score Comparison
GP10 versus GP500
MAFFT Score Comparison
GP10 versus GP500
Muscle3.6 Score Comparison
GP10 versus GP500
ProbCons Score Comparison
GP10 versus GP500
GP250
GP500
GP10 score
8
16
32
64

Figure B.2

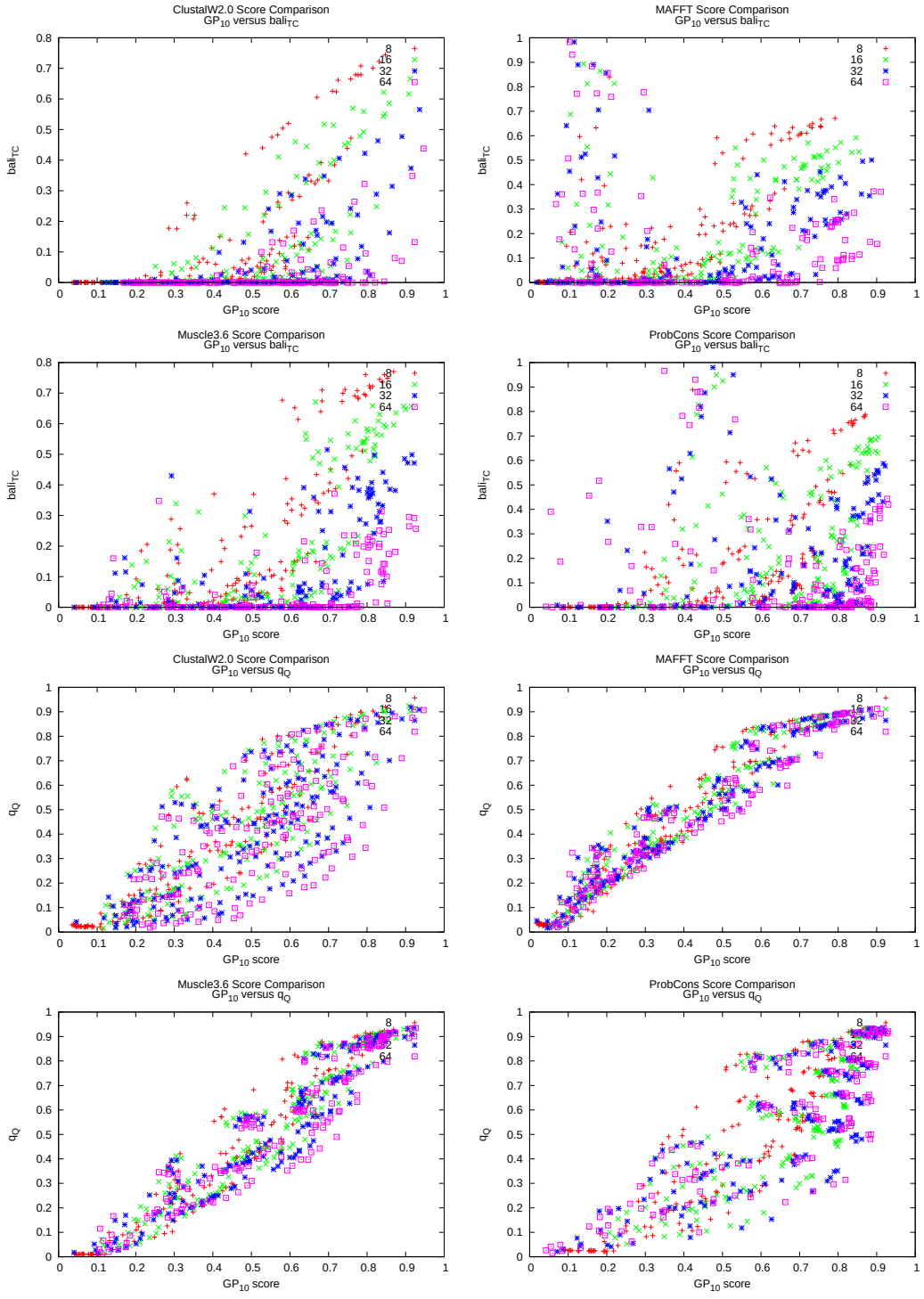

ClustalW2.0 Score Comparison
GP10 versus baliTC
MAFFT Score Comparison
GP10 versus baliTC
Muscle3.6 Score Comparison
GP10 versus baliTC
ProbCons Score Comparison
GP10 versus baliTC
ClustalW2.0 Score Comparison
GP10 versus qQ
MAFFT Score Comparison
GP10 versus qQ
Muscle3.6 Score Comparison
GP10 versus qQ
ProbCons Score Comparison
GP10 versus qQ
baliTC
qQ
GP10 score
8
16
32
64

Figure B.2

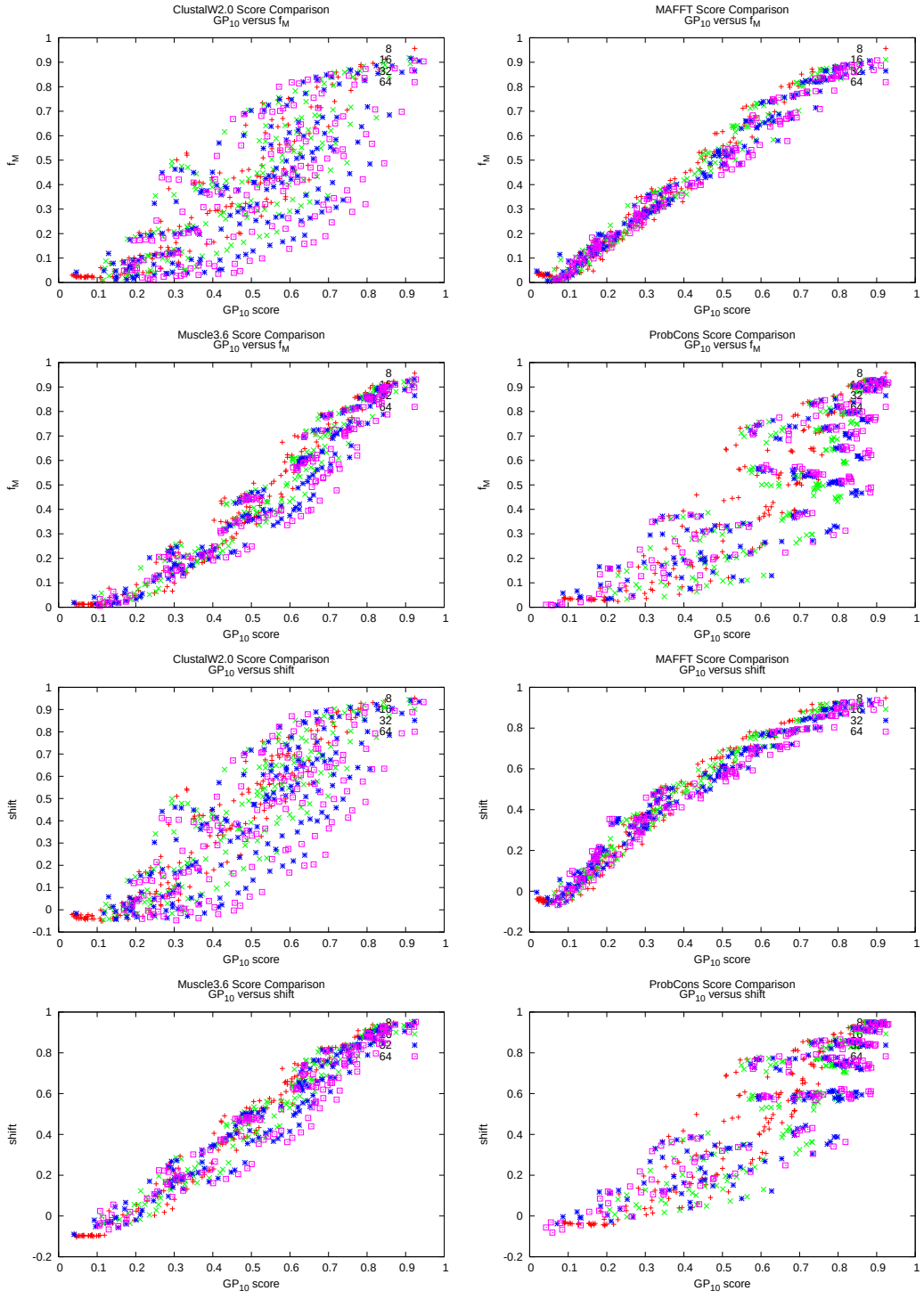

ClustalW2.0 Score Comparison
GP10 versus fM
MAFFT Score Comparison
GP10 versus fM
Muscle3.6 Score Comparison
GP10 versus fM
ProbCons Score Comparison
GP10 versus fM
ClustalW2.0 Score Comparison
GP10 versus shift
MAFFT Score Comparison
GP10 versus shift
Muscle3.6 Score Comparison
GP10 versus shift
ProbCons Score Comparison
GP10 versus shift
fM
shift
GP10 score
8
16
32
64

Figure B.2

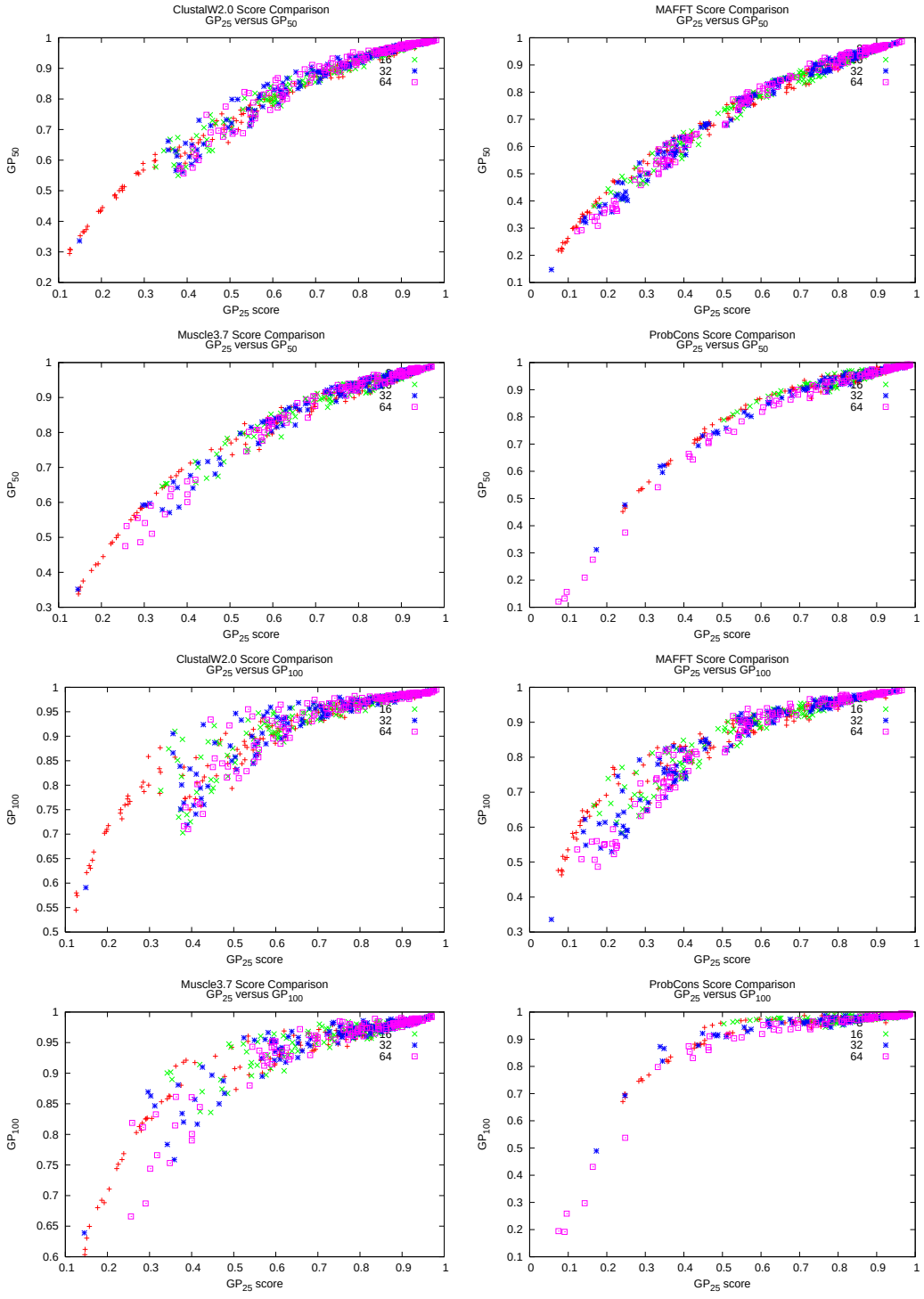

ClustalW2.0 Score Comparison
GP25 versus GP50
MAFFT Score Comparison
GP25 versus GP50
Muscle3.7 Score Comparison
GP25 versus GP50
ProbCons Score Comparison
GP25 versus GP50
ClustalW2.0 Score Comparison
GP25 versus GP100
MAFFT Score Comparison
GP25 versus GP100
Muscle3.7 Score Comparison
GP25 versus GP100
ProbCons Score Comparison
GP25 versus GP100
GP50
GP100
GP25 score
8
16
32
64

Figure B.2

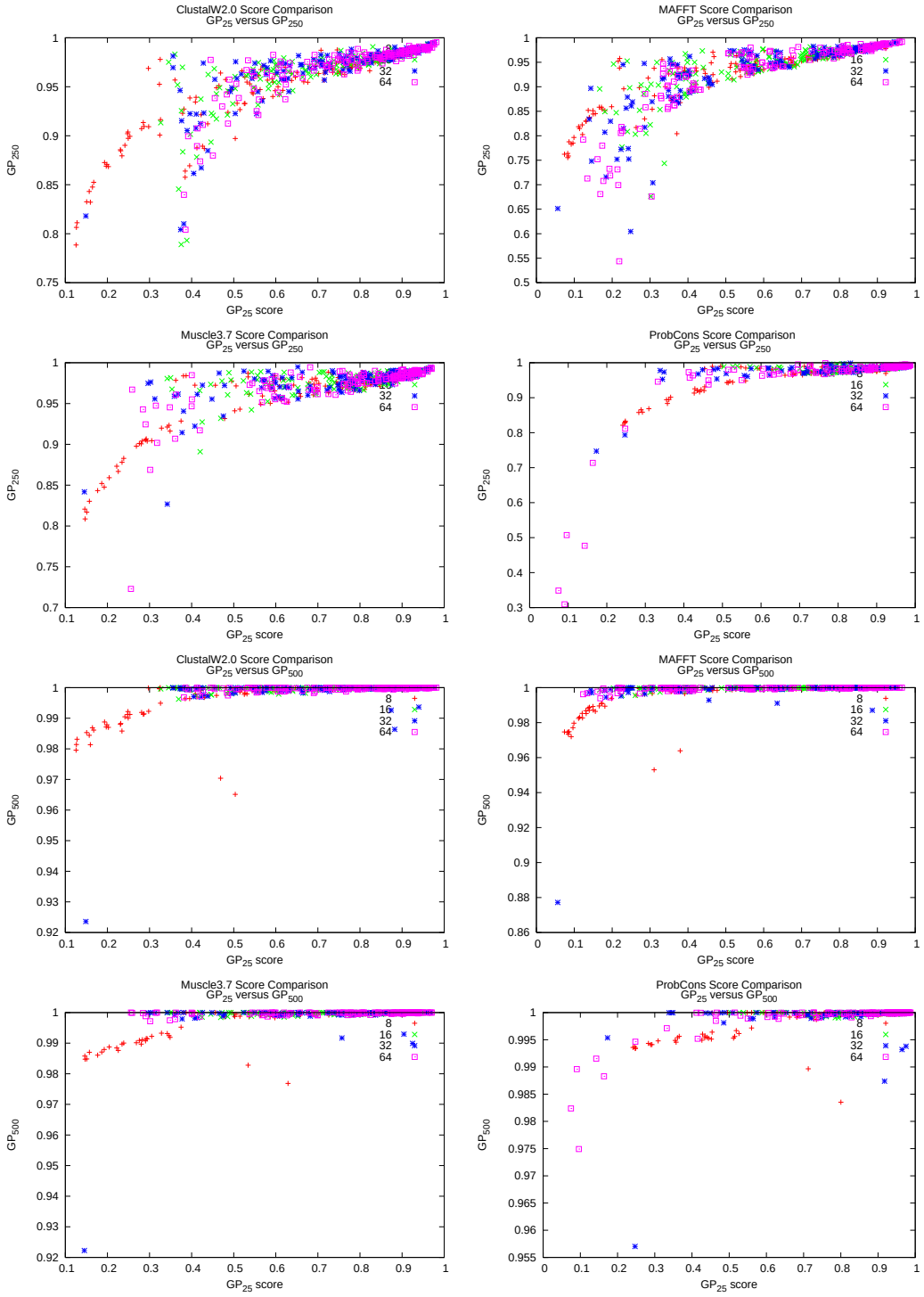

ClustalW2.0 Score Comparison
GP25 versus GP250
MAFFT Score Comparison
GP25 versus GP250
Muscle3.7 Score Comparison
GP25 versus GP250
ProbCons Score Comparison
GP25 versus GP250
ClustalW2.0 Score Comparison
GP25 versus GP500
MAFFT Score Comparison
GP25 versus GP500
Muscle3.7 Score Comparison
GP25 versus GP500
ProbCons Score Comparison
GP25 versus GP500
GP250
GP500
GP25 score
8
16
32
64

Figure B.2

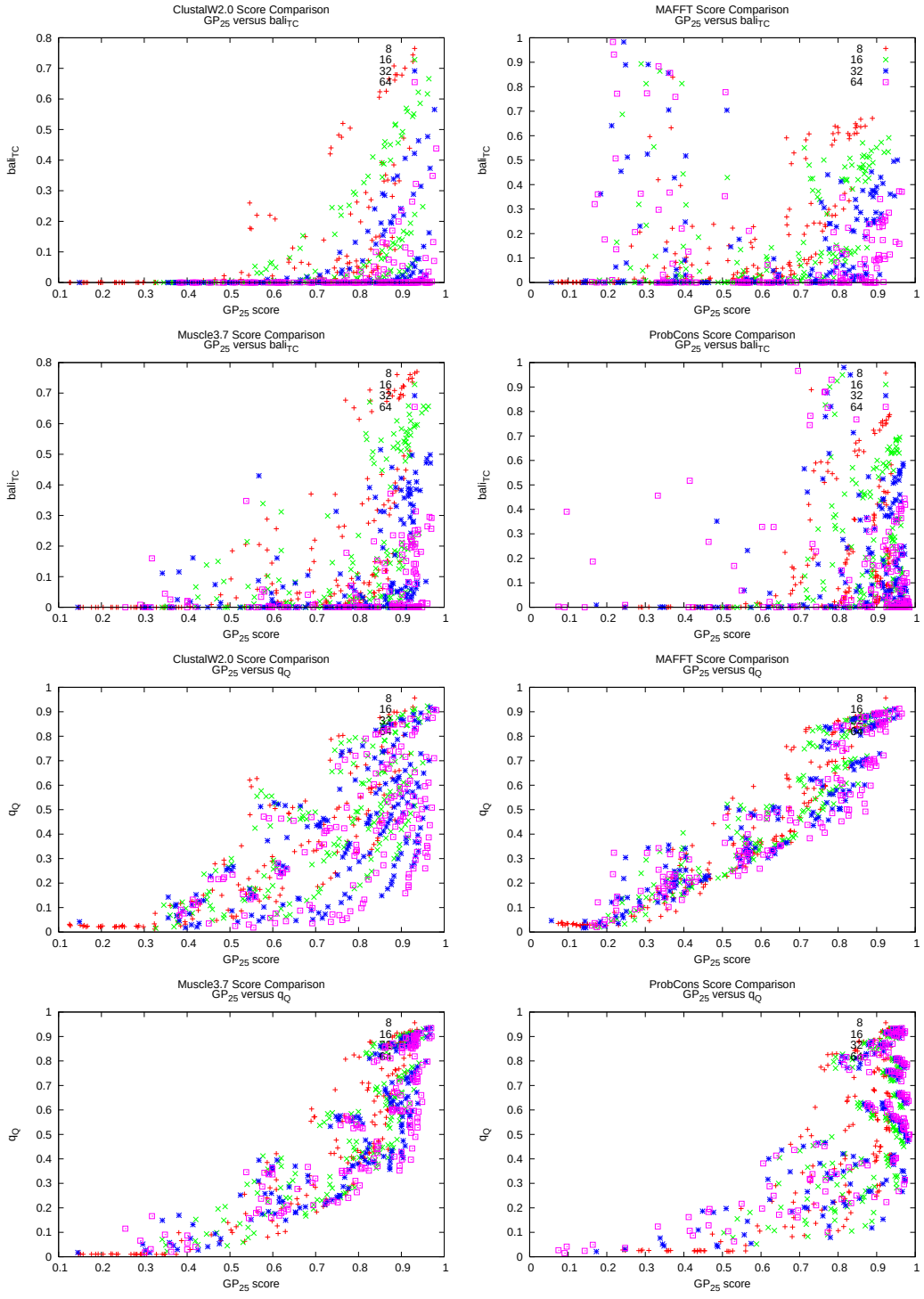

ClustalW2.0 Score Comparison
GP25 versus baliTC
MAFFT Score Comparison
GP25 versus baliTC
Muscle3.7 Score Comparison
GP25 versus baliTC
ProbCons Score Comparison
GP25 versus baliTC
ClustalW2.0 Score Comparison
GP25 versus qQ
MAFFT Score Comparison
GP25 versus qQ
Muscle3.7 Score Comparison
GP25 versus qQ
ProbCons Score Comparison
GP25 versus qQ
8
16
32
64
baliTC
qQ
GP25 score

Figure B.2

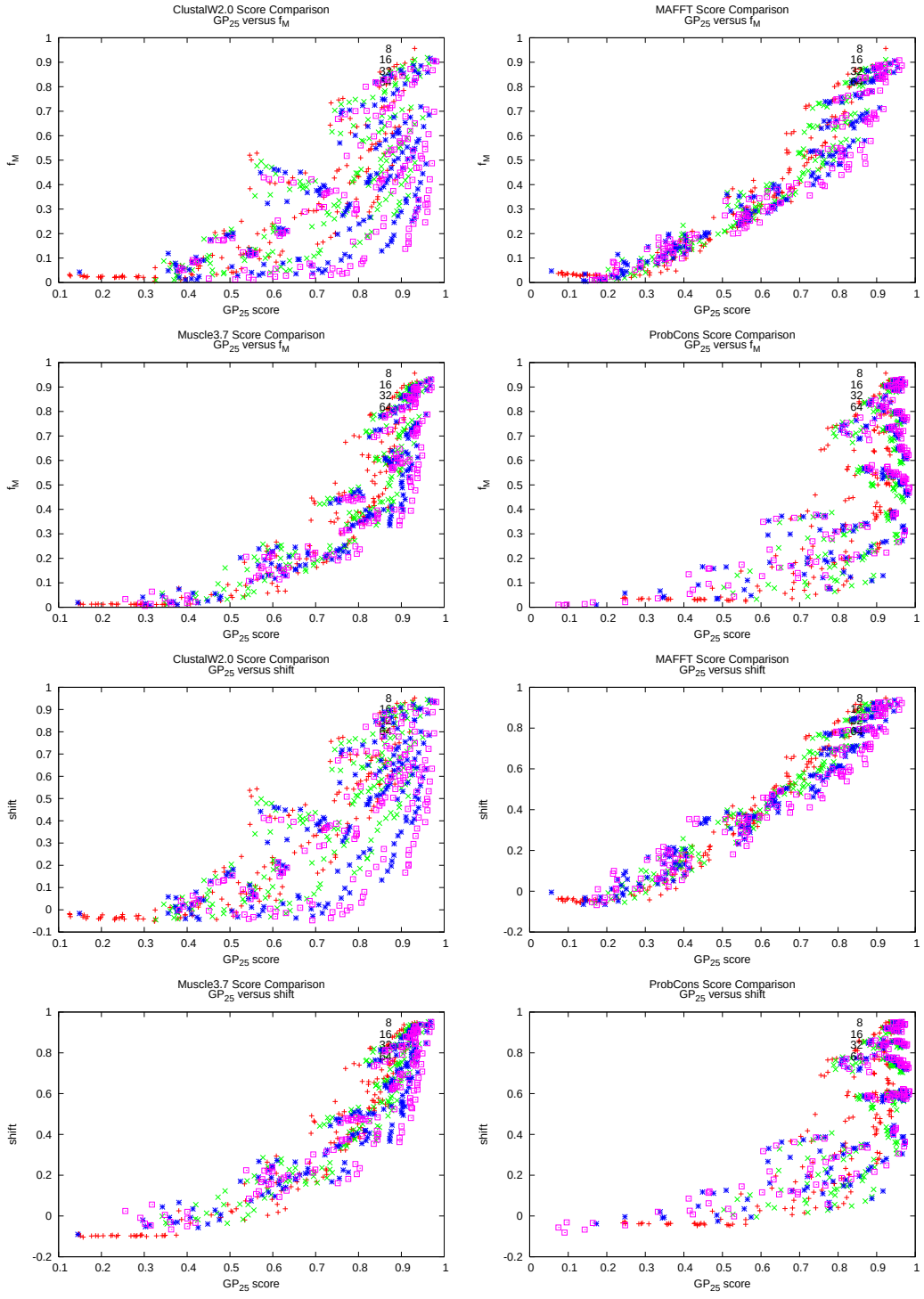

ClustalW2.0 Score Comparison
GP25 versus fM
MAFFT Score Comparison
GP25 versus fM
Muscle3.7 Score Comparison
GP25 versus fM
ProbCons Score Comparison
GP25 versus fM
ClustalW2.0 Score Comparison
GP25 versus shift
MAFFT Score Comparison
GP25 versus shift
Muscle3.7 Score Comparison
GP25 versus shift
ProbCons Score Comparison
GP25 versus shift
fM
shift
GP25 score
8
16
32
64

Figure B.2

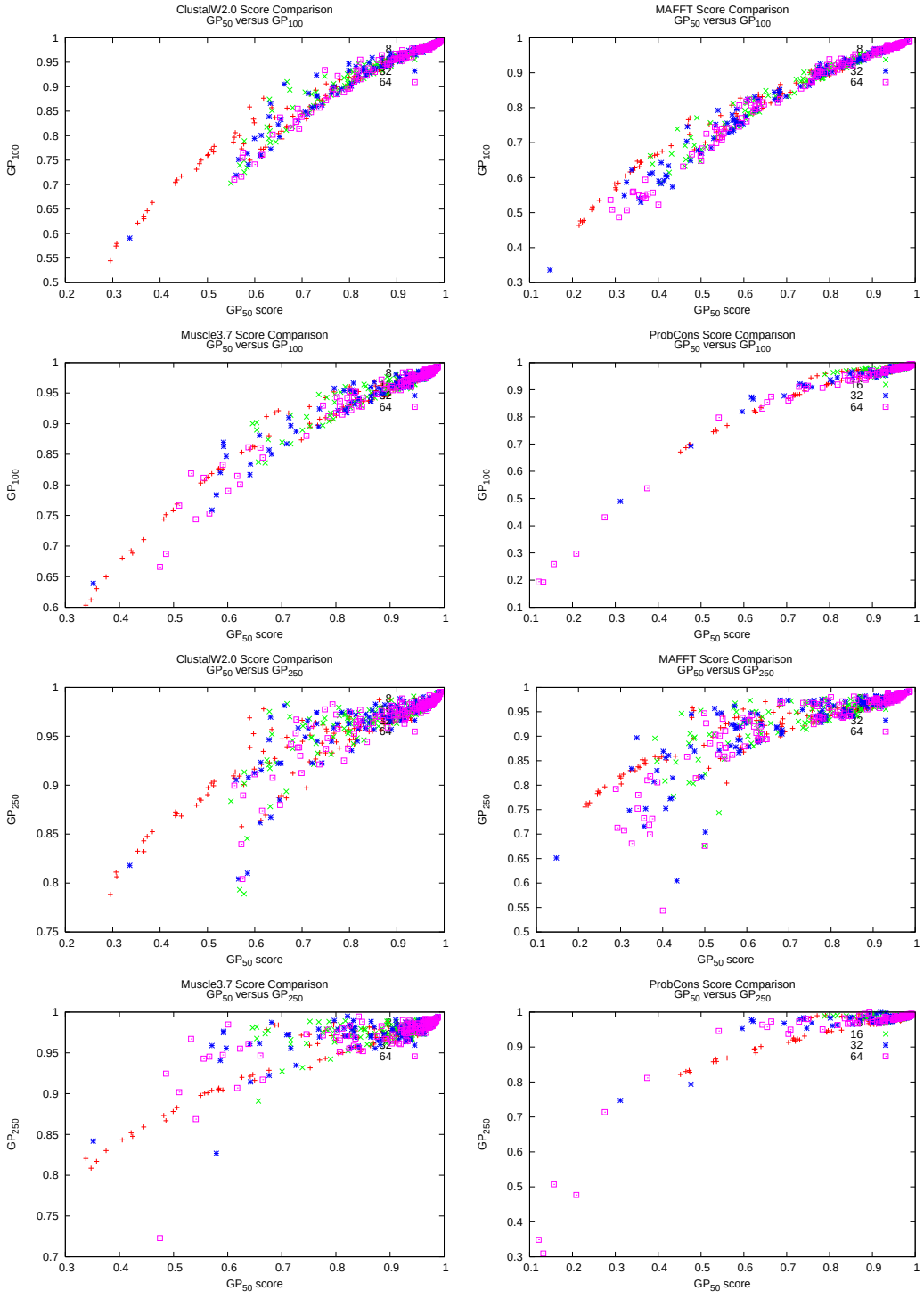

ClustalW2.0 Score Comparison
GP50 versus GP100
GP100
GP50 score
8
32
64
MAFFT Score Comparison
GP50 versus GP100
GP100
GP50 score
8
32
64
Muscle3.7 Score Comparison
GP50 versus GP100
GP100
GP50 score
8
32
64
ProbCons Score Comparison
GP50 versus GP100
GP100
GP50 score
16
32
64
ClustalW2.0 Score Comparison
GP50 versus GP250
GP250
GP50 score
8
32
64
MAFFT Score Comparison
GP50 versus GP250
GP250
GP50 score
32
64
Muscle3.7 Score Comparison
GP50 versus GP250
GP250
GP50 score
32
64
ProbCons Score Comparison
GP50 versus GP250
GP250
GP50 score
16
32
64

Figure B.2

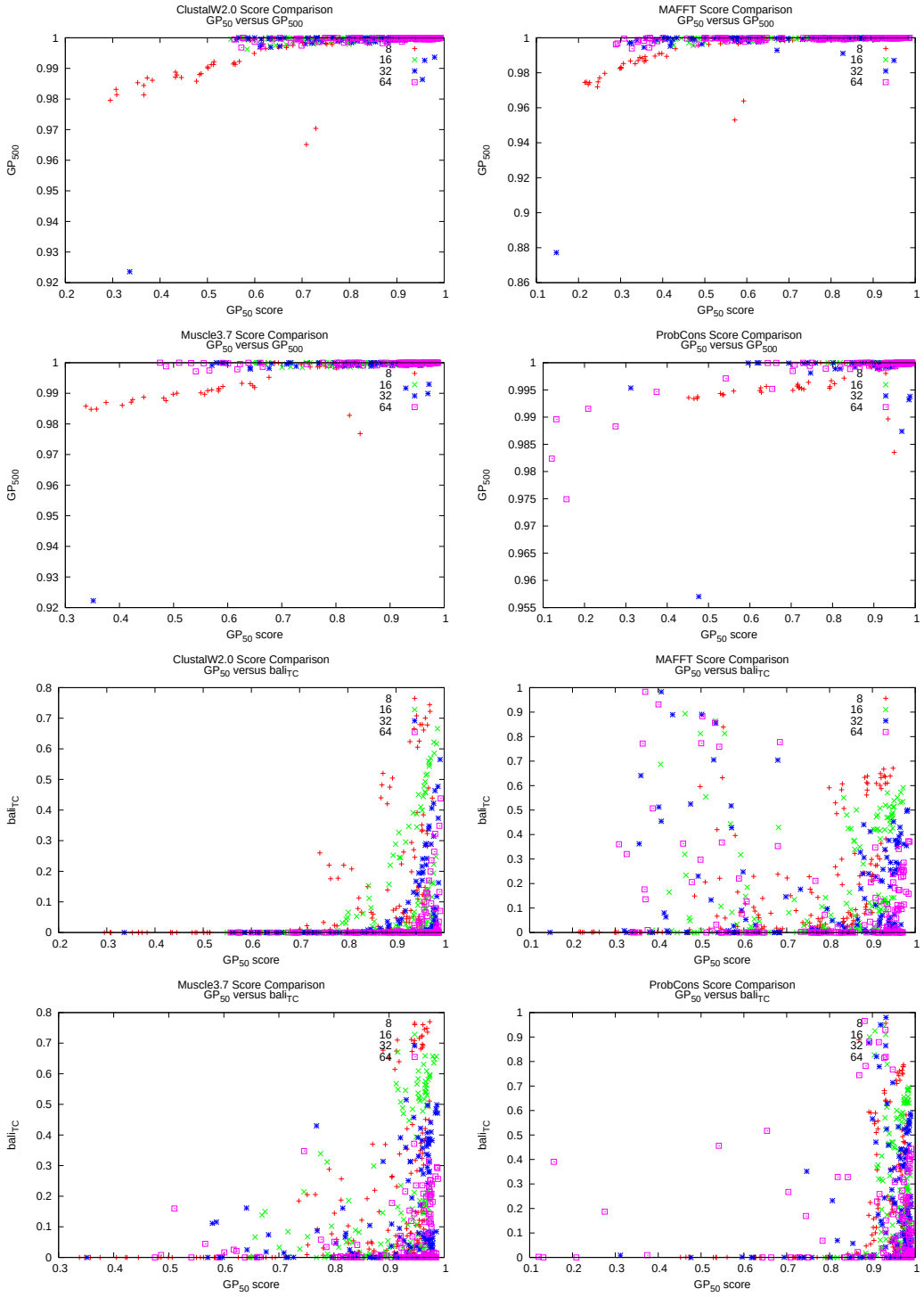

ClustalW2.0 Score Comparison
GP50 versus GP500
MAFFT Score Comparison
GP50 versus GP500
Muscle3.7 Score Comparison
GP50 versus GP500
ProbCons Score Comparison
GP50 versus GP500
ClustalW2.0 Score Comparison
GP50 versus baliTC
MAFFT Score Comparison
GP50 versus baliTC
Muscle3.7 Score Comparison
GP50 versus baliTC
ProbCons Score Comparison
GP50 versus baliTC
GP500
baliTC
GP50 score
8
16
32
64

Figure B.2

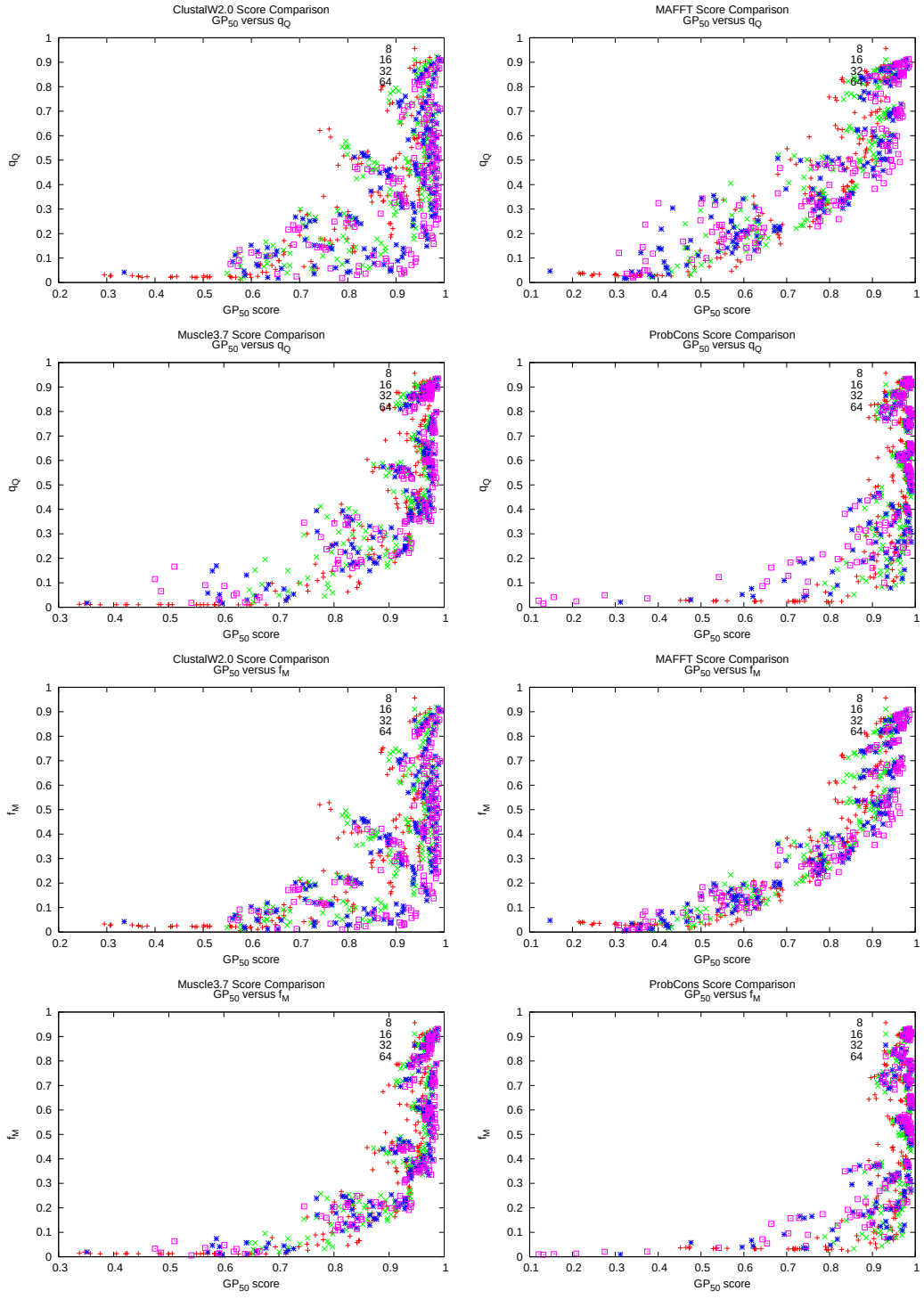

ClustalW2.0 Score Comparison
GP50 versus qQ
MAFFT Score Comparison
GP50 versus qQ
Muscle3.7 Score Comparison
GP50 versus qQ
ProbCons Score Comparison
GP50 versus qQ
ClustalW2.0 Score Comparison
GP50 versus fM
MAFFT Score Comparison
GP50 versus fM
Muscle3.7 Score Comparison
GP50 versus fM
ProbCons Score Comparison
GP50 versus fM
8
16
32
64
qQ
fM
GP50 score

Figure B.2

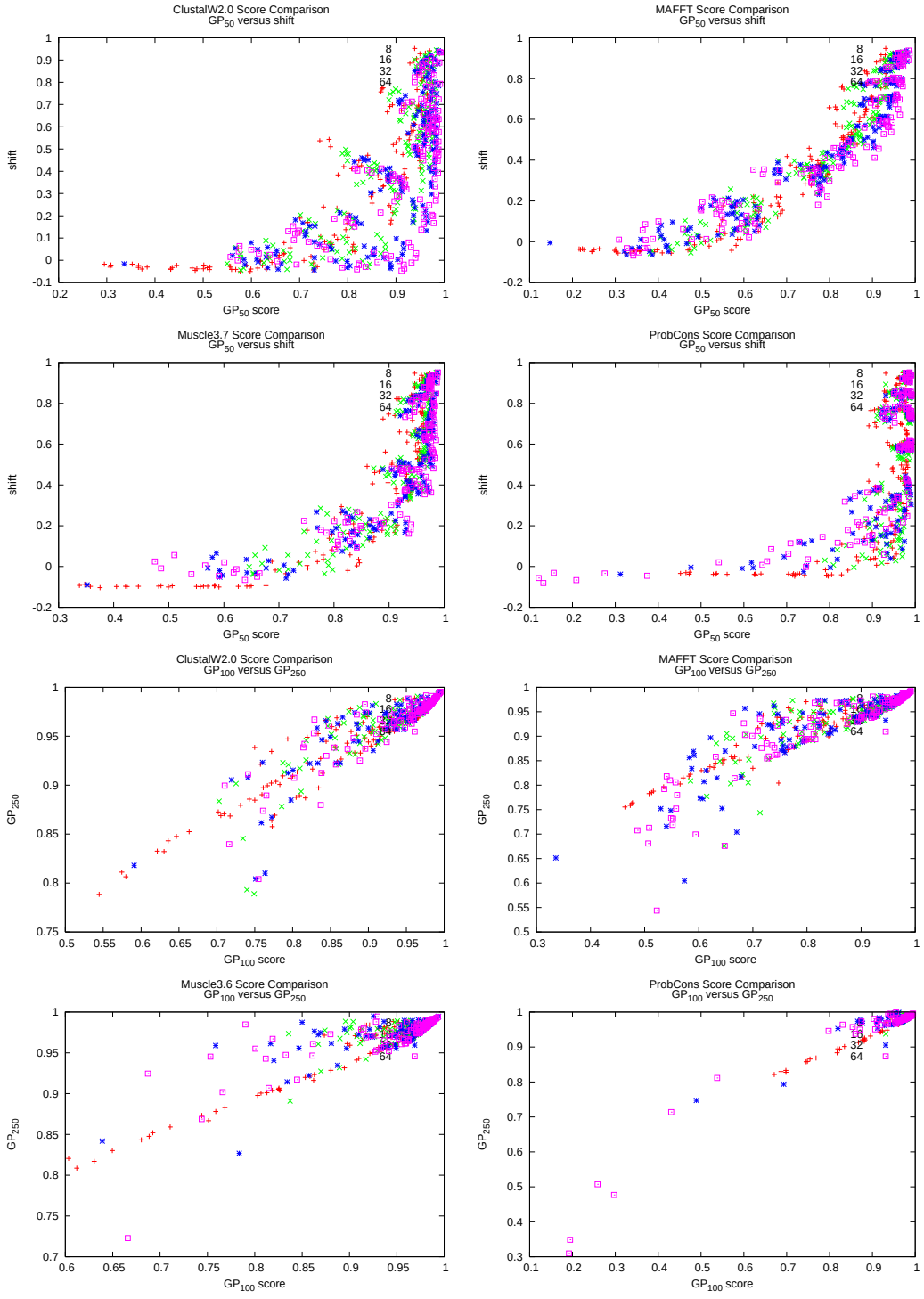

ClustalW2.0 Score Comparison
GP50 versus shift
MAFFT Score Comparison
GP50 versus shift
Muscle3.7 Score Comparison
GP50 versus shift
ProbCons Score Comparison
GP50 versus shift
ClustalW2.0 Score Comparison
GP100 versus GP250
MAFFT Score Comparison
GP100 versus GP250
Muscle3.6 Score Comparison
GP100 versus GP250
ProbCons Score Comparison
GP100 versus GP250
shift
GP50 score
GP250
GP100 score
8
16
32
64

Figure B.2

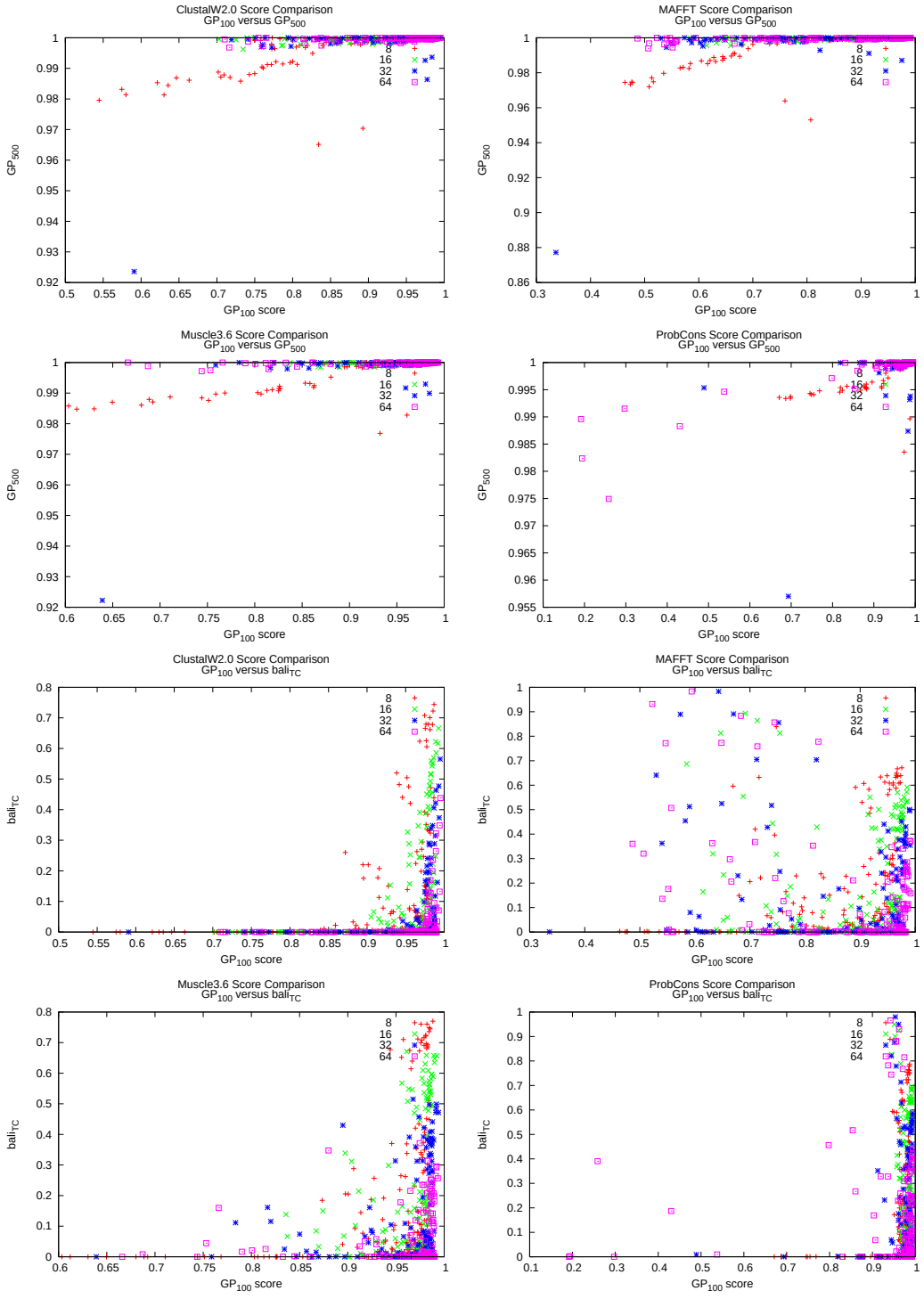

ClustalW2.0 Score Comparison
GP100 versus GP500
MAFFT Score Comparison
GP100 versus GP500
Muscle3.6 Score Comparison
GP100 versus GP500
ProbCons Score Comparison
GP100 versus GP500
ClustalW2.0 Score Comparison
GP100 versus baliTC
MAFFT Score Comparison
GP100 versus baliTC
Muscle3.6 Score Comparison
GP100 versus baliTC
ProbCons Score Comparison
GP100 versus baliTC
GP500
baliTC
GP100 score
8
16
32
64

Figure B.2

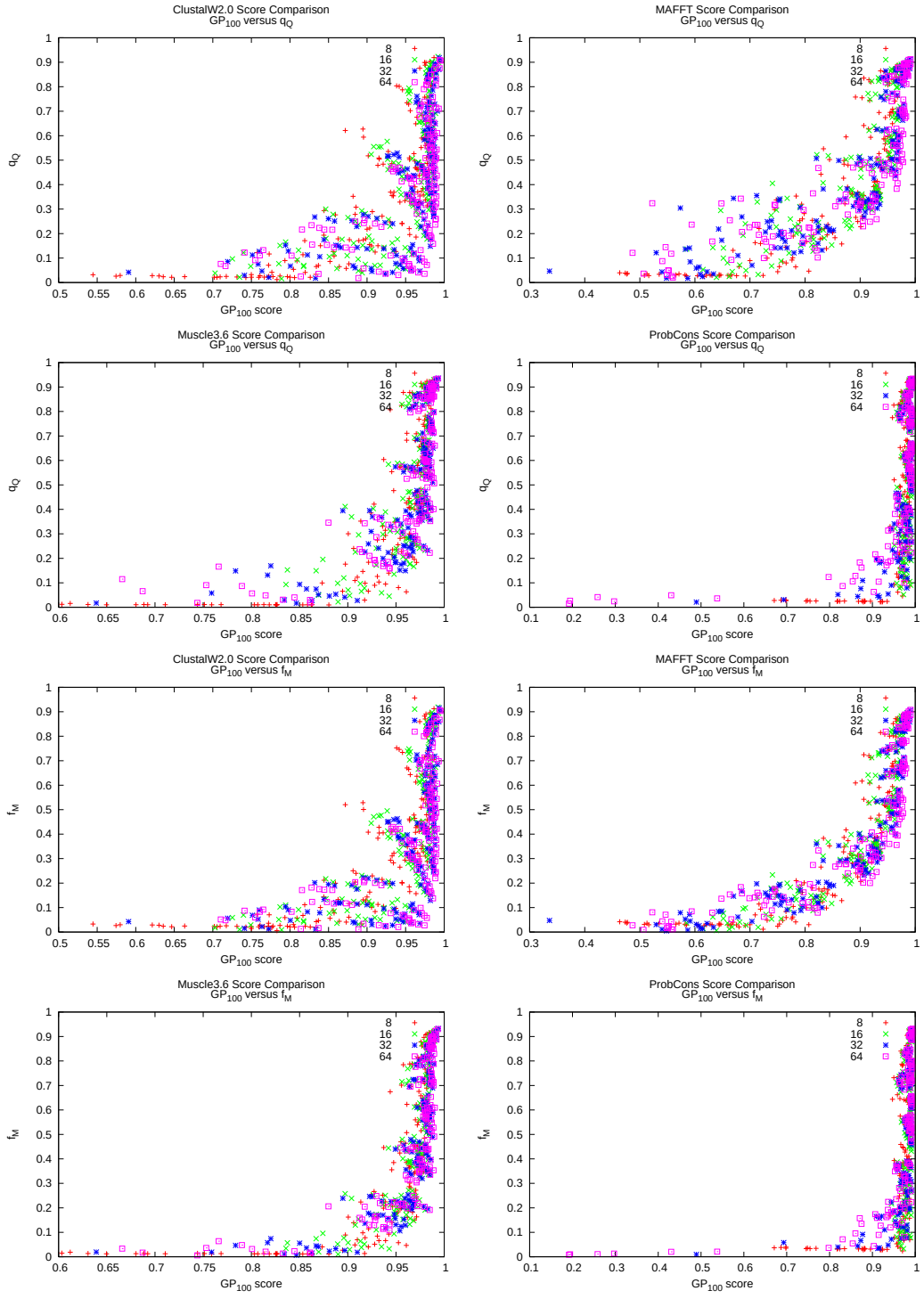

ClustalW2.0 Score Comparison
GP100 versus qQ
MAFFT Score Comparison
GP100 versus qQ
Muscle3.6 Score Comparison
GP100 versus qQ
ProbCons Score Comparison
GP100 versus qQ
ClustalW2.0 Score Comparison
GP100 versus fM
MAFFT Score Comparison
GP100 versus fM
Muscle3.6 Score Comparison
GP100 versus fM
ProbCons Score Comparison
GP100 versus fM
8
16
32
64
qQ
fM
GP100 score

Figure B.2

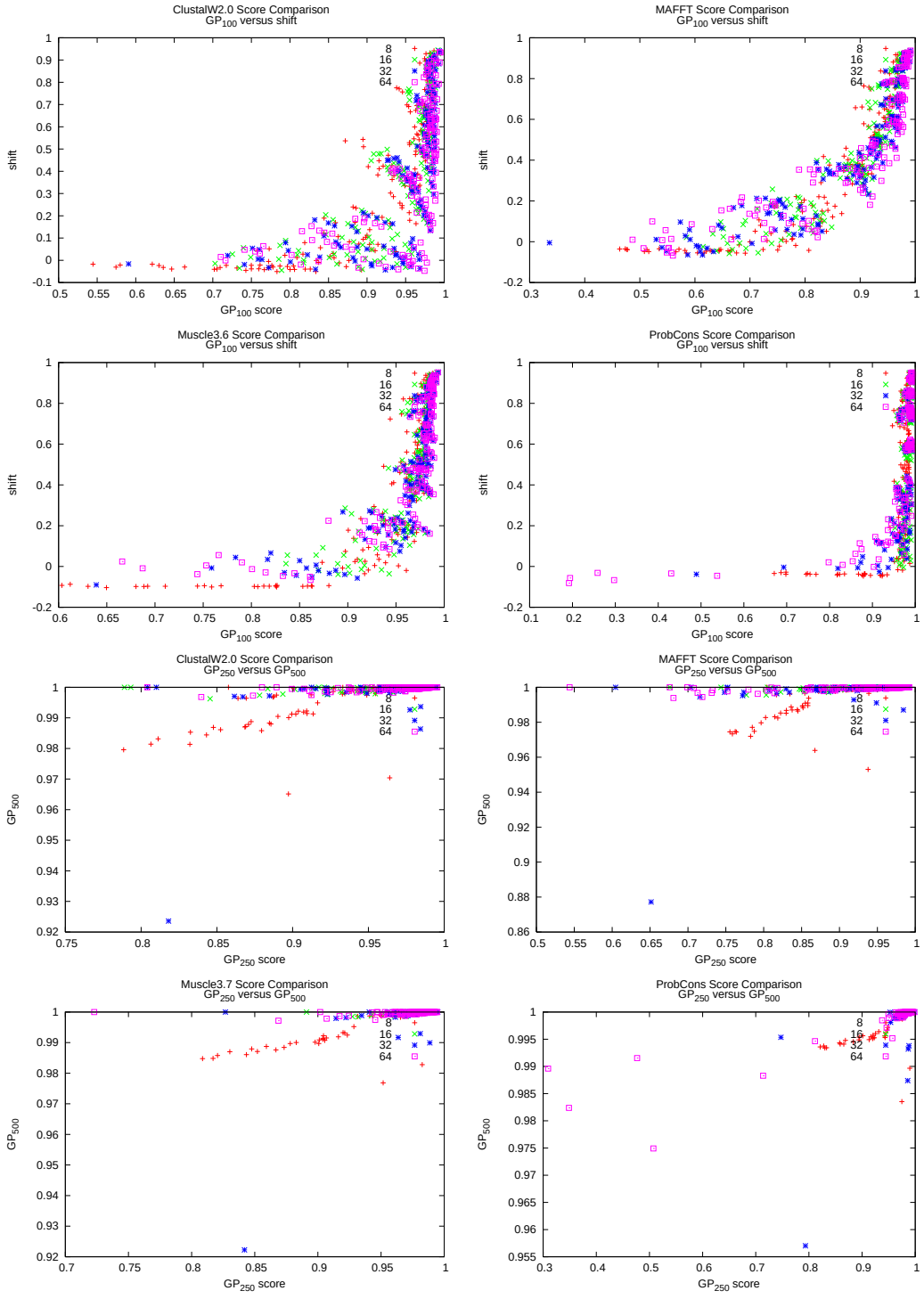

ClustalW2.0 Score Comparison
GP100 versus shift
MAFFT Score Comparison
GP100 versus shift
Muscle3.6 Score Comparison
GP100 versus shift
ProbCons Score Comparison
GP100 versus shift
ClustalW2.0 Score Comparison
GP250 versus GP500
MAFFT Score Comparison
GP250 versus GP500
Muscle3.7 Score Comparison
GP250 versus GP500
ProbCons Score Comparison
GP250 versus GP500
shift
GP100 score
GP500
GP250 score
8
16
32
64

Figure B.2

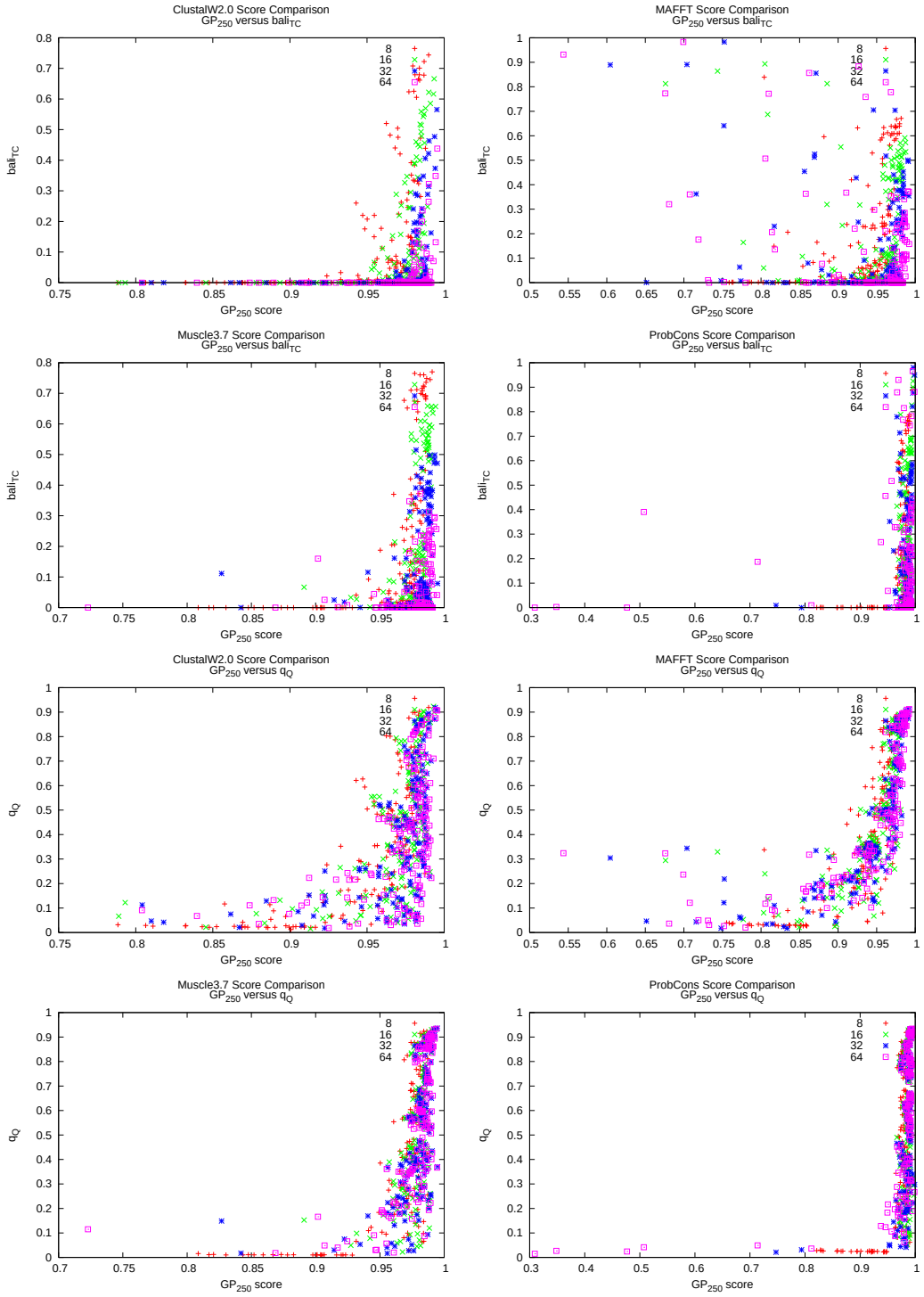

ClustalW2.0 Score Comparison
GP250 versus baliTC
MAFFT Score Comparison
GP250 versus baliTC
Muscle3.7 Score Comparison
GP250 versus baliTC
ProbCons Score Comparison
GP250 versus baliTC
ClustalW2.0 Score Comparison
GP250 versus qQ
MAFFT Score Comparison
GP250 versus qQ
Muscle3.7 Score Comparison
GP250 versus qQ
ProbCons Score Comparison
GP250 versus qQ
8
16
32
64
baliTC
qQ
GP250 score

Figure B.2

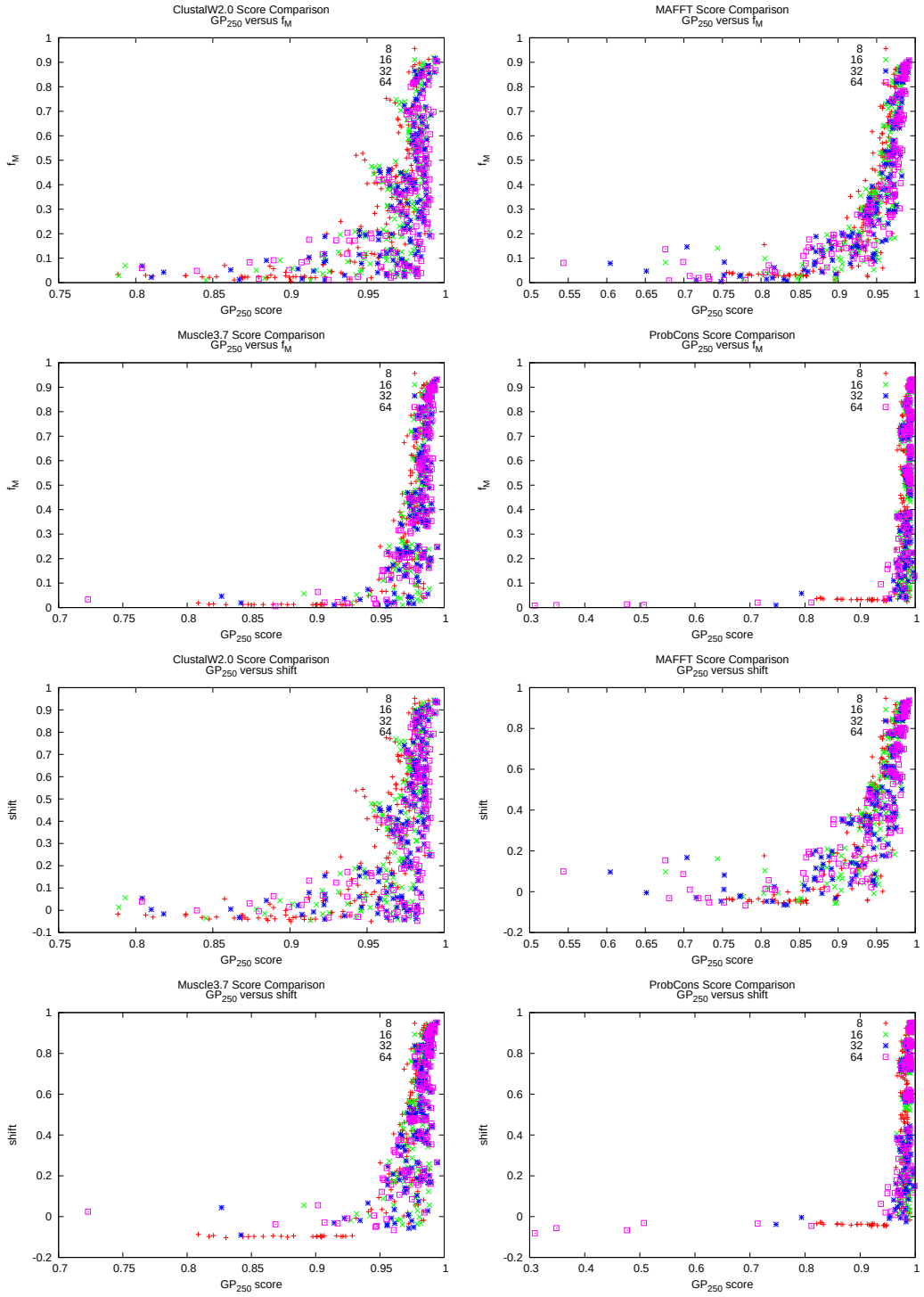

ClustalW2.0 Score Comparison
GP250 versus fM
MAFFT Score Comparison
GP250 versus fM
Muscle3.7 Score Comparison
GP250 versus fM
ProbCons Score Comparison
GP250 versus fM
ClustalW2.0 Score Comparison
GP250 versus shift
MAFFT Score Comparison
GP250 versus shift
Muscle3.7 Score Comparison
GP250 versus shift
ProbCons Score Comparison
GP250 versus shift
8
16
32
64
fM
shift
GP250 score

Figure B.2

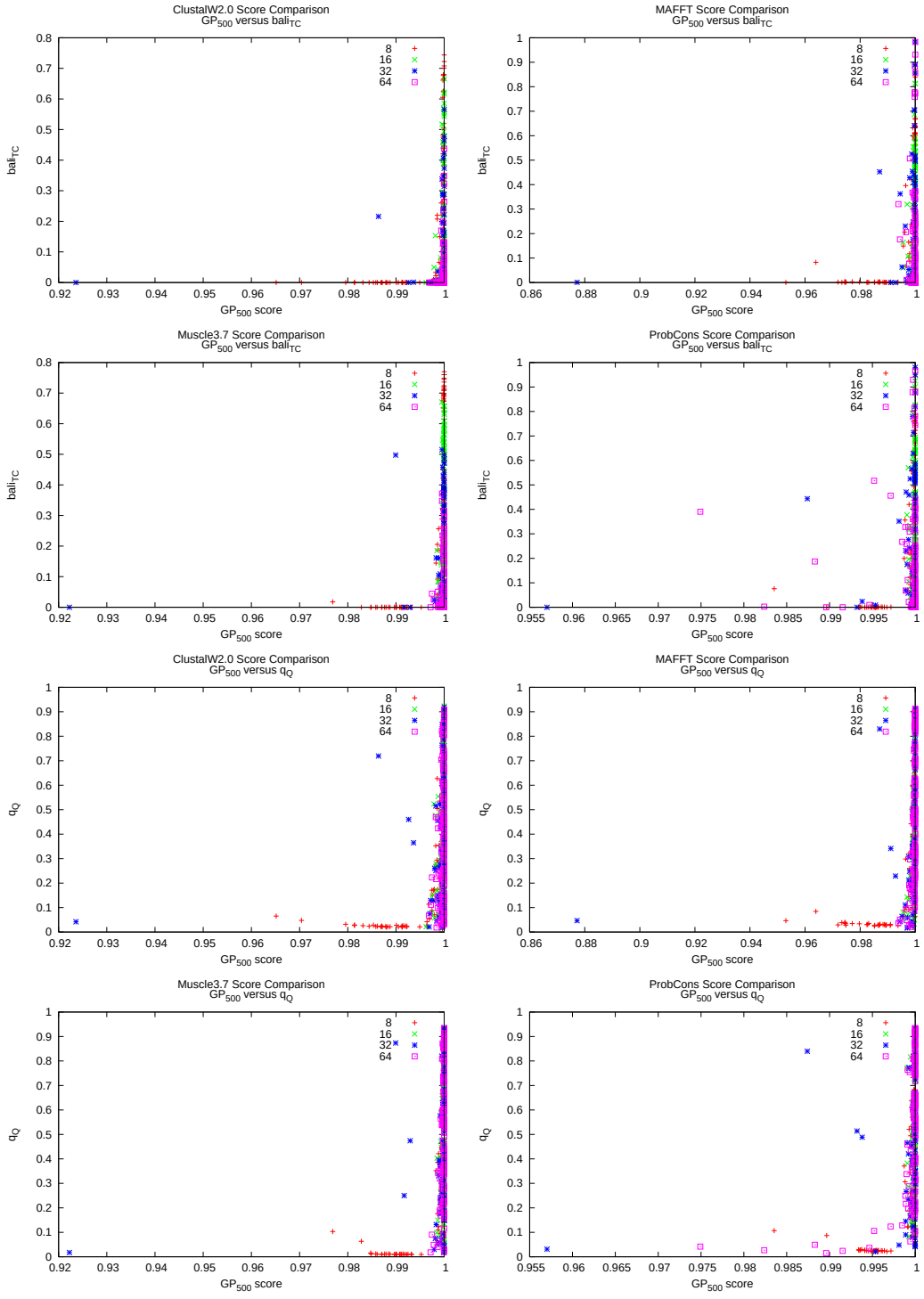

ClustalW2.0 Score Comparison
GP500 versus baliTC
MAFFT Score Comparison
GP500 versus baliTC
Muscle3.7 Score Comparison
GP500 versus baliTC
ProbCons Score Comparison
GP500 versus baliTC
ClustalW2.0 Score Comparison
GP500 versus qQ
MAFFT Score Comparison
GP500 versus qQ
Muscle3.7 Score Comparison
GP500 versus qQ
ProbCons Score Comparison
GP500 versus qQ
8
16
32
64
baliTC
qQ
GP500 score

Figure B.2

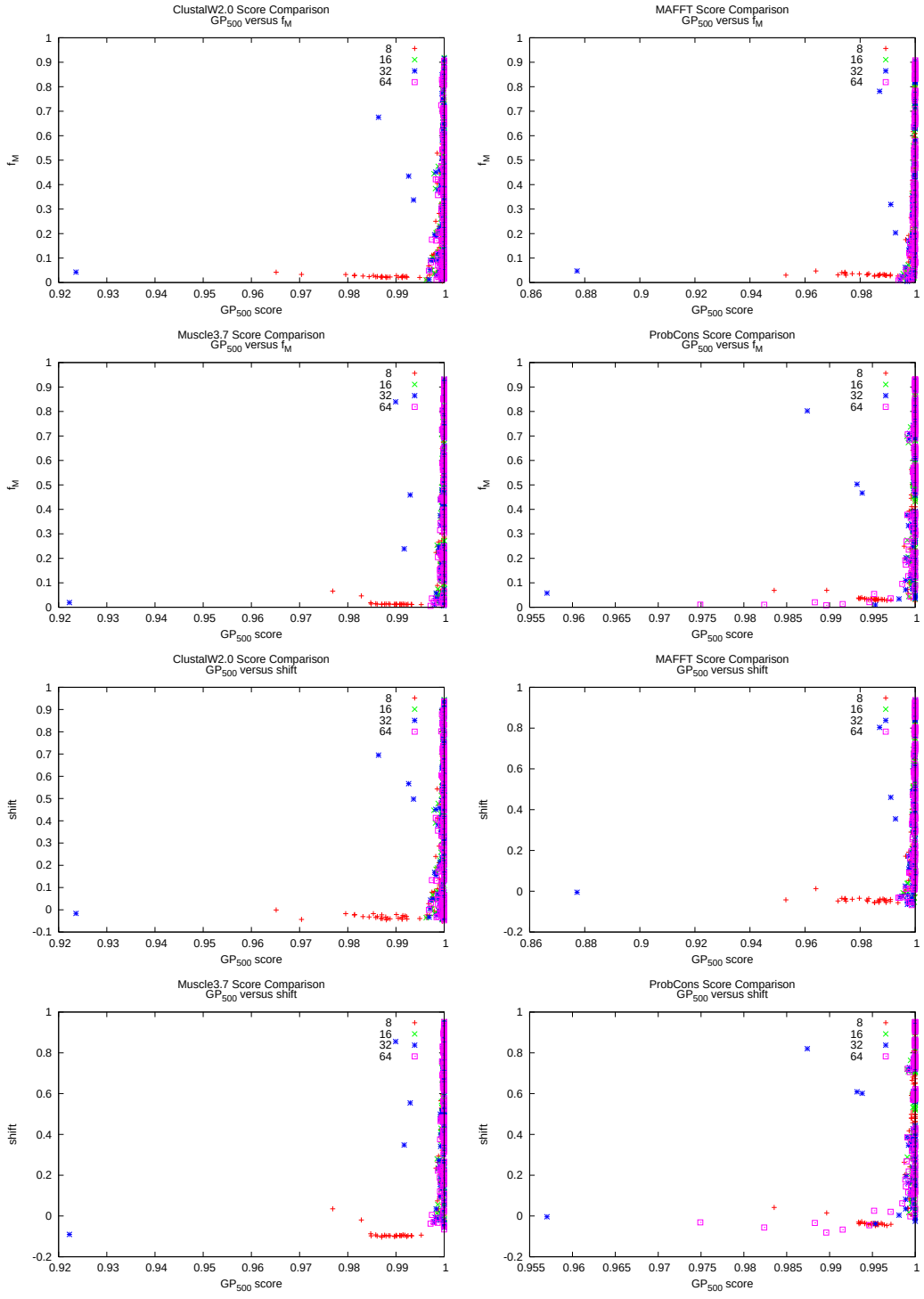

ClustalW2.0 Score Comparison
GP500 versus fM
MAFFT Score Comparison
GP500 versus fM
Muscle3.7 Score Comparison
GP500 versus fM
ProbCons Score Comparison
GP500 versus fM
ClustalW2.0 Score Comparison
GP500 versus shift
MAFFT Score Comparison
GP500 versus shift
Muscle3.7 Score Comparison
GP500 versus shift
ProbCons Score Comparison
GP500 versus shift
8
16
32
64
fM
shift
GP500 score

Figure B.2

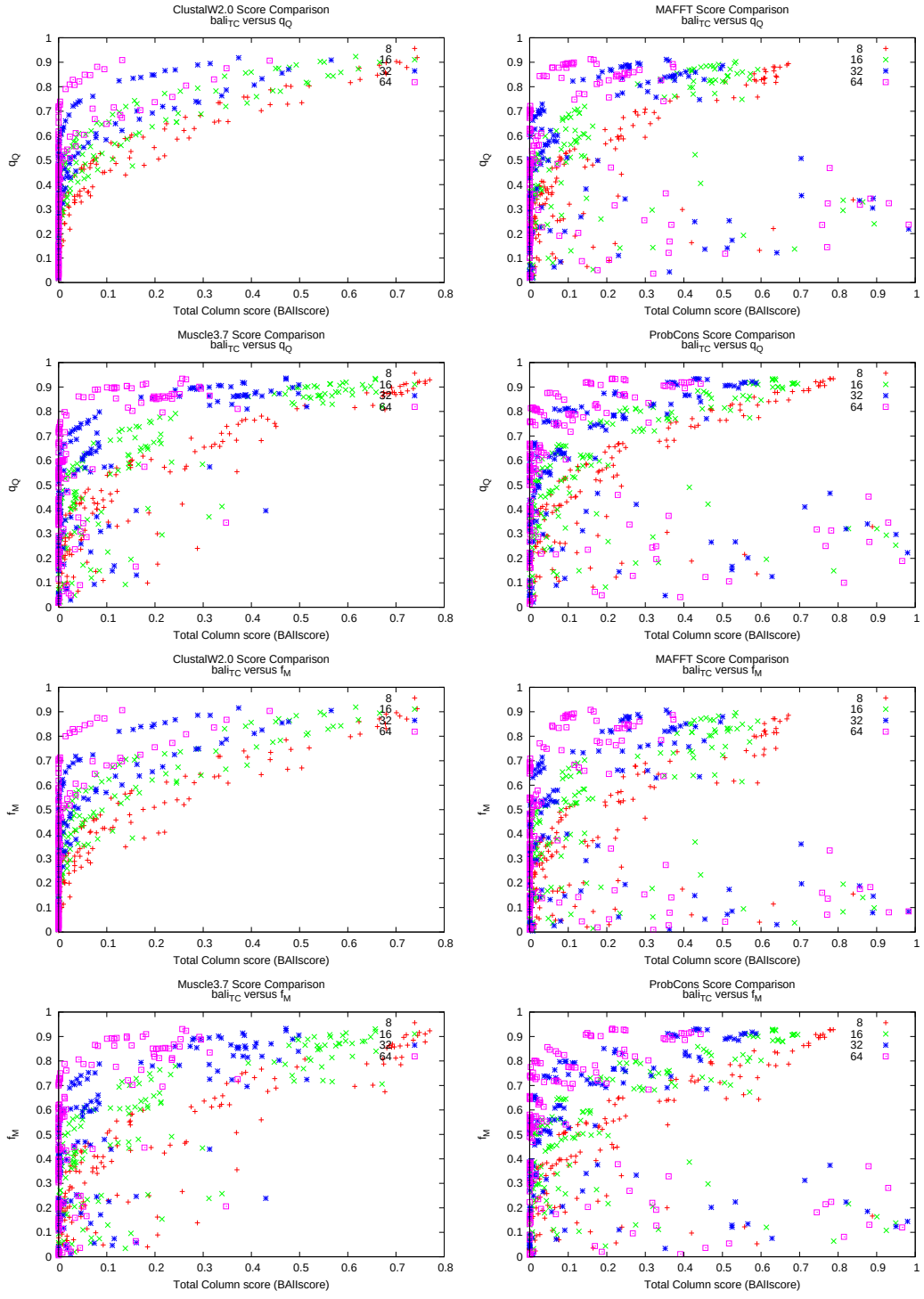

ClustalW2.0 Score Comparison
bali_TC versus q_Q
MAFFT Score Comparison
bali_TC versus q_Q
Muscle3.7 Score Comparison
bali_TC versus q_Q
ProbCons Score Comparison
bali_TC versus q_Q
ClustalW2.0 Score Comparison
bali_TC versus f_M
MAFFT Score Comparison
bali_TC versus f_M
Muscle3.7 Score Comparison
bali_TC versus f_M
ProbCons Score Comparison
bali_TC versus f_M
q_Q
f_M
Total Column score (BAIIscore)
8
16
32
64

Figure B.2

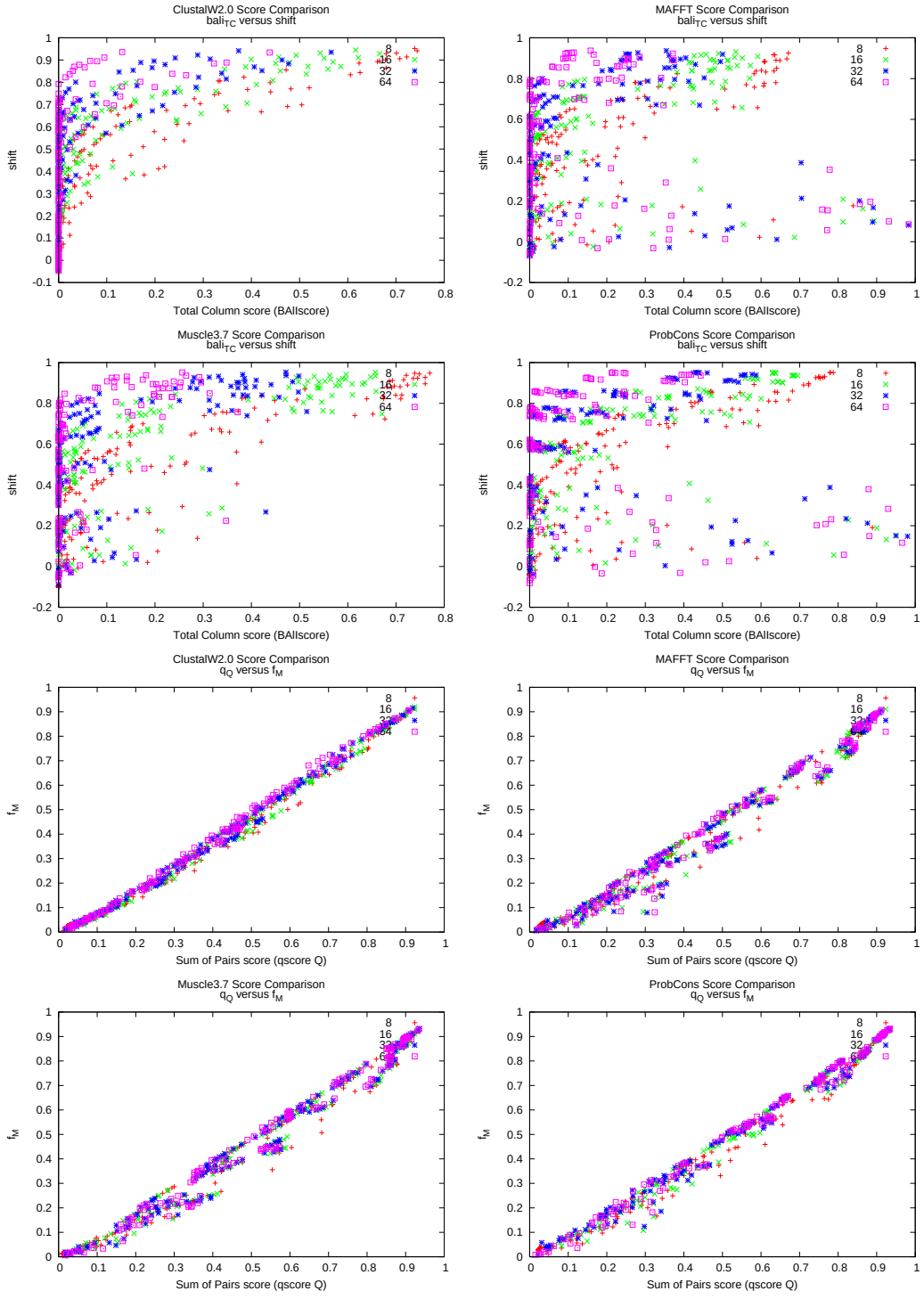

ClustalW2.0 Score Comparison
bali_TC versus shift
MAFFT Score Comparison
bali_TC versus shift
Muscle3.7 Score Comparison
bali_TC versus shift
ProbCons Score Comparison
bali_TC versus shift
ClustalW2.0 Score Comparison
q_Q versus f_M
MAFFT Score Comparison
q_Q versus f_M
Muscle3.7 Score Comparison
q_Q versus f_M
ProbCons Score Comparison
q_Q versus f_M
shift
f_M
Total Column score (BAIlscore)
Sum of Pairs score (qscore Q)
8
16
32
64

Figure B.2

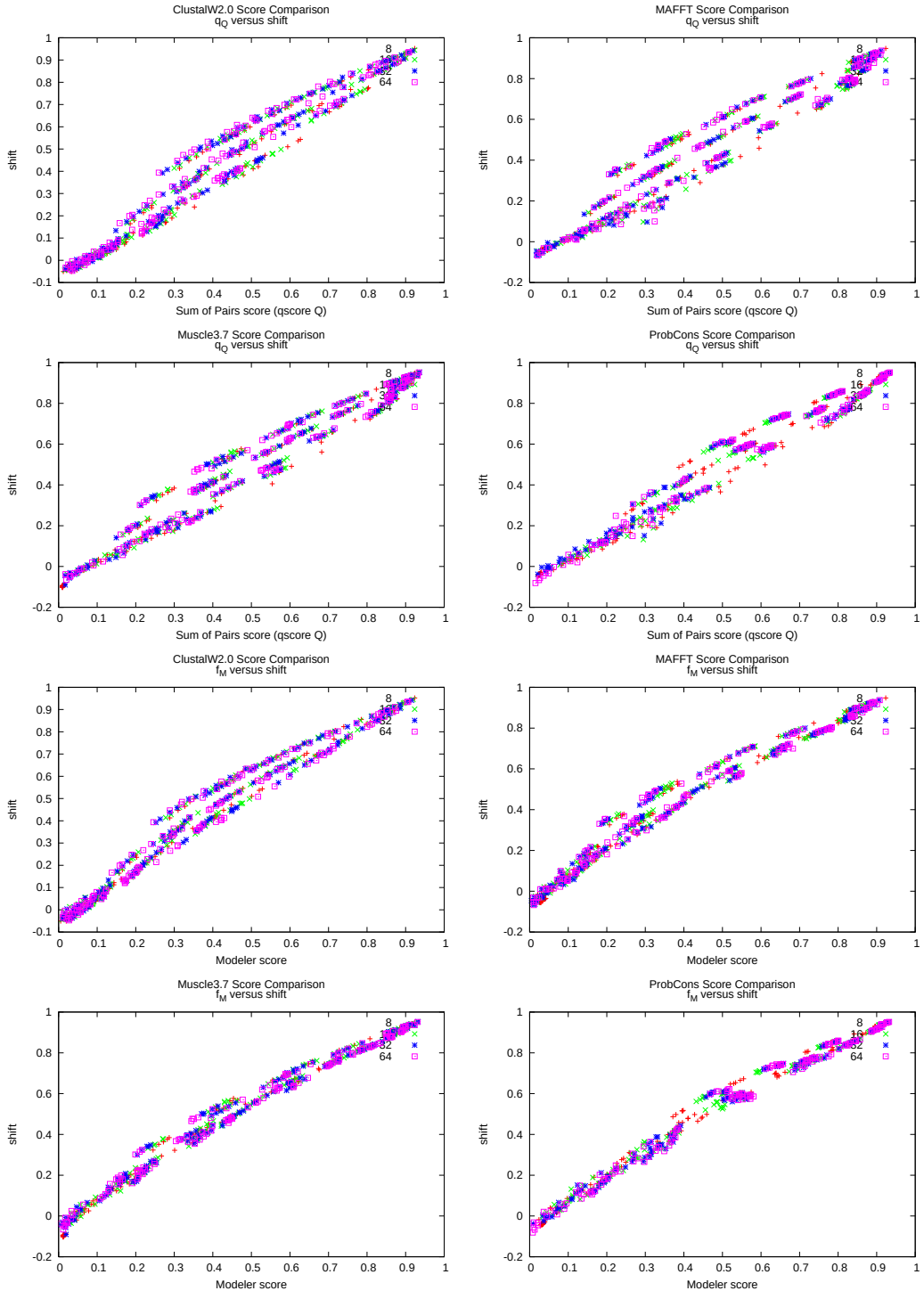

ClustalW2.0 Score Comparison
$q_Q$ versus shift
MAFFT Score Comparison
$q_Q$ versus shift
Muscle3.7 Score Comparison
$q_Q$ versus shift
ProbCons Score Comparison
$q_Q$ versus shift
ClustalW2.0 Score Comparison
$f_M$ versus shift
MAFFT Score Comparison
$f_M$ versus shift
Muscle3.7 Score Comparison
$f_M$ versus shift
ProbCons Score Comparison
$f_M$ versus shift
shift
Sum of Pairs score (qscore Q)
Modeler score
8
16
32
64

Figure B.2

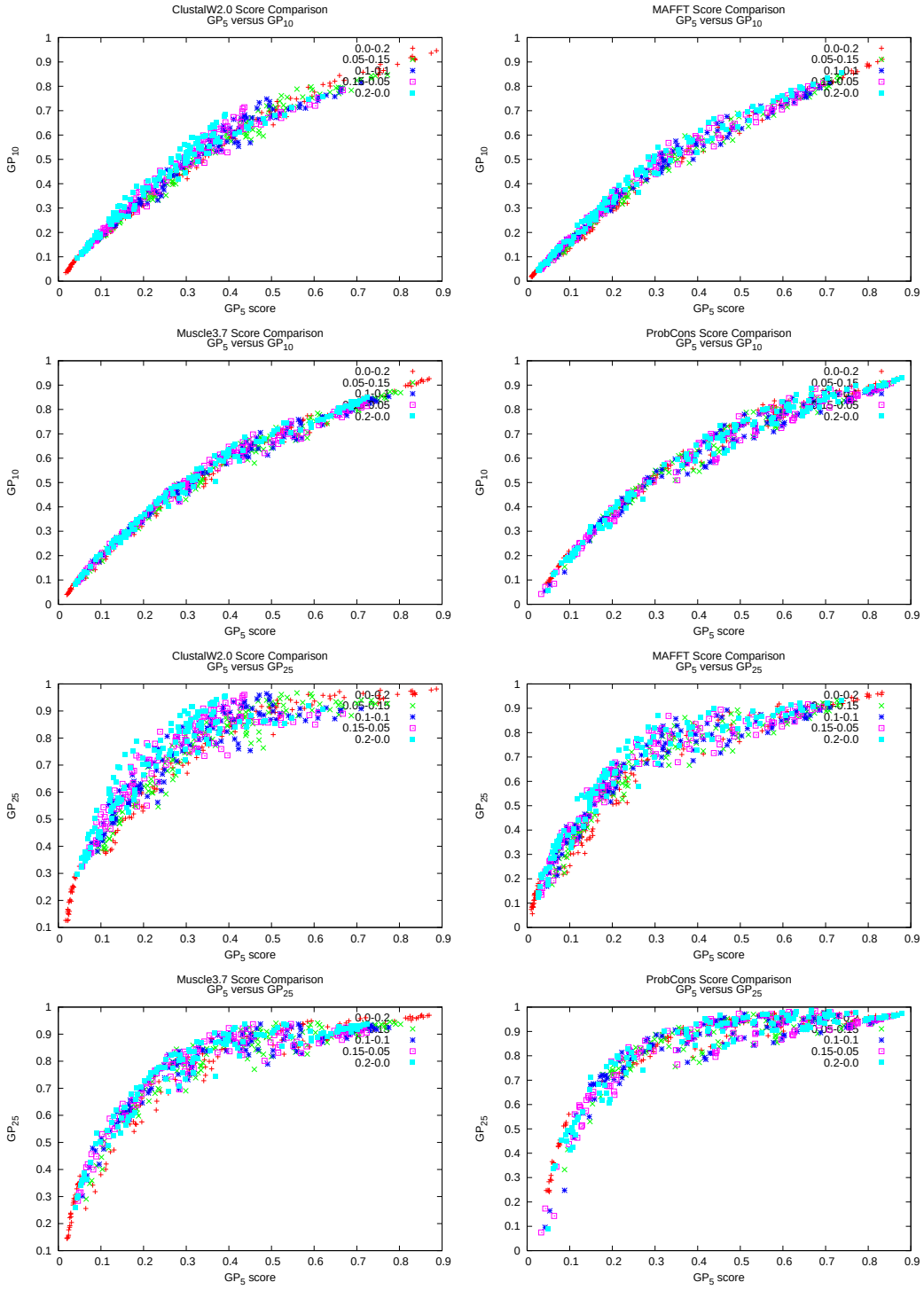

ClustalW2.0 Score Comparison
GP5 versus GP10
MAFFT Score Comparison
GP5 versus GP10
Muscle3.7 Score Comparison
GP5 versus GP10
ProbCons Score Comparison
GP5 versus GP10
ClustalW2.0 Score Comparison
GP5 versus GP25
MAFFT Score Comparison
GP5 versus GP25
Muscle3.7 Score Comparison
GP5 versus GP25
ProbCons Score Comparison
GP5 versus GP25
GP10
GP25
GP5 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

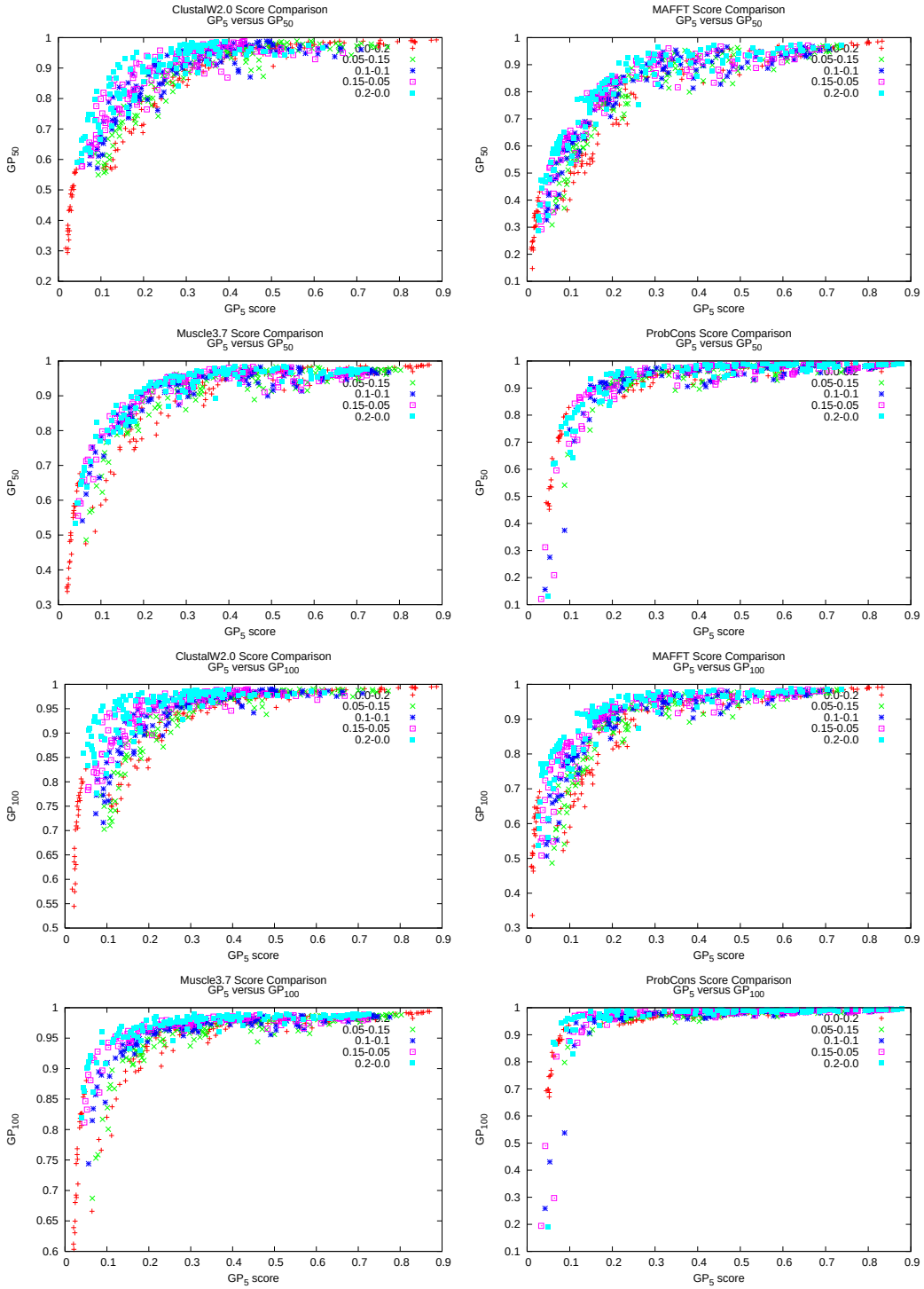

ClustalW2.0 Score Comparison
GP5 versus GP50
MAFFT Score Comparison
GP5 versus GP50
Muscle3.7 Score Comparison
GP5 versus GP50
ProbCons Score Comparison
GP5 versus GP50
ClustalW2.0 Score Comparison
GP5 versus GP100
MAFFT Score Comparison
GP5 versus GP100
Muscle3.7 Score Comparison
GP5 versus GP100
ProbCons Score Comparison
GP5 versus GP100
GP50
GP100
GP5 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

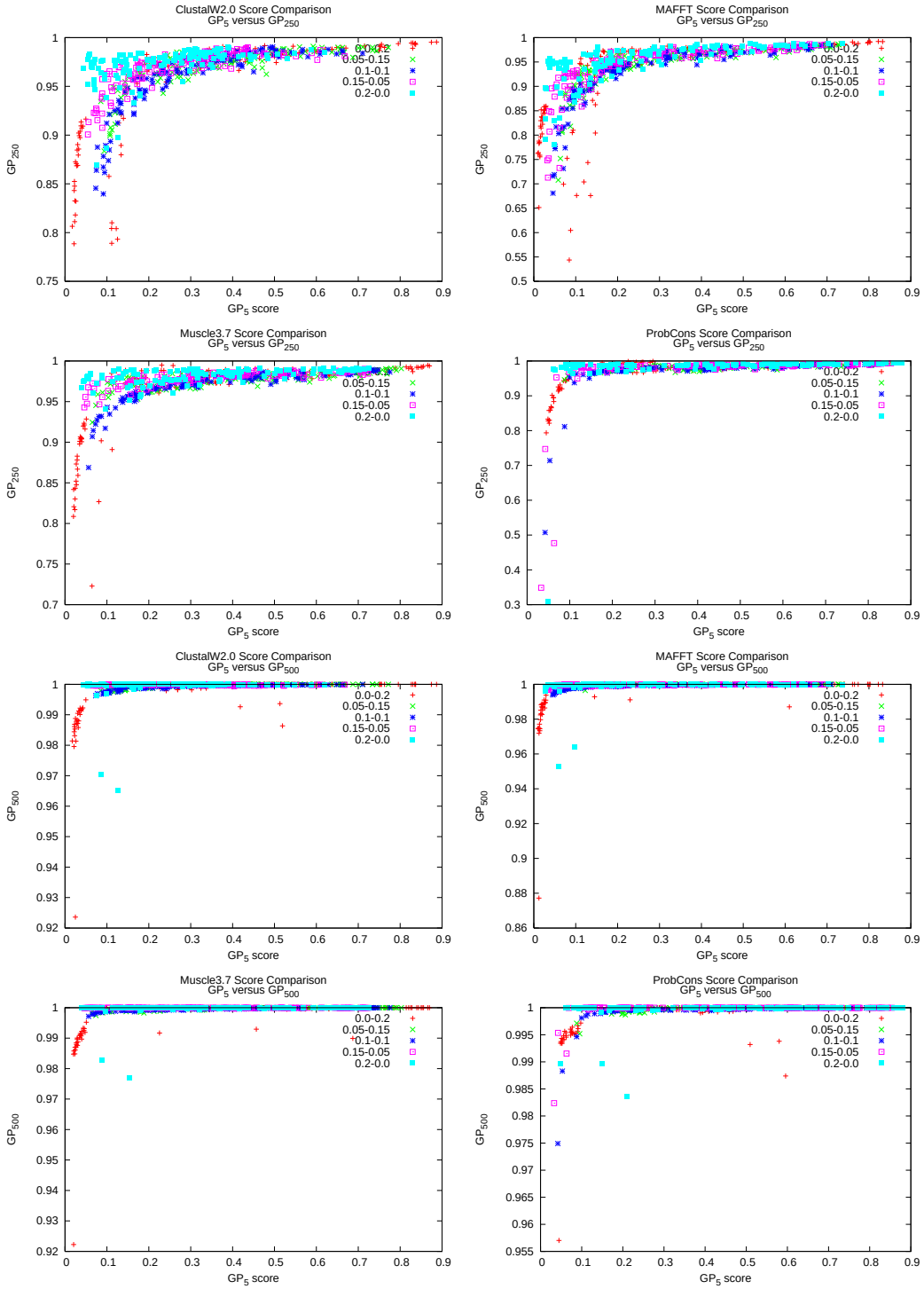

ClustalW2.0 Score Comparison
GP5 versus GP250
MAFFT Score Comparison
GP5 versus GP250
Muscle3.7 Score Comparison
GP5 versus GP250
ProbCons Score Comparison
GP5 versus GP250
ClustalW2.0 Score Comparison
GP5 versus GP500
MAFFT Score Comparison
GP5 versus GP500
Muscle3.7 Score Comparison
GP5 versus GP500
ProbCons Score Comparison
GP5 versus GP500
GP250
GP500
GP5 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

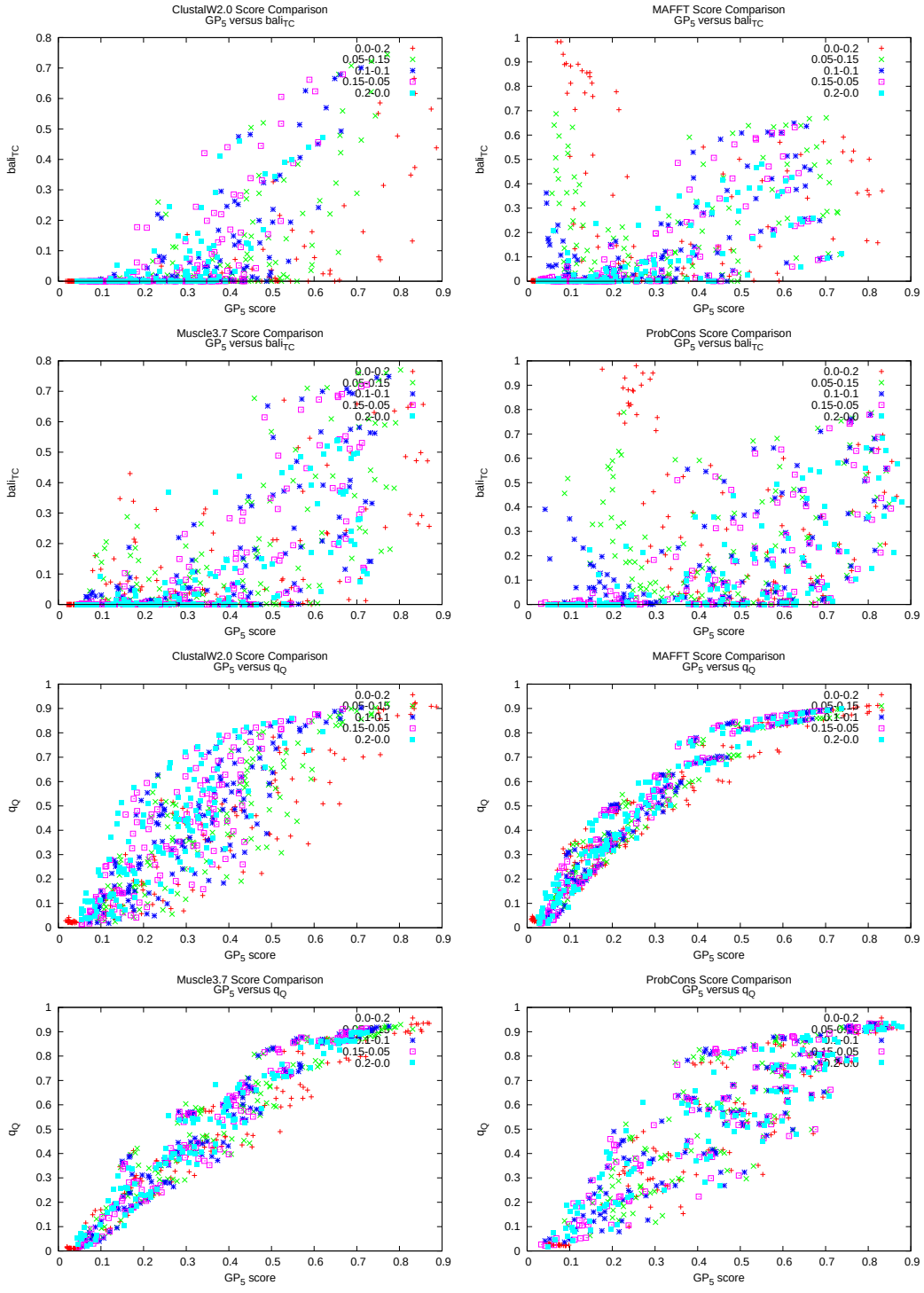

ClustalW2.0 Score Comparison
GP5 versus balitc
MAFFT Score Comparison
GP5 versus balitc
Muscle3.7 Score Comparison
GP5 versus balitc
ProbCons Score Comparison
GP5 versus balitc
ClustalW2.0 Score Comparison
GP5 versus qQ
MAFFT Score Comparison
GP5 versus qQ
Muscle3.7 Score Comparison
GP5 versus qQ
ProbCons Score Comparison
GP5 versus qQ
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
balitc
qQ
GP5 score

Figure B.2

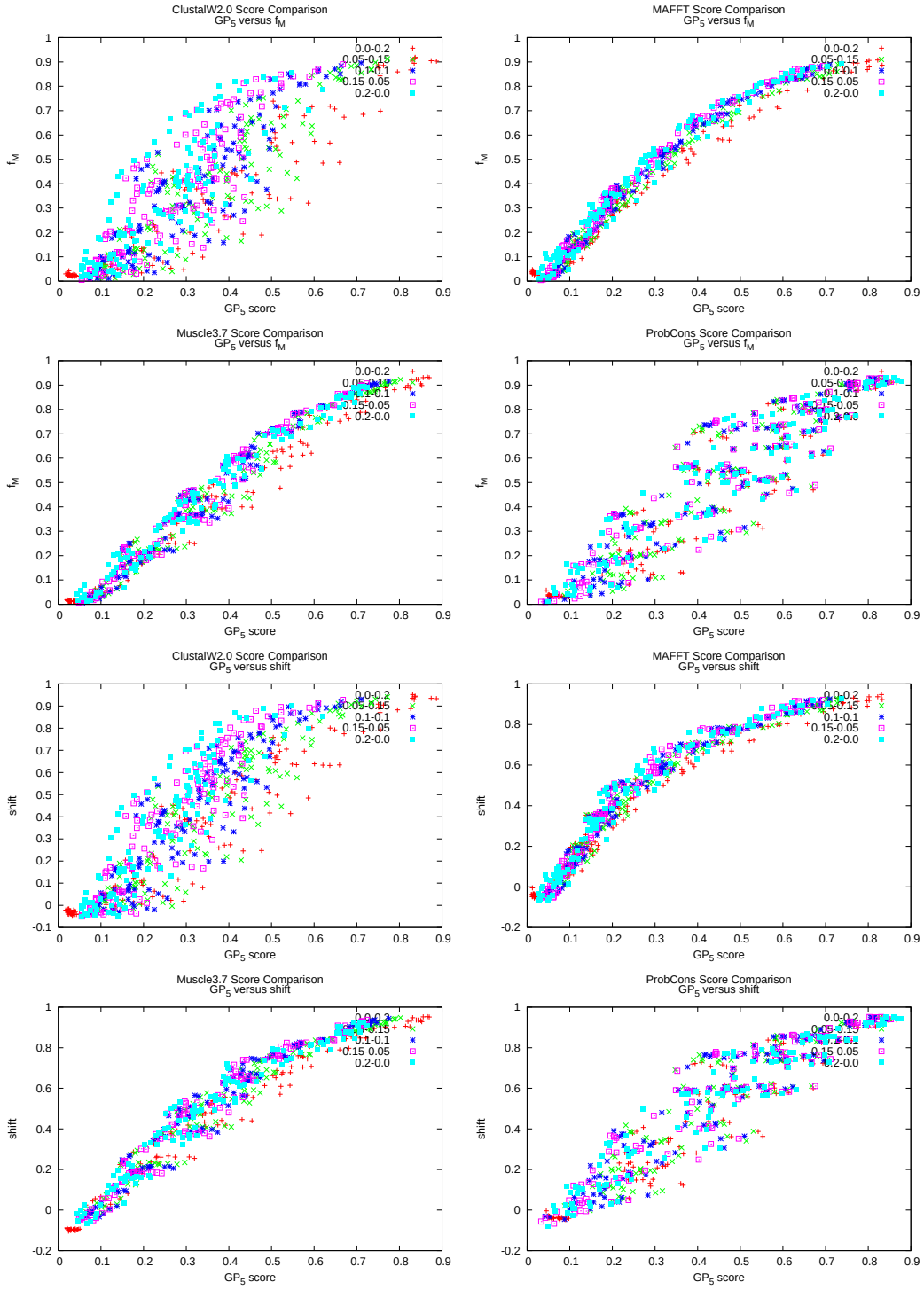

ClustalW2.0 Score Comparison
GP5 versus fM
MAFFT Score Comparison
GP5 versus fM
Muscle3.7 Score Comparison
GP5 versus fM
ProbCons Score Comparison
GP5 versus fM
ClustalW2.0 Score Comparison
GP5 versus shift
MAFFT Score Comparison
GP5 versus shift
Muscle3.7 Score Comparison
GP5 versus shift
ProbCons Score Comparison
GP5 versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
fM
shift
GP5 score

Figure B.2

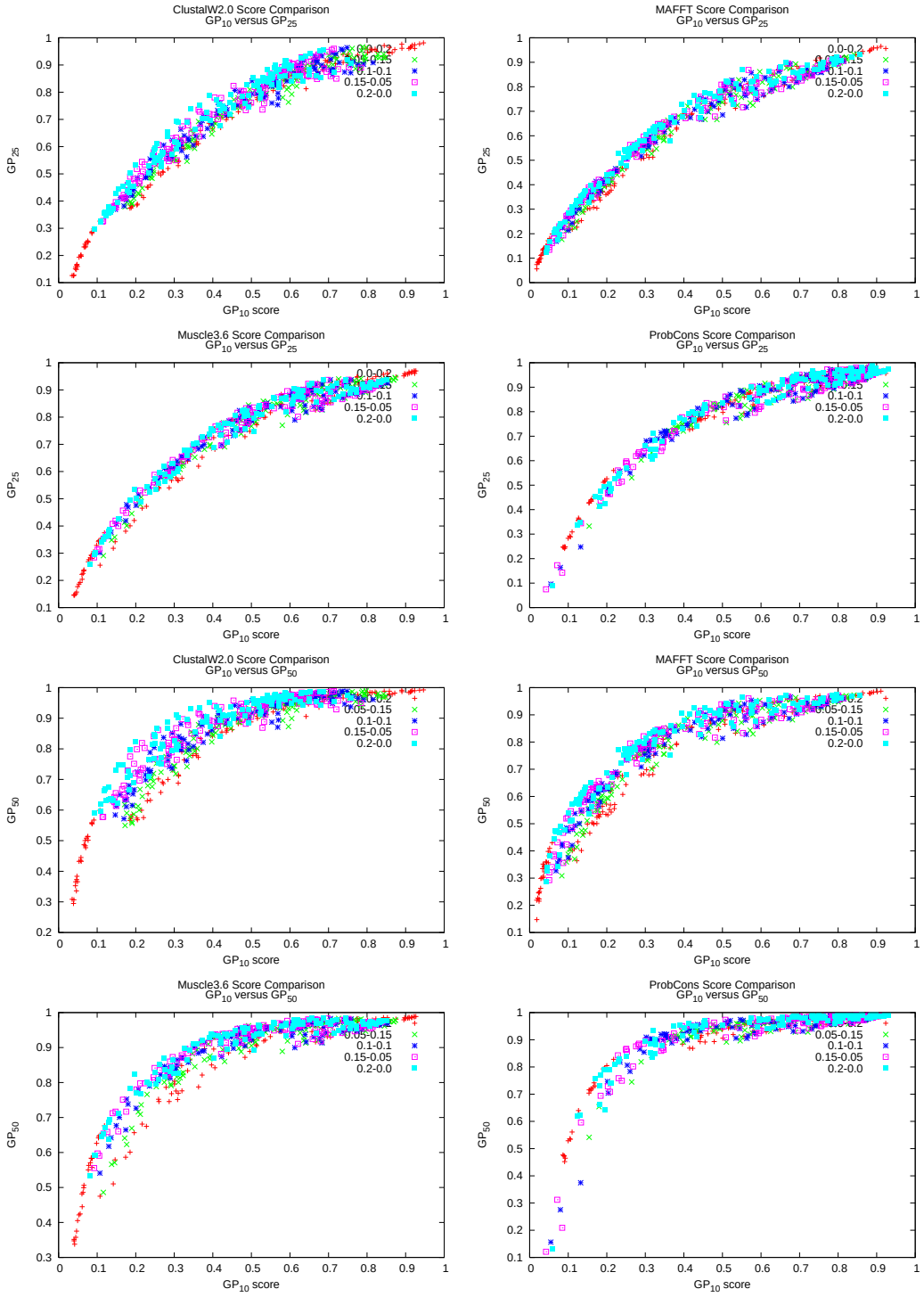

ClustalW2.0 Score Comparison
GP10 versus GP25
MAFFT Score Comparison
GP10 versus GP25
Muscle3.6 Score Comparison
GP10 versus GP25
ProbCons Score Comparison
GP10 versus GP25
ClustalW2.0 Score Comparison
GP10 versus GP50
MAFFT Score Comparison
GP10 versus GP50
Muscle3.6 Score Comparison
GP10 versus GP50
ProbCons Score Comparison
GP10 versus GP50
GP25
GP50
GP10 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

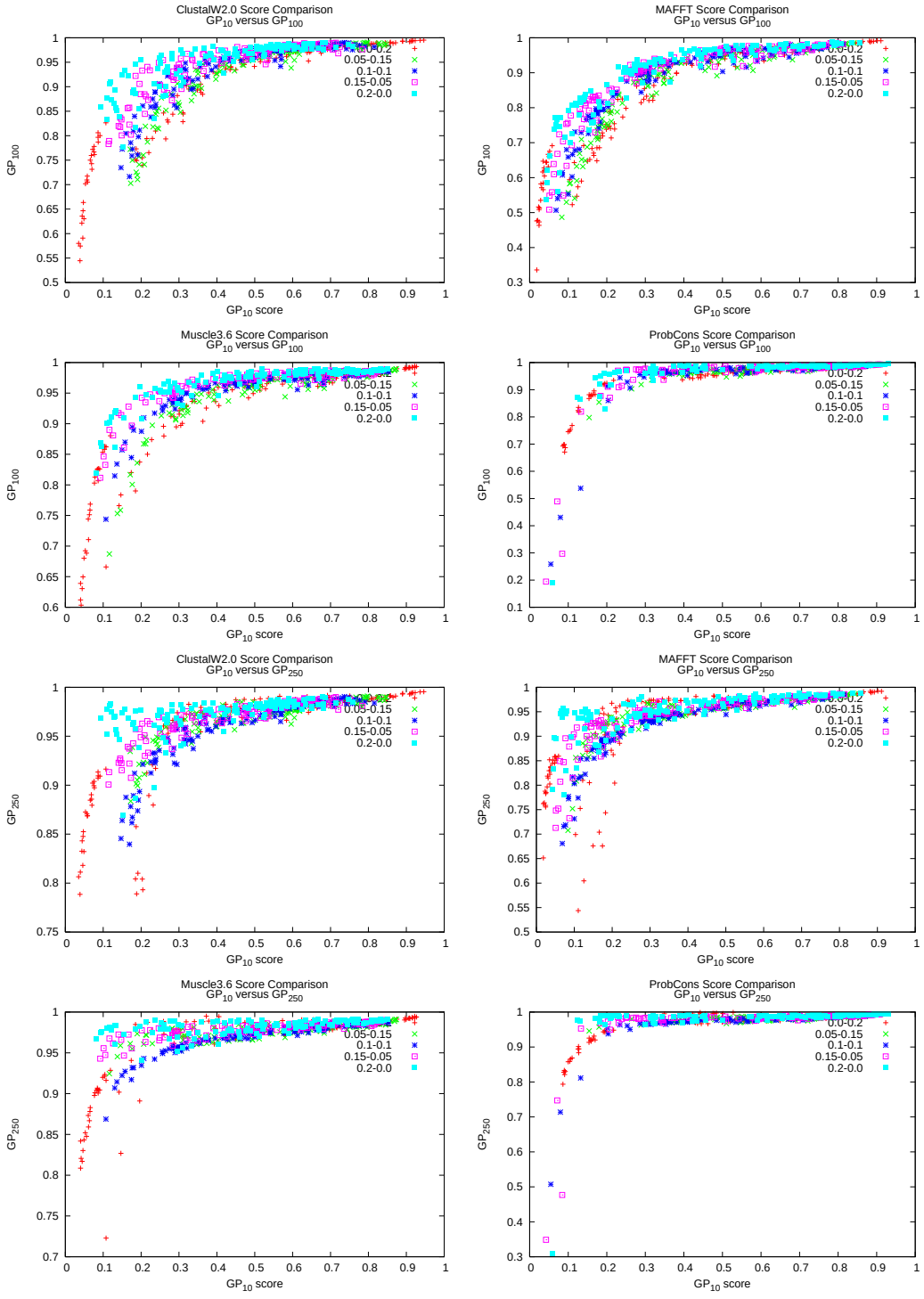

ClustalW2.0 Score Comparison
GP10 versus GP100
MAFFT Score Comparison
GP10 versus GP100
Muscle3.6 Score Comparison
GP10 versus GP100
ProbCons Score Comparison
GP10 versus GP100
ClustalW2.0 Score Comparison
GP10 versus GP250
MAFFT Score Comparison
GP10 versus GP250
Muscle3.6 Score Comparison
GP10 versus GP250
ProbCons Score Comparison
GP10 versus GP250
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
GP100
GP250
GP10 score

Figure B.2

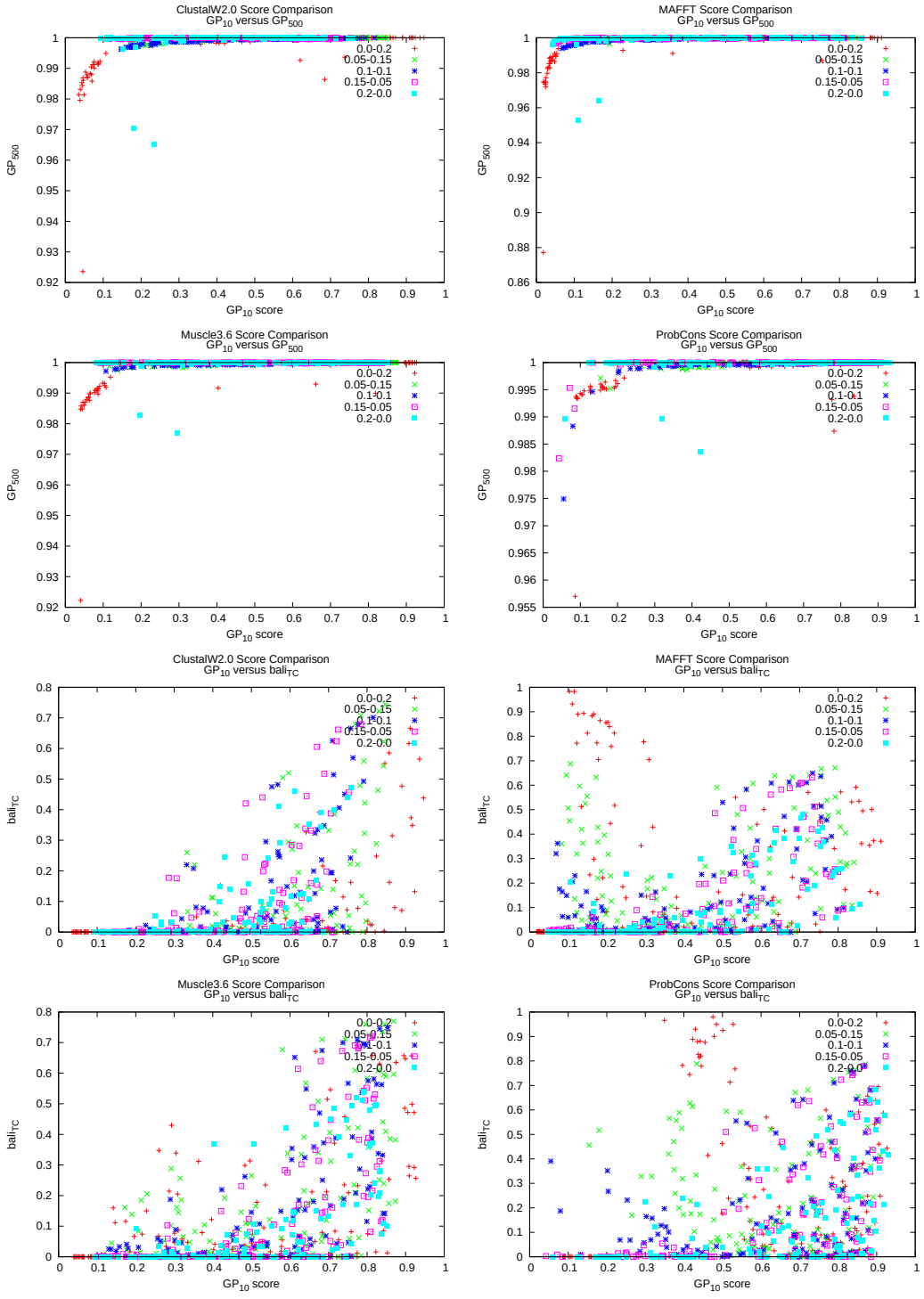

ClustalW2.0 Score Comparison
GP10 versus GP500
MAFFT Score Comparison
GP10 versus GP500
Muscle3.6 Score Comparison
GP10 versus GP500
ProbCons Score Comparison
GP10 versus GP500
ClustalW2.0 Score Comparison
GP10 versus baliTC
MAFFT Score Comparison
GP10 versus baliTC
Muscle3.6 Score Comparison
GP10 versus baliTC
ProbCons Score Comparison
GP10 versus baliTC
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
GP500
baliTC
GP10 score

Figure B.2

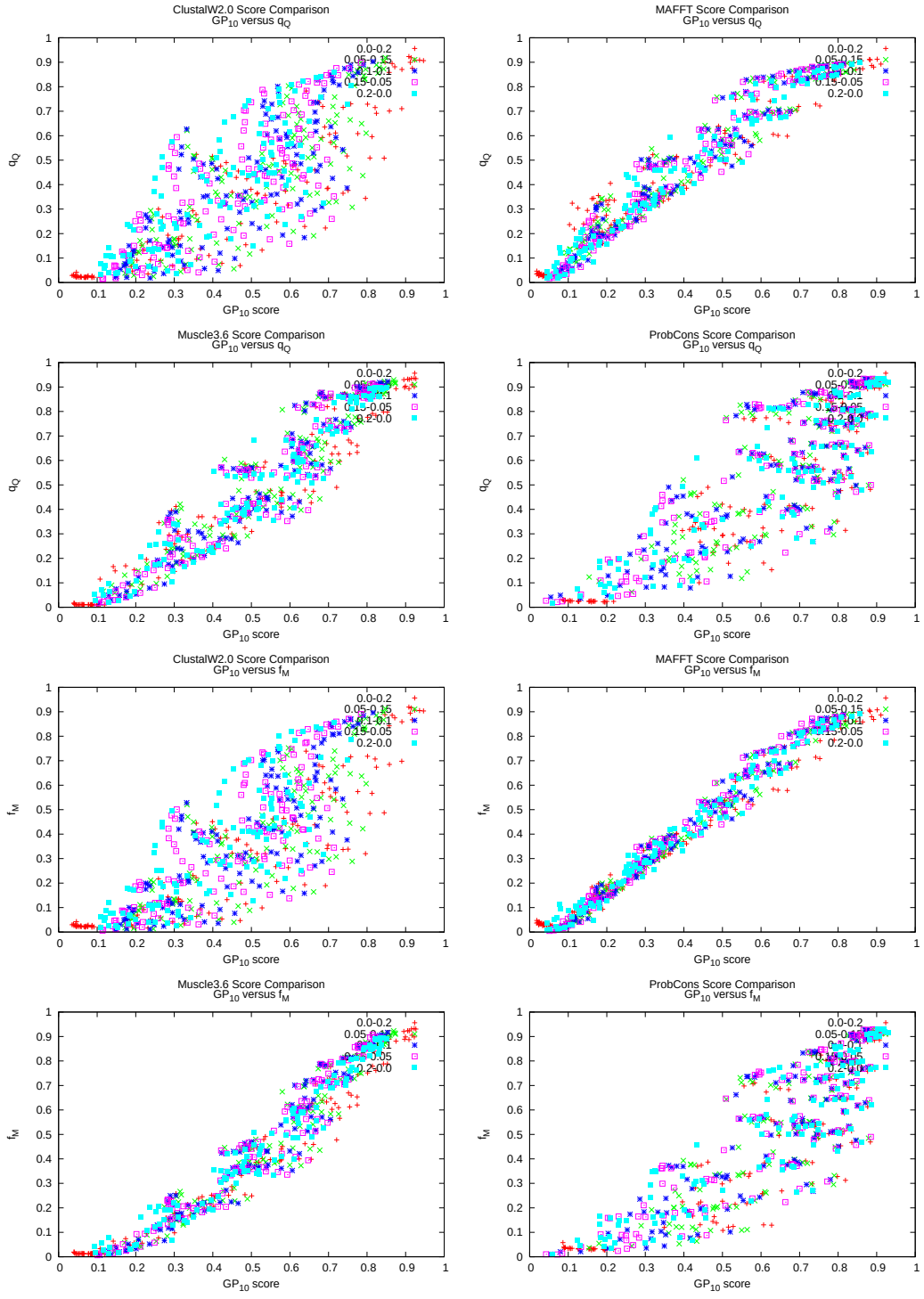

ClustalW2.0 Score Comparison
GP10 versus qQ
MAFFT Score Comparison
GP10 versus qQ
Muscle3.6 Score Comparison
GP10 versus qQ
ProbCons Score Comparison
GP10 versus qQ
ClustalW2.0 Score Comparison
GP10 versus fM
MAFFT Score Comparison
GP10 versus fM
Muscle3.6 Score Comparison
GP10 versus fM
ProbCons Score Comparison
GP10 versus fM
qQ
fM
GP10 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

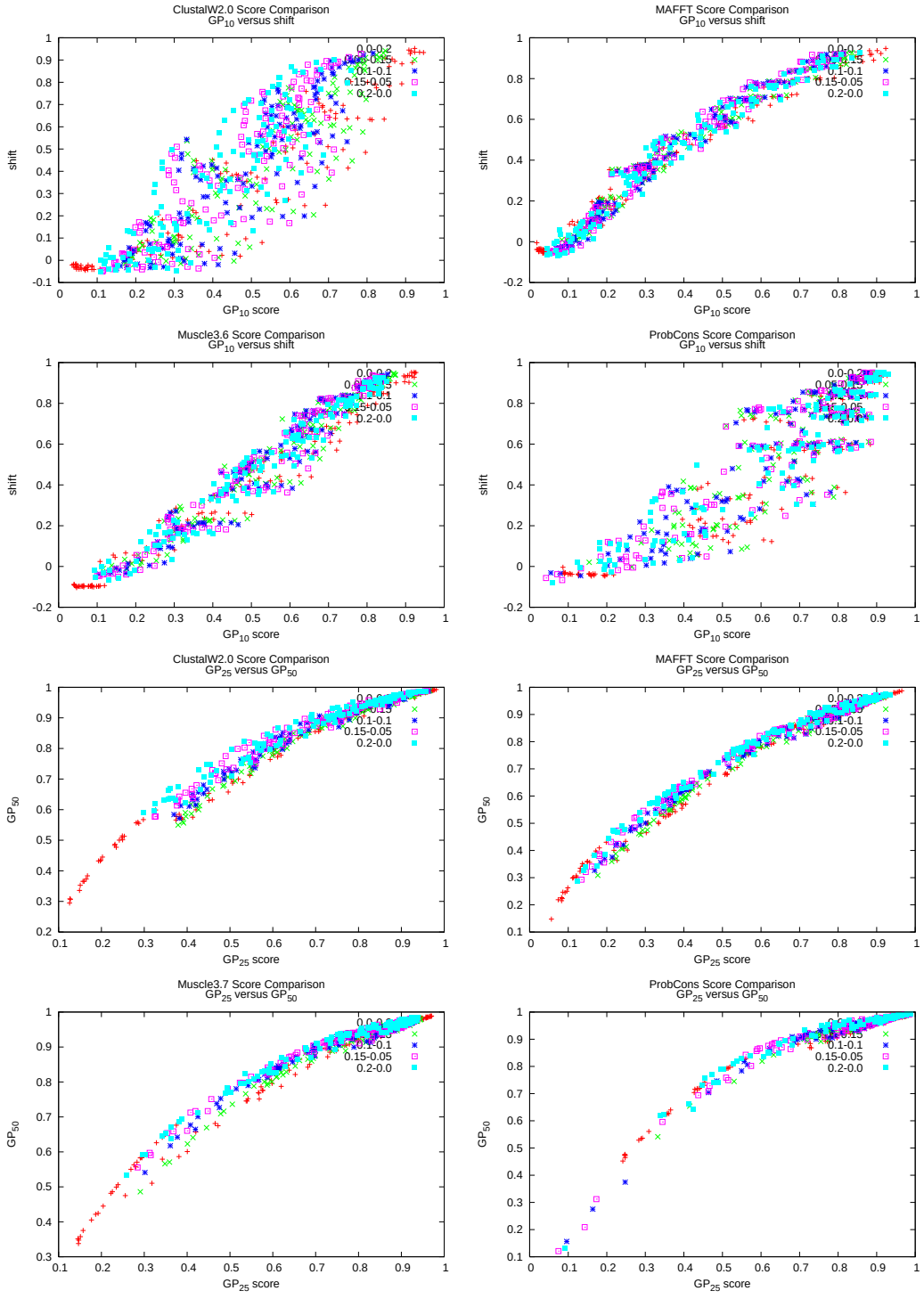

ClustalW2.0 Score Comparison
GP10 versus shift
MAFFT Score Comparison
GP10 versus shift
Muscle3.6 Score Comparison
GP10 versus shift
ProbCons Score Comparison
GP10 versus shift
ClustalW2.0 Score Comparison
GP25 versus GP50
MAFFT Score Comparison
GP25 versus GP50
Muscle3.7 Score Comparison
GP25 versus GP50
ProbCons Score Comparison
GP25 versus GP50
shift
GP10 score
GP50
GP25 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

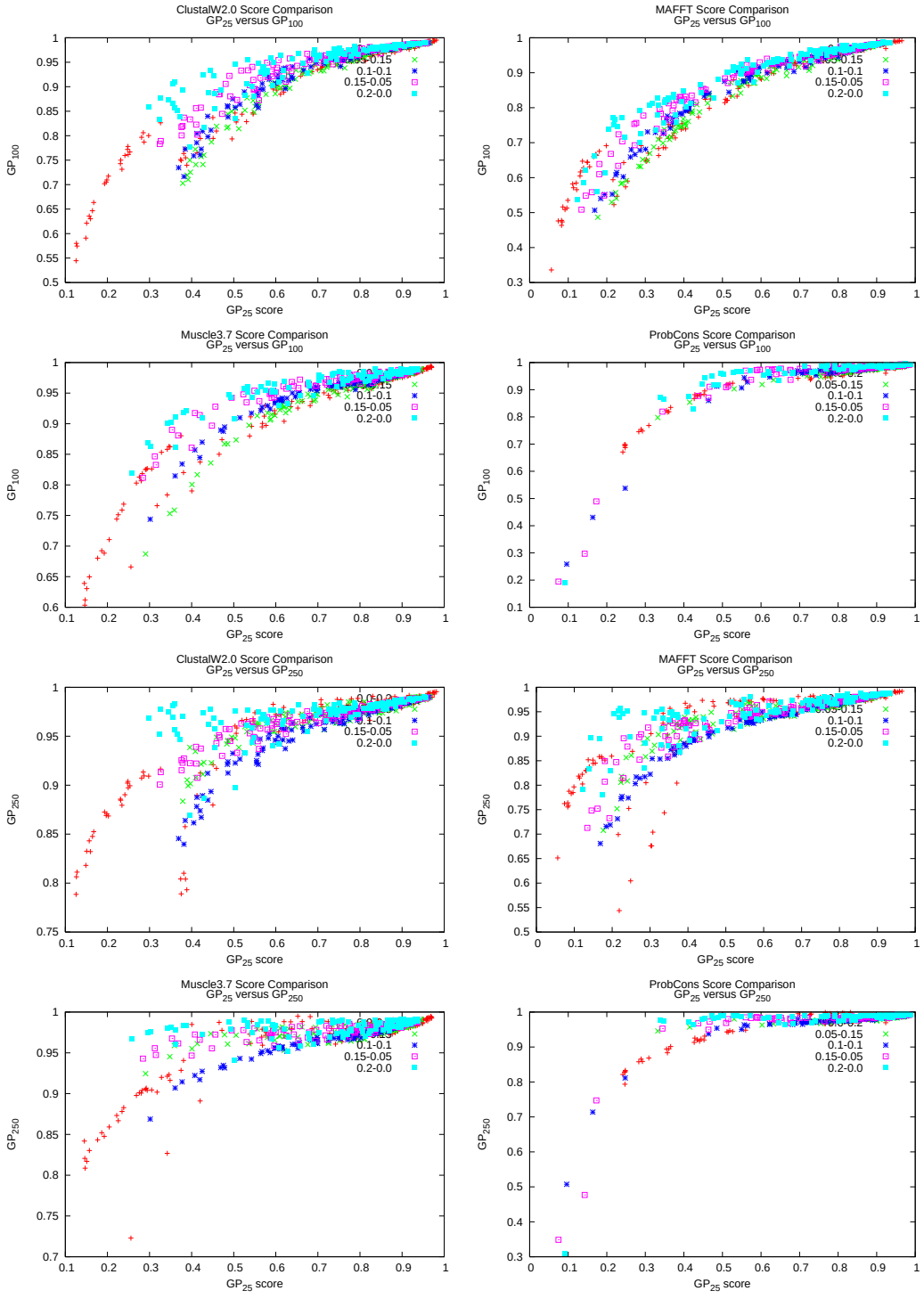

ClustalW2.0 Score Comparison
GP25 versus GP100
MAFFT Score Comparison
GP25 versus GP100
Muscle3.7 Score Comparison
GP25 versus GP100
ProbCons Score Comparison
GP25 versus GP100
ClustalW2.0 Score Comparison
GP25 versus GP250
MAFFT Score Comparison
GP25 versus GP250
Muscle3.7 Score Comparison
GP25 versus GP250
ProbCons Score Comparison
GP25 versus GP250
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
GP100
GP250
GP25 score

Figure B.2

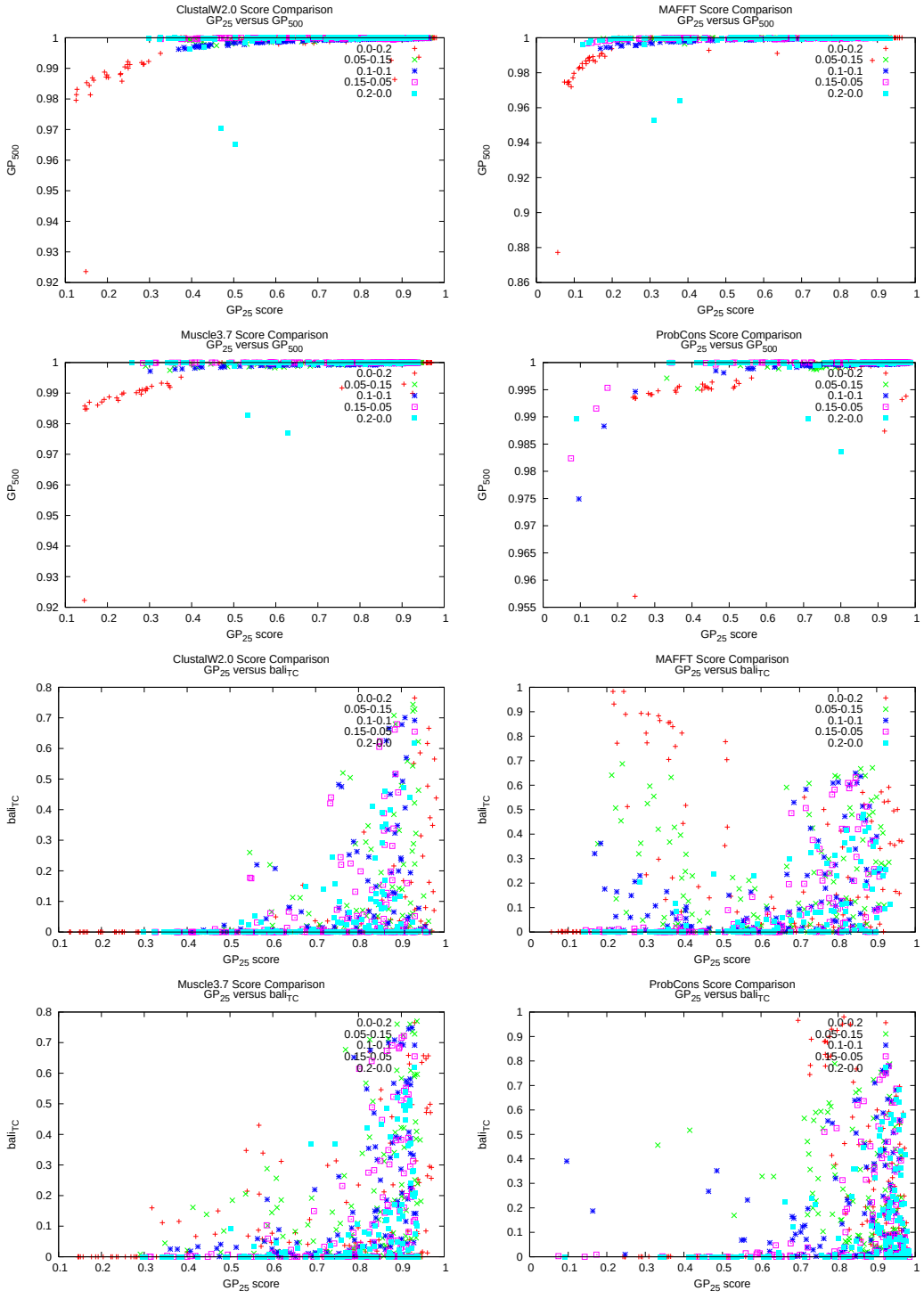

ClustalW2.0 Score Comparison
GP25 versus GP500
MAFFT Score Comparison
GP25 versus GP500
Muscle3.7 Score Comparison
GP25 versus GP500
ProbCons Score Comparison
GP25 versus GP500
ClustalW2.0 Score Comparison
GP25 versus balitc
MAFFT Score Comparison
GP25 versus balitc
Muscle3.7 Score Comparison
GP25 versus balitc
ProbCons Score Comparison
GP25 versus balitc
GP500
balitc
GP25 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

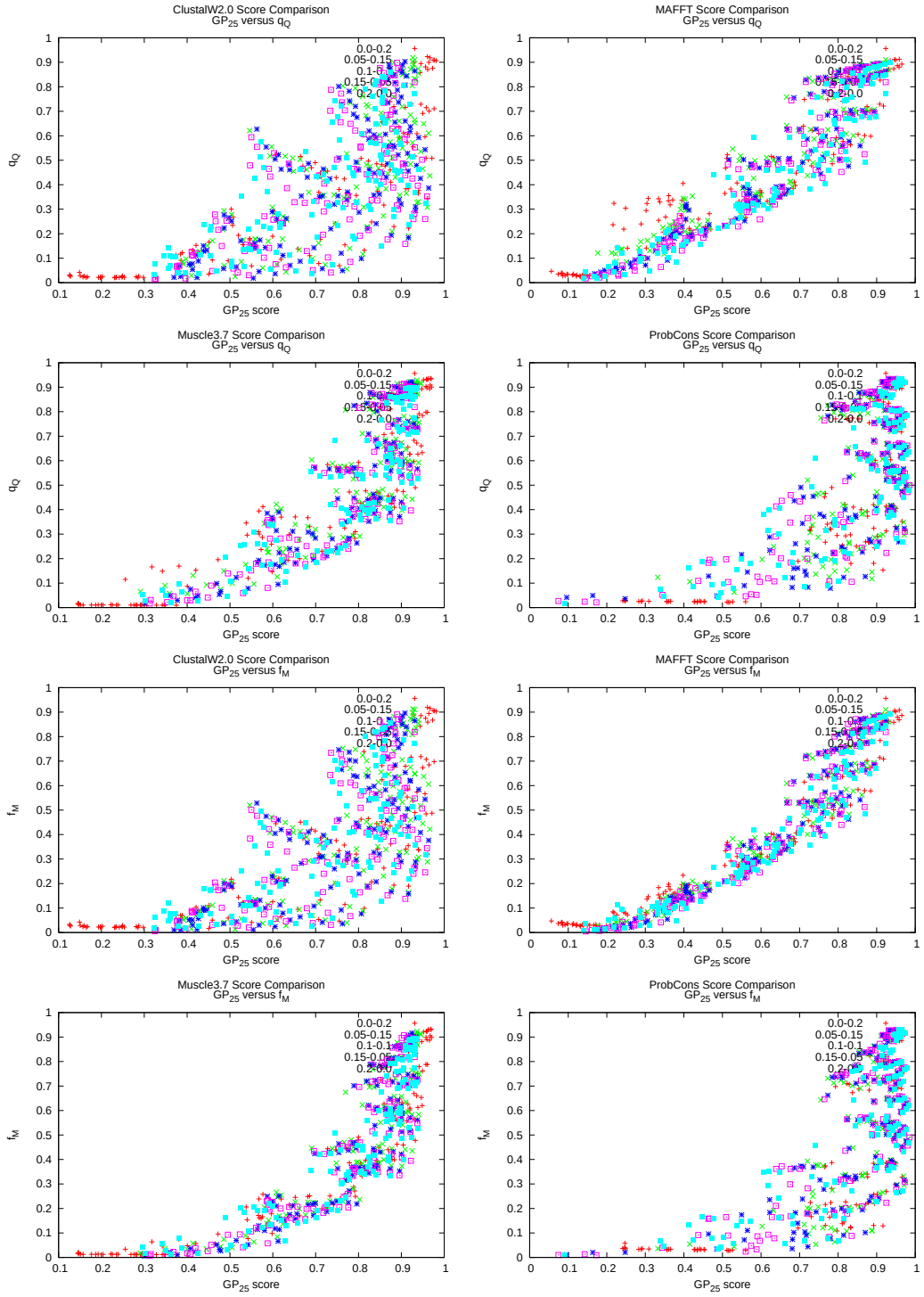

ClustalW2.0 Score Comparison
GP25 versus qQ
MAFFT Score Comparison
GP25 versus qQ
Muscle3.7 Score Comparison
GP25 versus qQ
ProbCons Score Comparison
GP25 versus qQ
ClustalW2.0 Score Comparison
GP25 versus fM
MAFFT Score Comparison
GP25 versus fM
Muscle3.7 Score Comparison
GP25 versus fM
ProbCons Score Comparison
GP25 versus fM
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
qQ
fM
GP25 score

Figure B.2

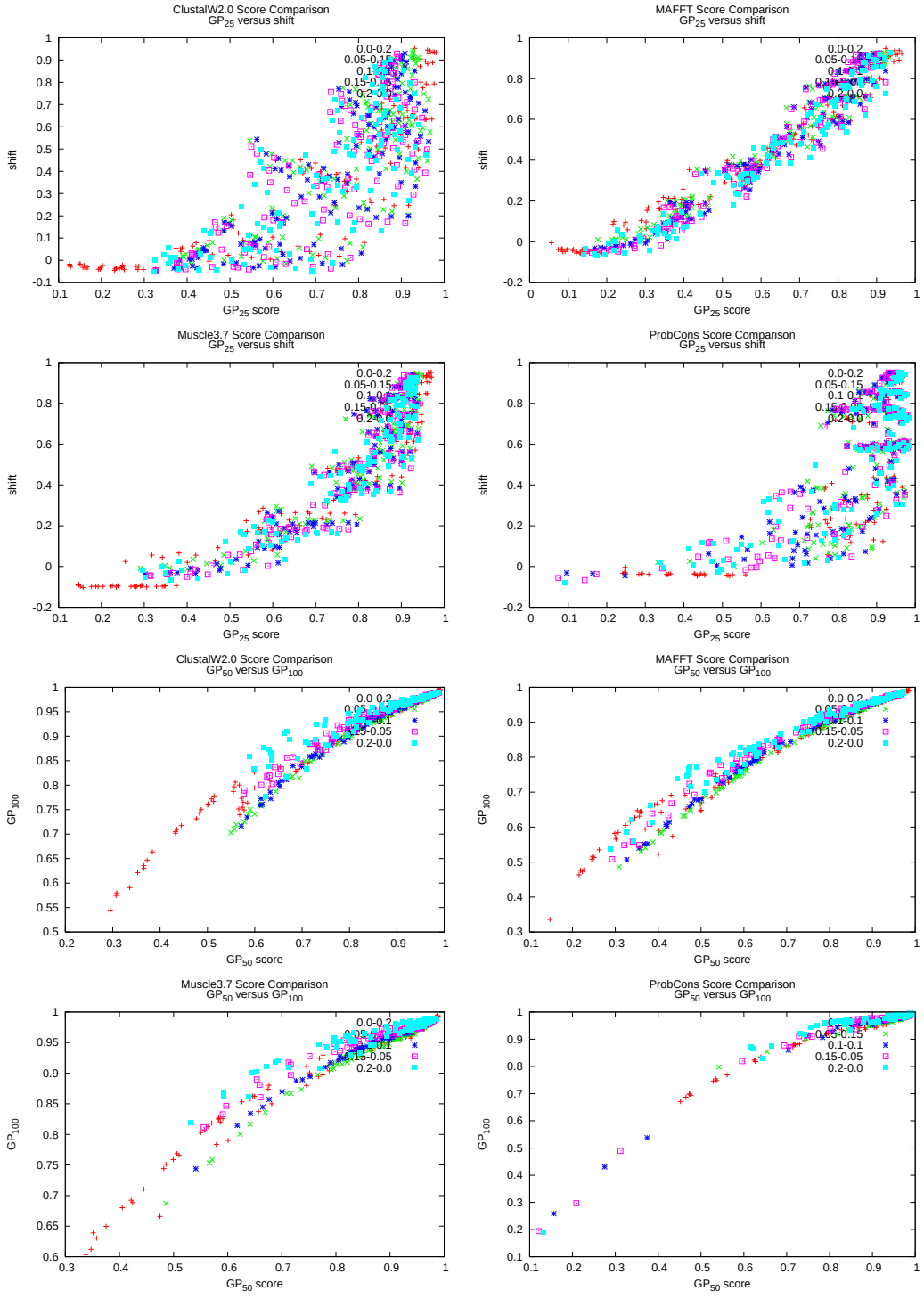

ClustalW2.0 Score Comparison
GP25 versus shift
MAFFT Score Comparison
GP25 versus shift
Muscle3.7 Score Comparison
GP25 versus shift
ProbCons Score Comparison
GP25 versus shift
ClustalW2.0 Score Comparison
GP50 versus GP100
MAFFT Score Comparison
GP50 versus GP100
Muscle3.7 Score Comparison
GP50 versus GP100
ProbCons Score Comparison
GP50 versus GP100
shift
GP25 score
GP100
GP50 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

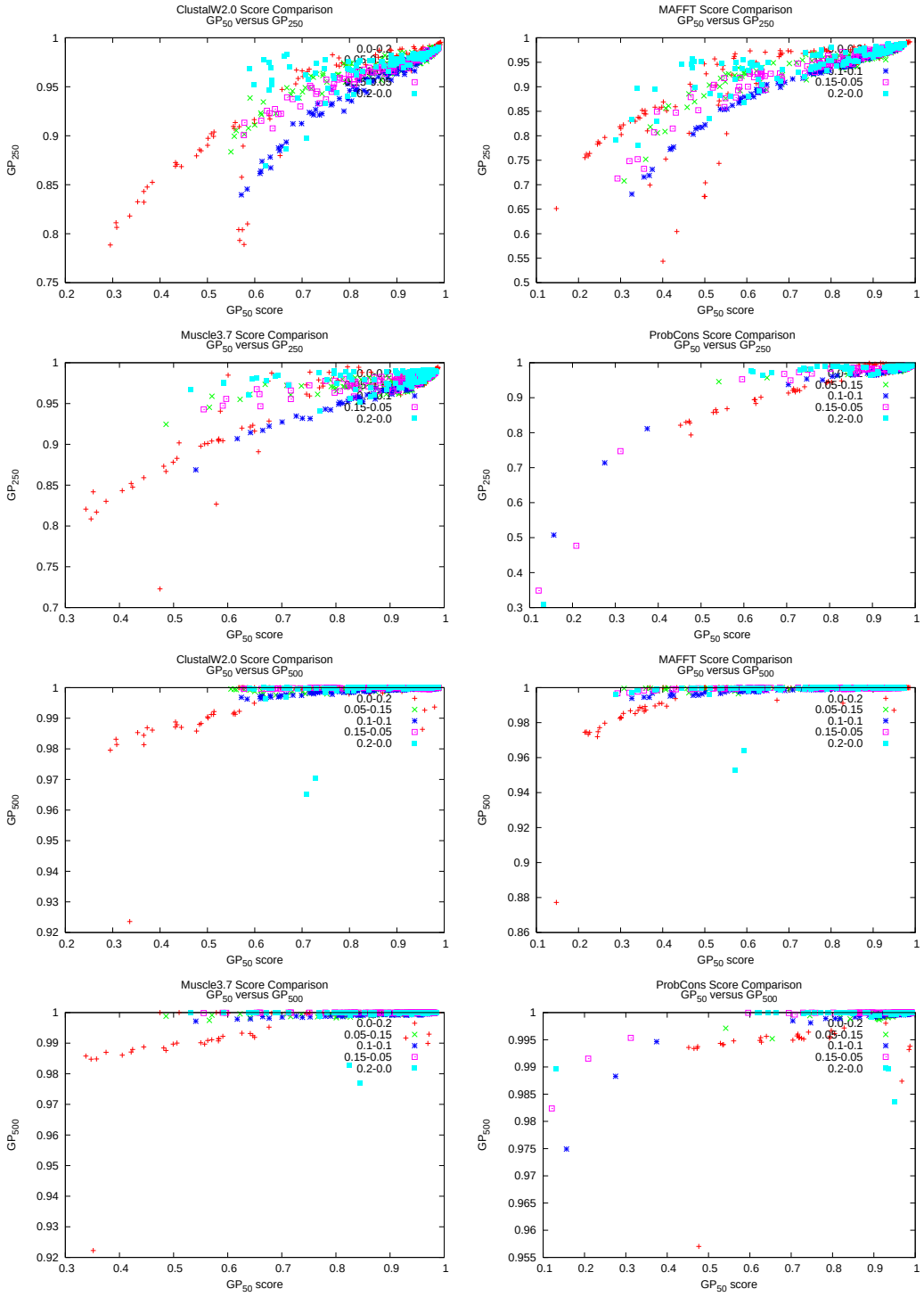

ClustalW2.0 Score Comparison
GP50 versus GP250
MAFFT Score Comparison
GP50 versus GP250
Muscle3.7 Score Comparison
GP50 versus GP250
ProbCons Score Comparison
GP50 versus GP250
ClustalW2.0 Score Comparison
GP50 versus GP500
MAFFT Score Comparison
GP50 versus GP500
Muscle3.7 Score Comparison
GP50 versus GP500
ProbCons Score Comparison
GP50 versus GP500
GP250
GP500
GP50 score
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

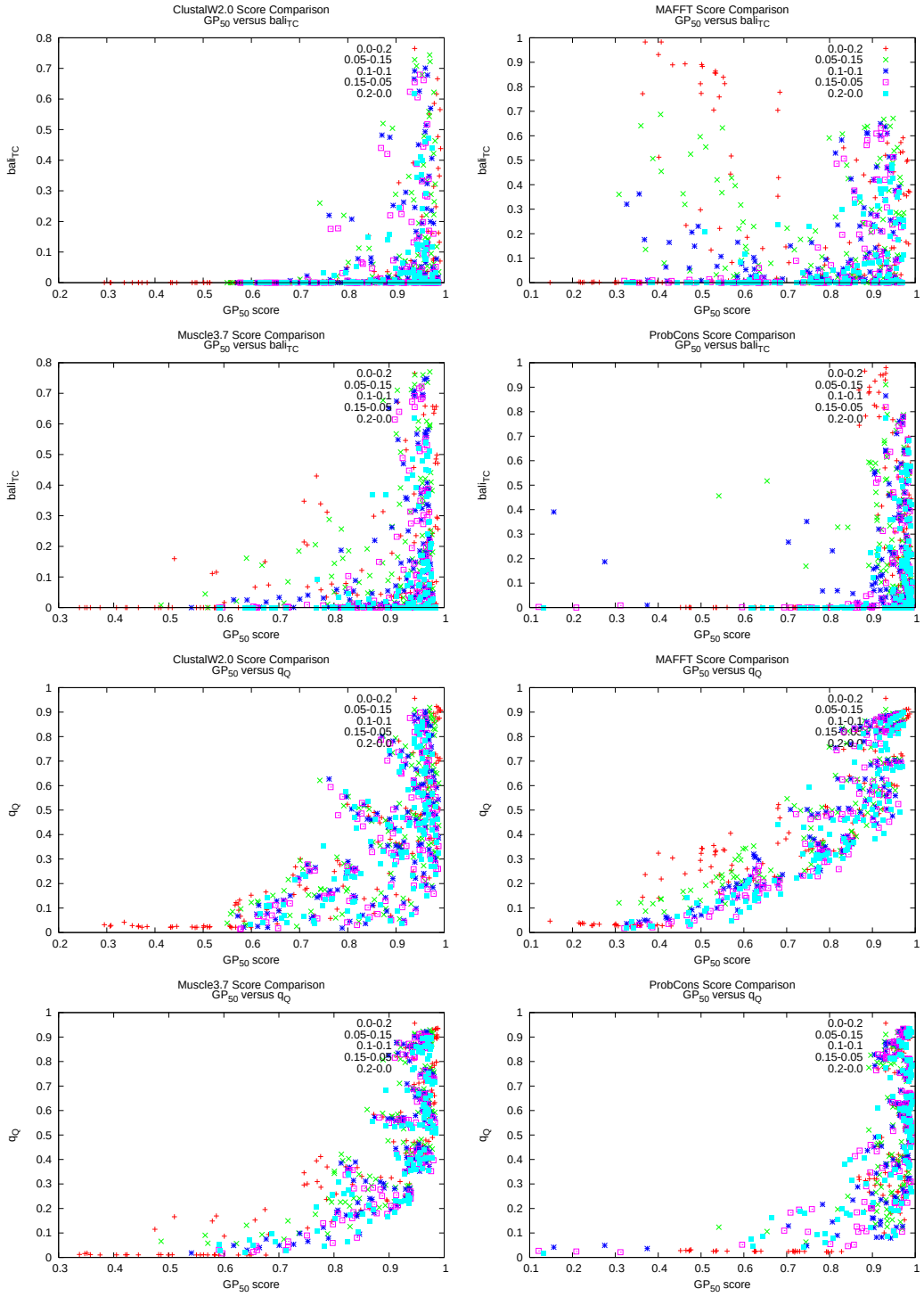

ClustalW2.0 Score Comparison
GP50 versus baliTC
MAFFT Score Comparison
GP50 versus baliTC
Muscle3.7 Score Comparison
GP50 versus baliTC
ProbCons Score Comparison
GP50 versus baliTC
ClustalW2.0 Score Comparison
GP50 versus qQ
MAFFT Score Comparison
GP50 versus qQ
Muscle3.7 Score Comparison
GP50 versus qQ
ProbCons Score Comparison
GP50 versus qQ
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
baliTC
qQ
GP50 score

Figure B.2

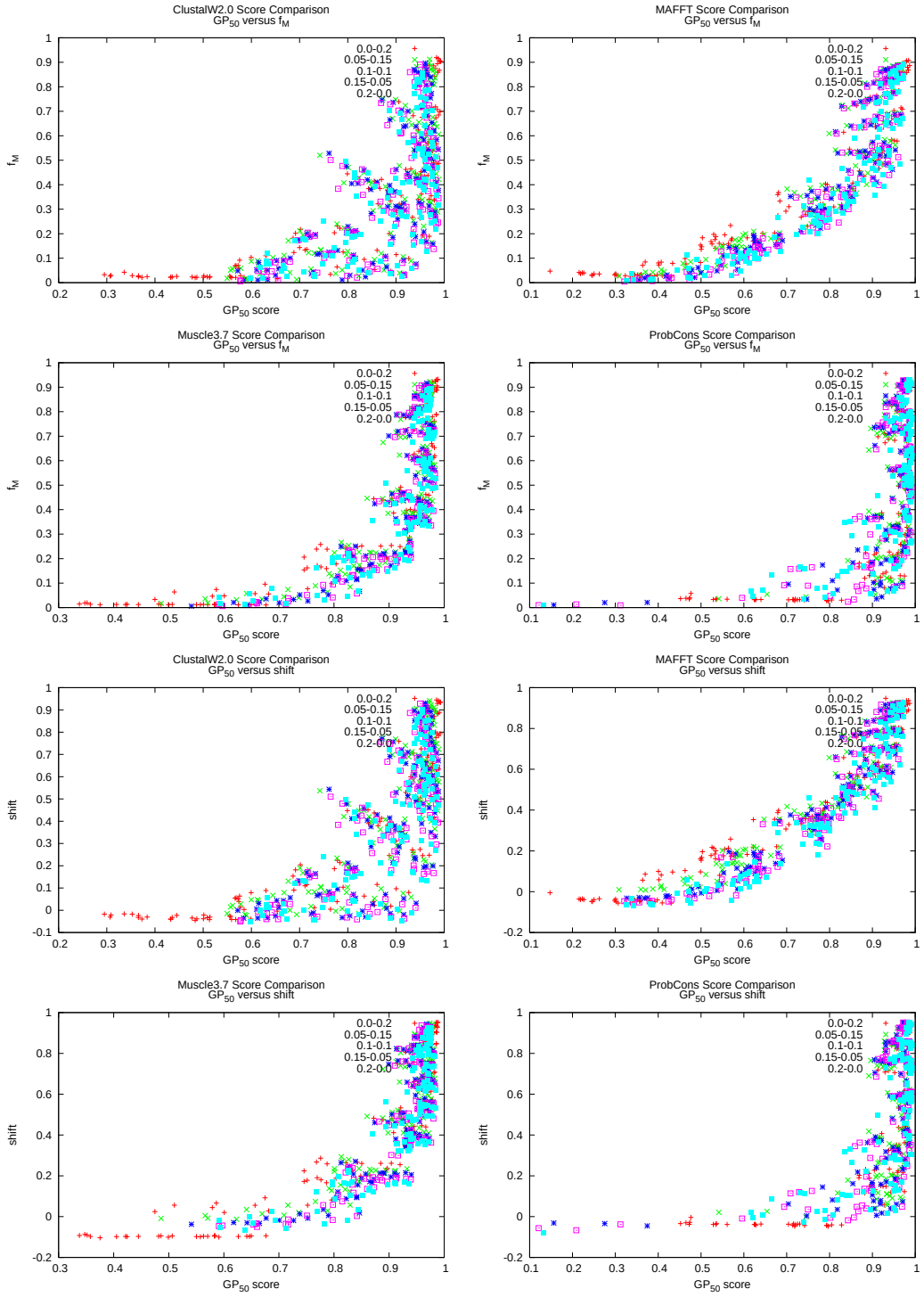

ClustalW2.0 Score Comparison
GP50 versus fM
MAFFT Score Comparison
GP50 versus fM
Muscle3.7 Score Comparison
GP50 versus fM
ProbCons Score Comparison
GP50 versus fM
ClustalW2.0 Score Comparison
GP50 versus shift
MAFFT Score Comparison
GP50 versus shift
Muscle3.7 Score Comparison
GP50 versus shift
ProbCons Score Comparison
GP50 versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
fM
shift
GP50 score

Figure B.2

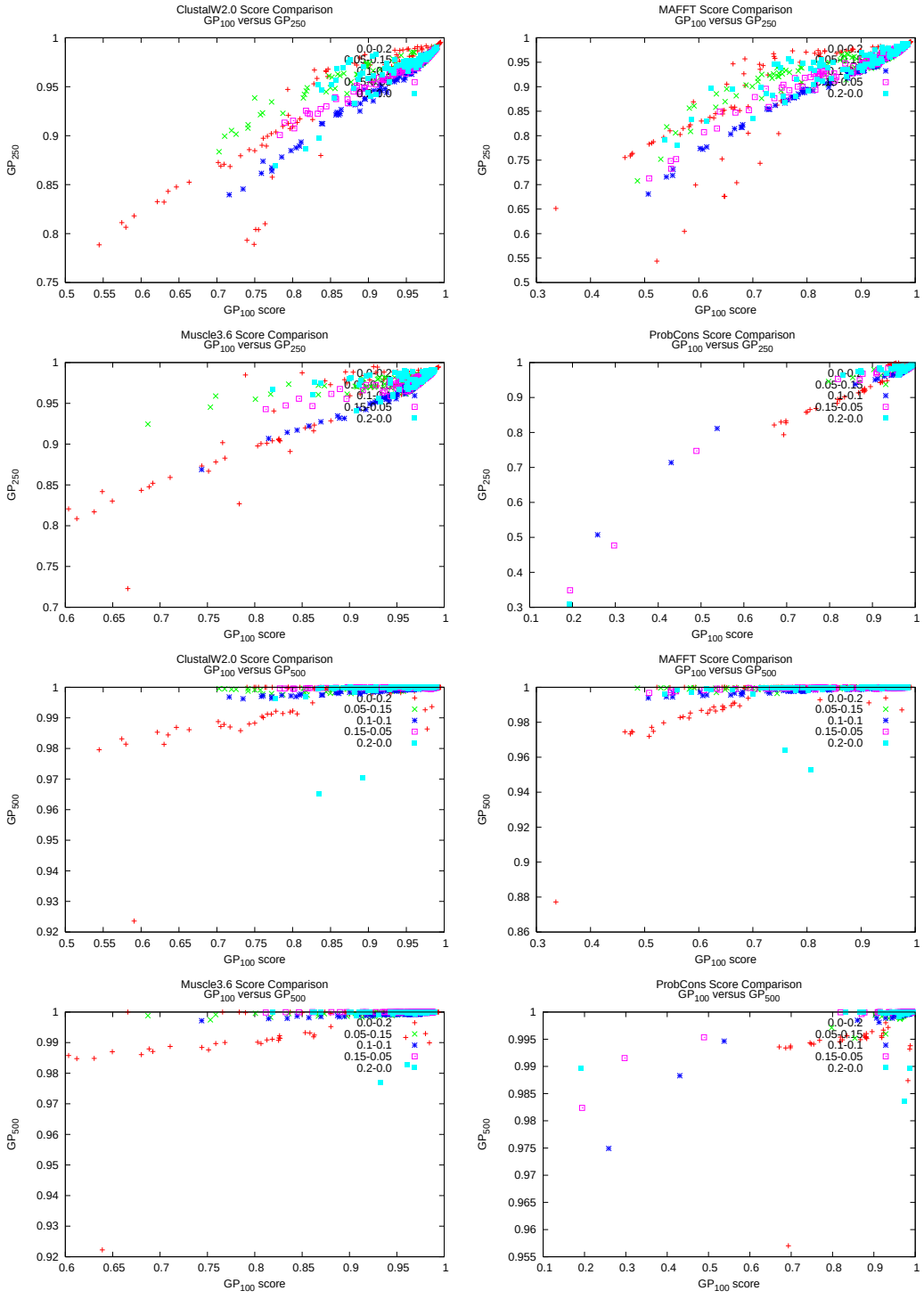

ClustalW2.0 Score Comparison
GP100 versus GP250
MAFFT Score Comparison
GP100 versus GP250
Muscle3.6 Score Comparison
GP100 versus GP250
ProbCons Score Comparison
GP100 versus GP250
ClustalW2.0 Score Comparison
GP100 versus GP500
MAFFT Score Comparison
GP100 versus GP500
Muscle3.6 Score Comparison
GP100 versus GP500
ProbCons Score Comparison
GP100 versus GP500
GP100 score
GP250
GP500
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0

Figure B.2

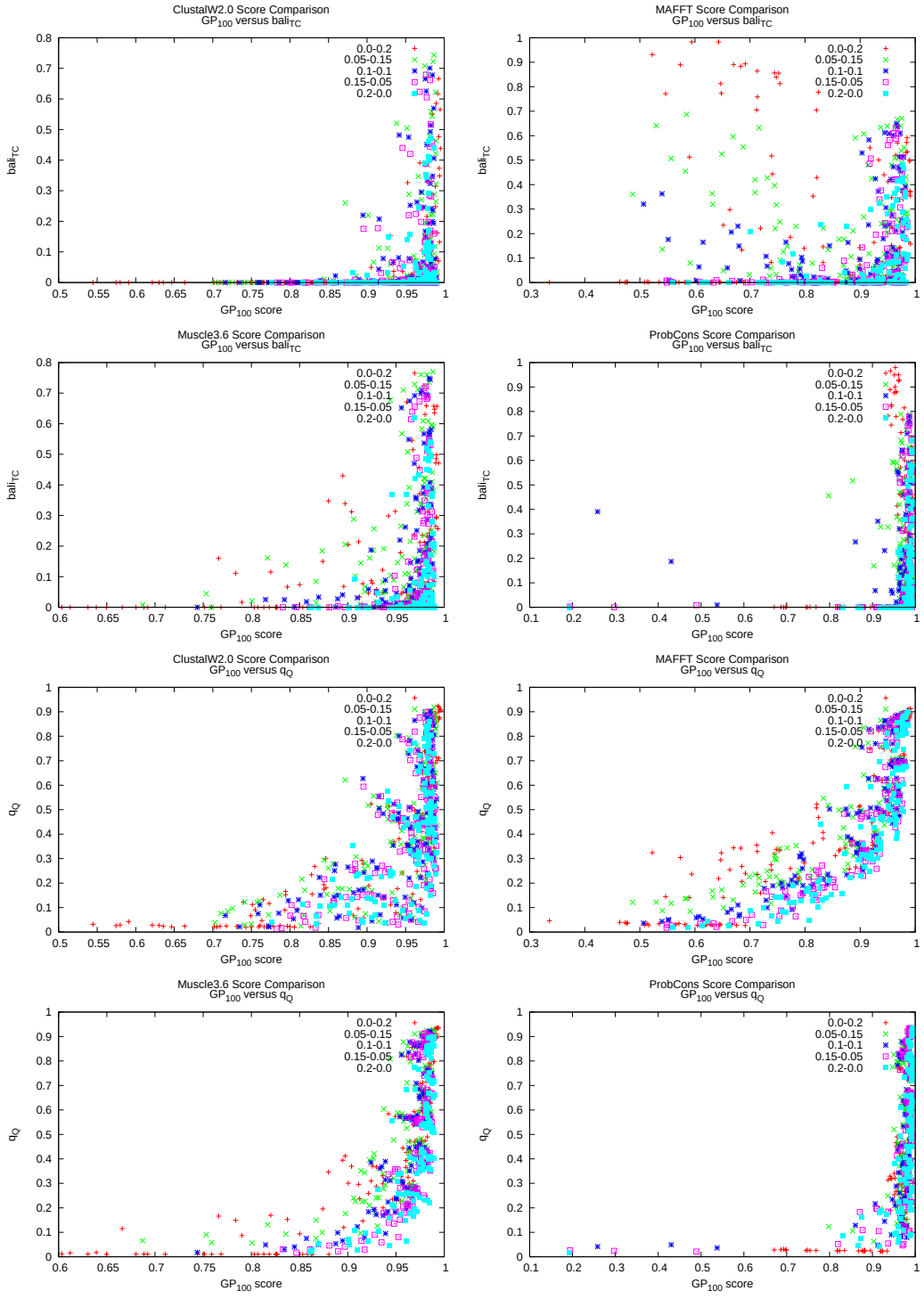

ClustalW2.0 Score Comparison
GP100 versus baliTC
MAFFT Score Comparison
GP100 versus baliTC
Muscle3.6 Score Comparison
GP100 versus baliTC
ProbCons Score Comparison
GP100 versus baliTC
ClustalW2.0 Score Comparison
GP100 versus qQ
MAFFT Score Comparison
GP100 versus qQ
Muscle3.6 Score Comparison
GP100 versus qQ
ProbCons Score Comparison
GP100 versus qQ
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
baliTC
qQ
GP100 score

Figure B.2

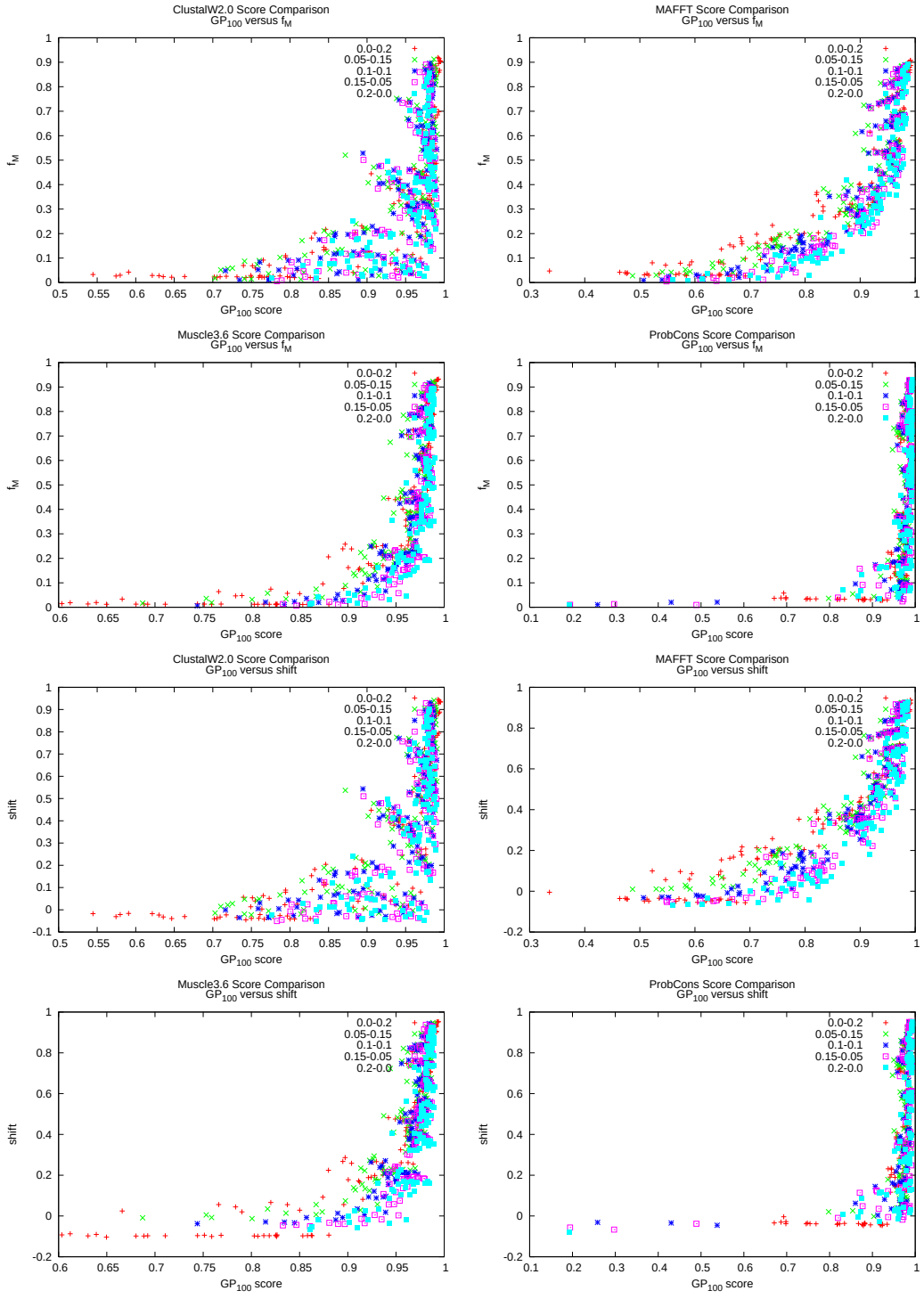

ClustalW2.0 Score Comparison
GP100 versus fM
MAFFT Score Comparison
GP100 versus fM
Muscle3.6 Score Comparison
GP100 versus fM
ProbCons Score Comparison
GP100 versus fM
ClustalW2.0 Score Comparison
GP100 versus shift
MAFFT Score Comparison
GP100 versus shift
Muscle3.6 Score Comparison
GP100 versus shift
ProbCons Score Comparison
GP100 versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
fM
shift
GP100 score

Figure B.2

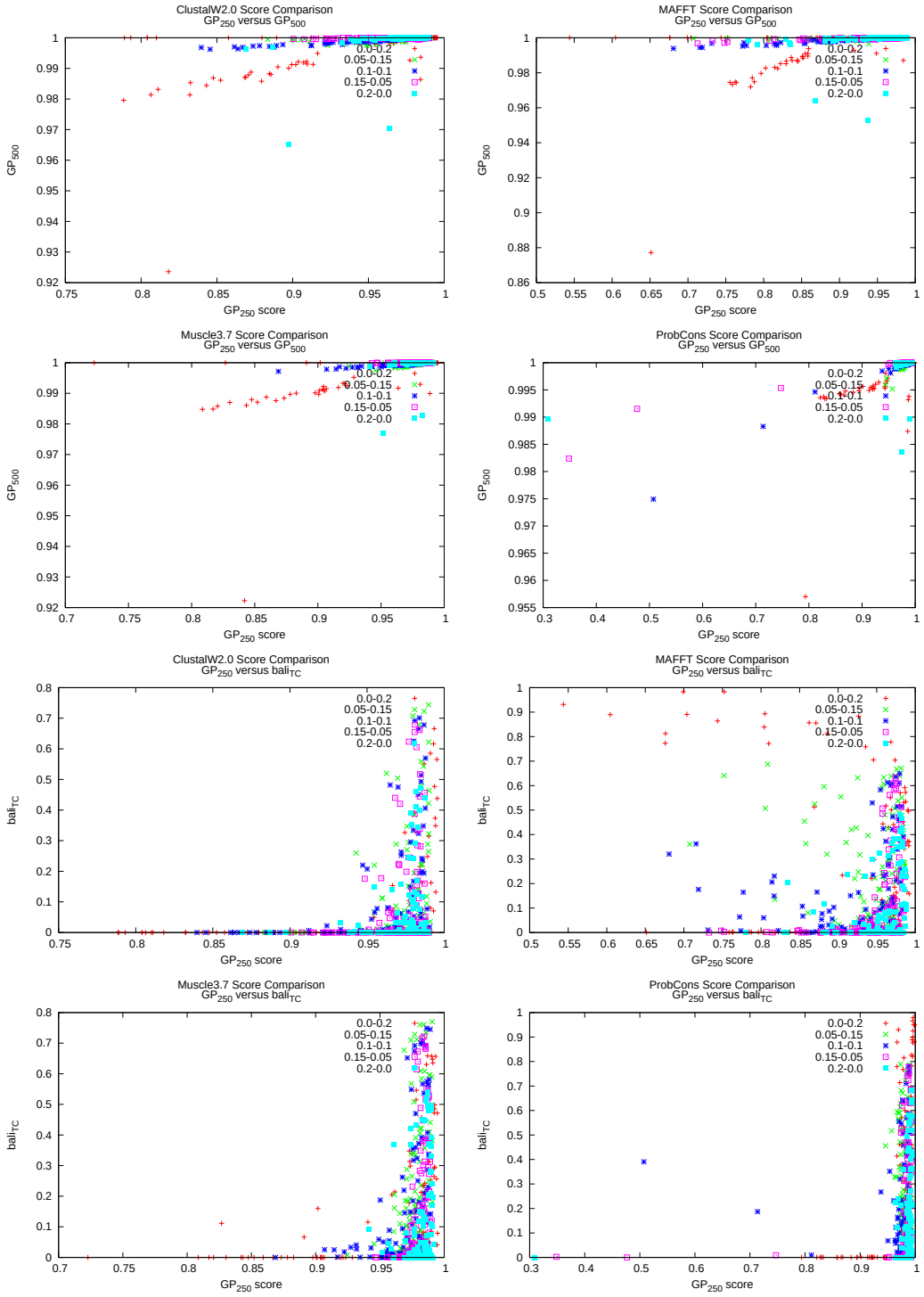

ClustalW2.0 Score Comparison
GP250 versus GP500
MAFFT Score Comparison
GP250 versus GP500
Muscle3.7 Score Comparison
GP250 versus GP500
ProbCons Score Comparison
GP250 versus GP500
ClustalW2.0 Score Comparison
GP250 versus baliTC
MAFFT Score Comparison
GP250 versus baliTC
Muscle3.7 Score Comparison
GP250 versus baliTC
ProbCons Score Comparison
GP250 versus baliTC
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
GP500
baliTC
GP250 score

Figure B.2

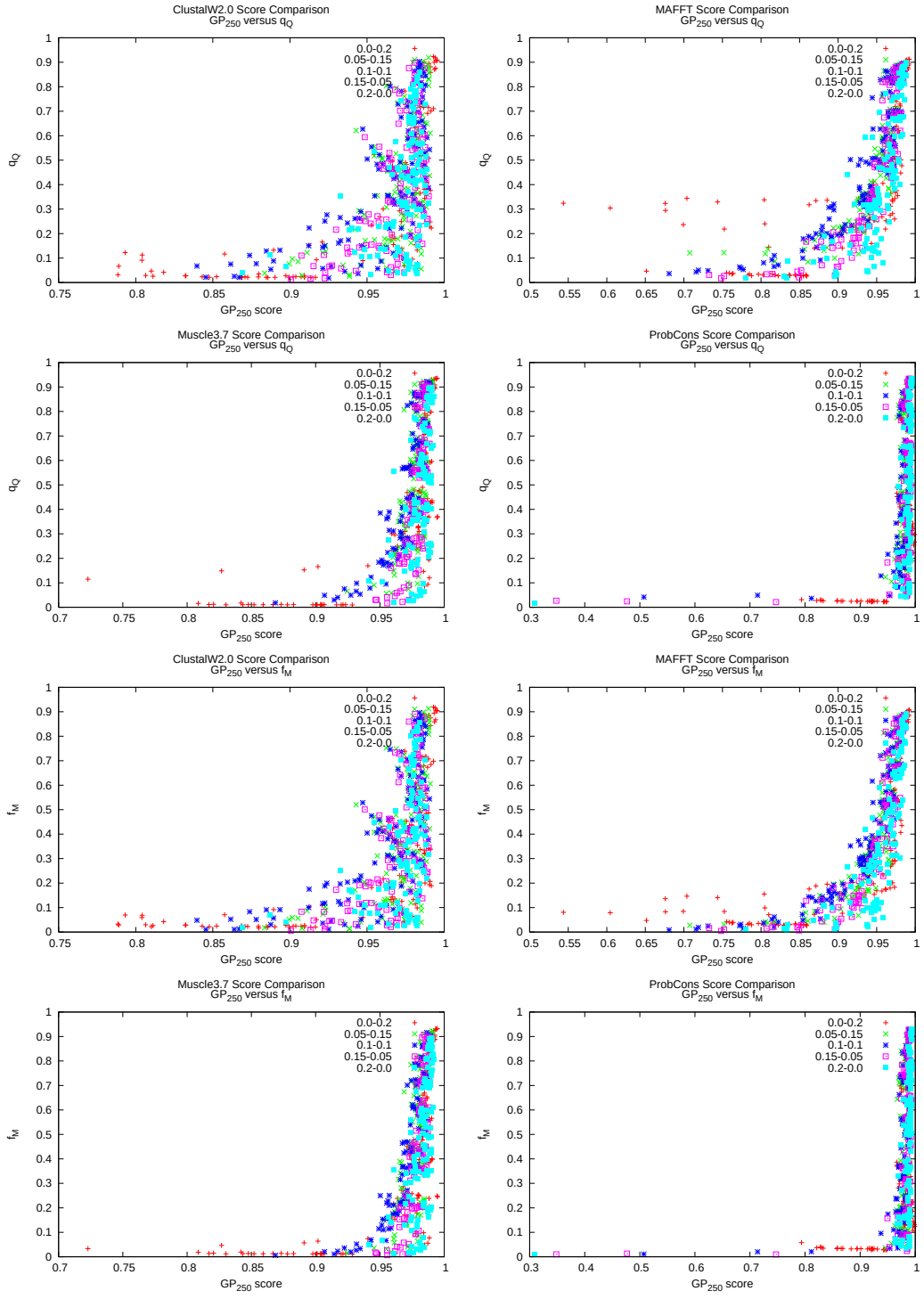

ClustalW2.0 Score Comparison
GP250 versus qQ
MAFFT Score Comparison
GP250 versus qQ
Muscle3.7 Score Comparison
GP250 versus qQ
ProbCons Score Comparison
GP250 versus qQ
ClustalW2.0 Score Comparison
GP250 versus fM
MAFFT Score Comparison
GP250 versus fM
Muscle3.7 Score Comparison
GP250 versus fM
ProbCons Score Comparison
GP250 versus fM
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
qQ
fM
GP250 score

Figure B.2

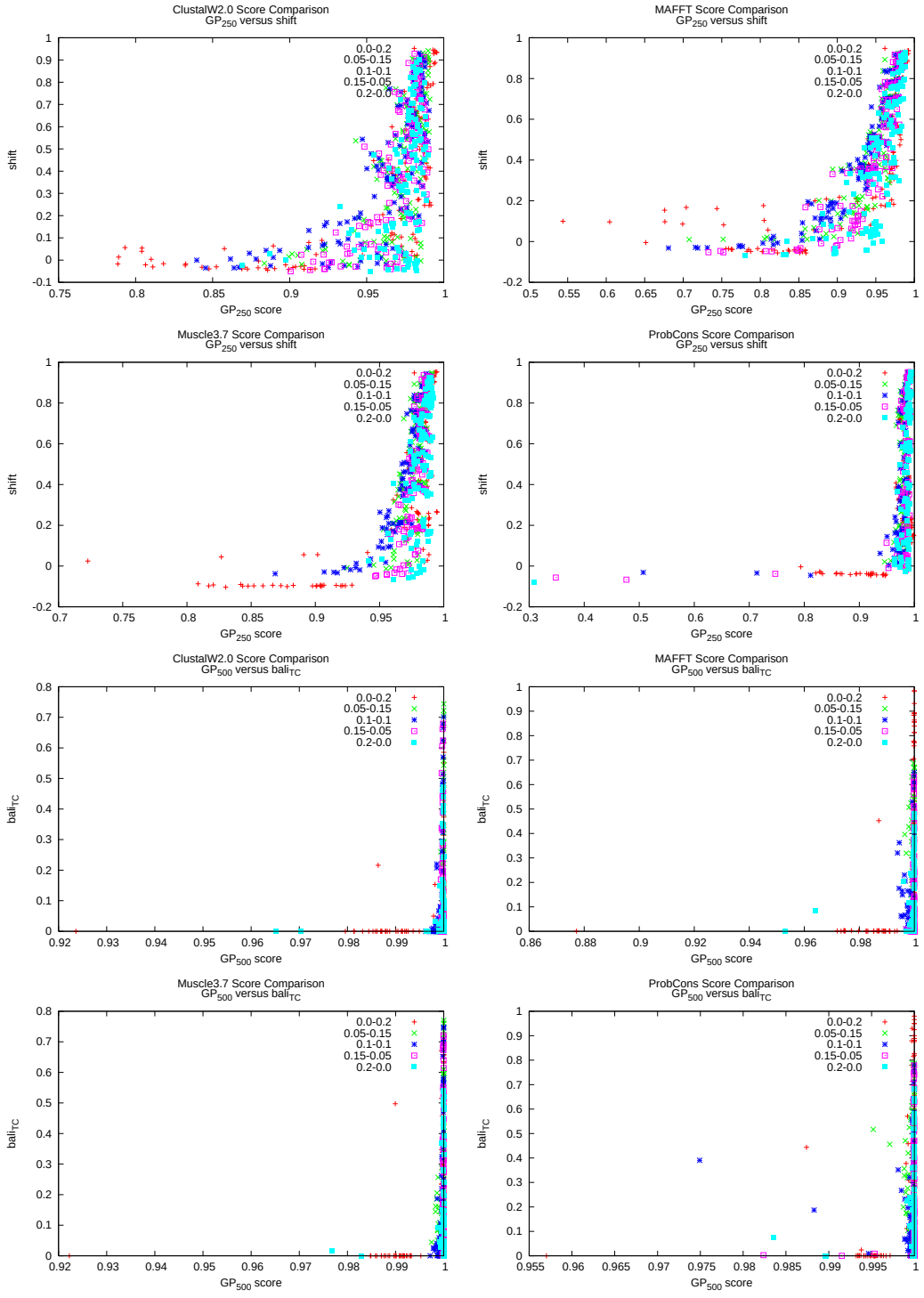

ClustalW2.0 Score Comparison
GP250 versus shift
MAFFT Score Comparison
GP250 versus shift
Muscle3.7 Score Comparison
GP250 versus shift
ProbCons Score Comparison
GP250 versus shift
ClustalW2.0 Score Comparison
GP500 versus balTC
MAFFT Score Comparison
GP500 versus balTC
Muscle3.7 Score Comparison
GP500 versus balTC
ProbCons Score Comparison
GP500 versus balTC
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
shift
balTC
GP250 score
GP500 score

Figure B.2

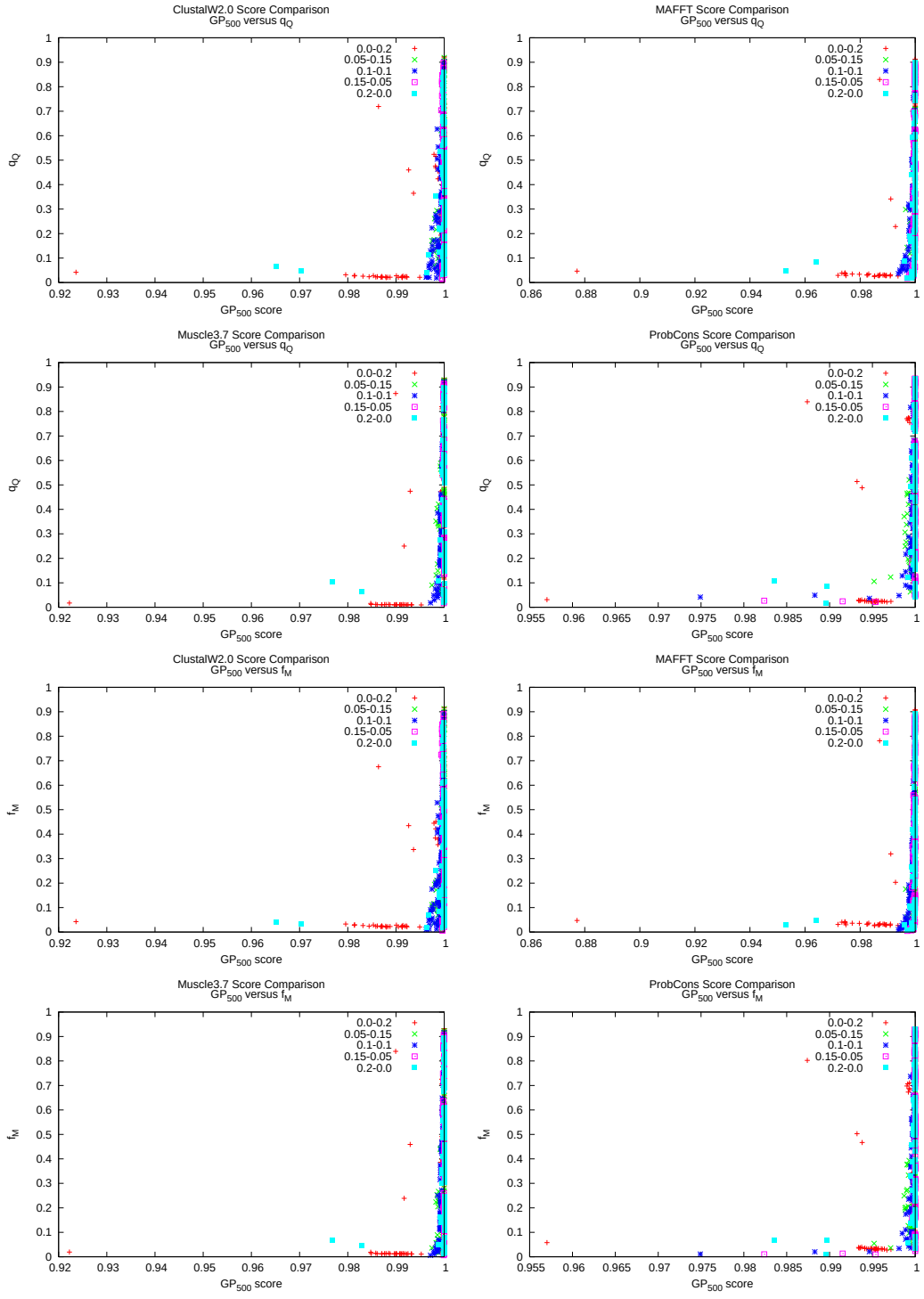

ClustalW2.0 Score Comparison
GP500 versus qQ
MAFFT Score Comparison
GP500 versus qQ
Muscle3.7 Score Comparison
GP500 versus qQ
ProbCons Score Comparison
GP500 versus qQ
ClustalW2.0 Score Comparison
GP500 versus fM
MAFFT Score Comparison
GP500 versus fM
Muscle3.7 Score Comparison
GP500 versus fM
ProbCons Score Comparison
GP500 versus fM
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
qQ
fM
GP500 score

Figure B.2

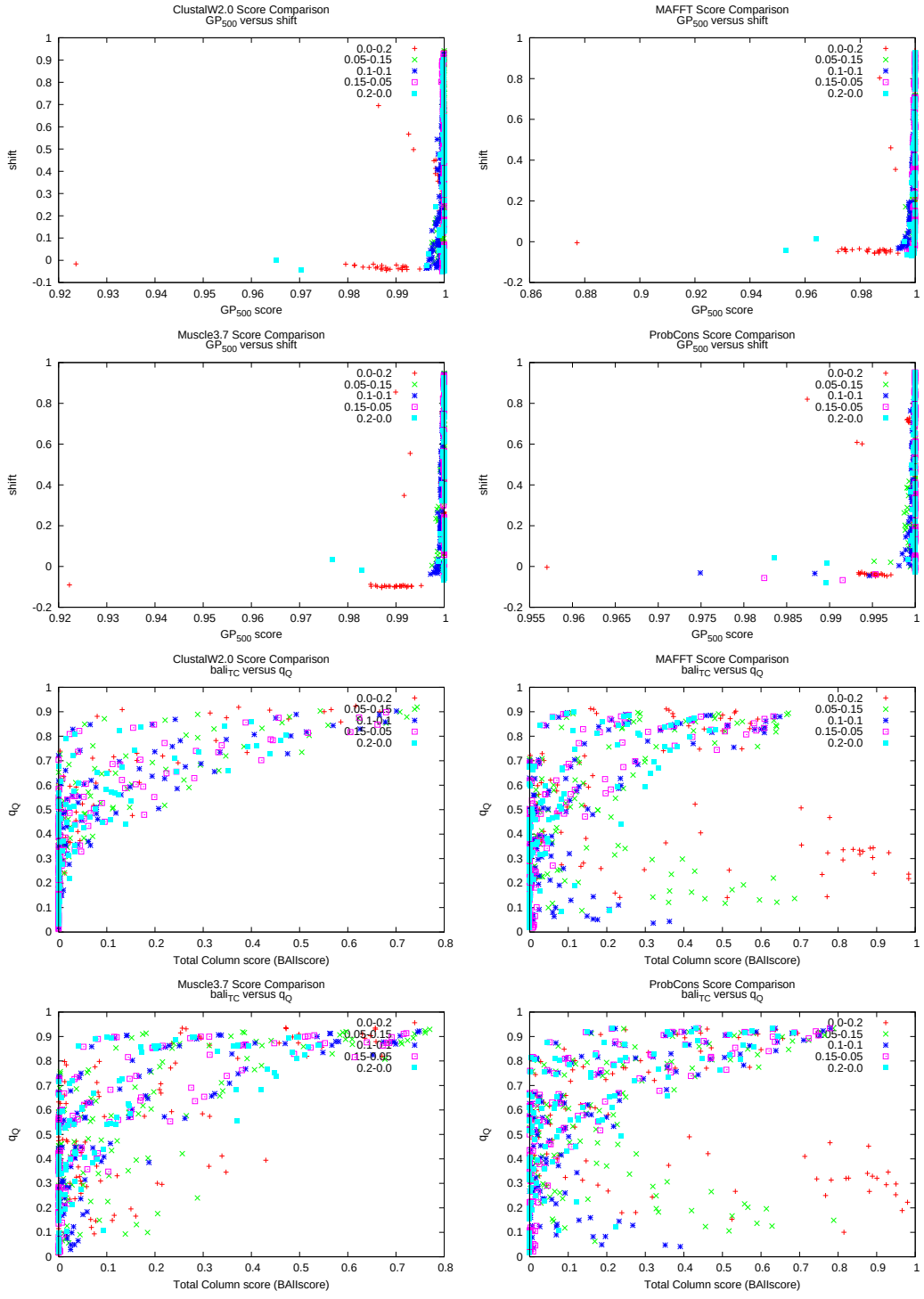

ClustalW2.0 Score Comparison
GP500 versus shift
MAFFT Score Comparison
GP500 versus shift
Muscle3.7 Score Comparison
GP500 versus shift
ProbCons Score Comparison
GP500 versus shift
ClustalW2.0 Score Comparison
balitc versus qQ
MAFFT Score Comparison
balitc versus qQ
Muscle3.7 Score Comparison
balitc versus qQ
ProbCons Score Comparison
balitc versus qQ
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
shift
GP500 score
qQ
Total Column score (BAIIscore)

Figure B.2

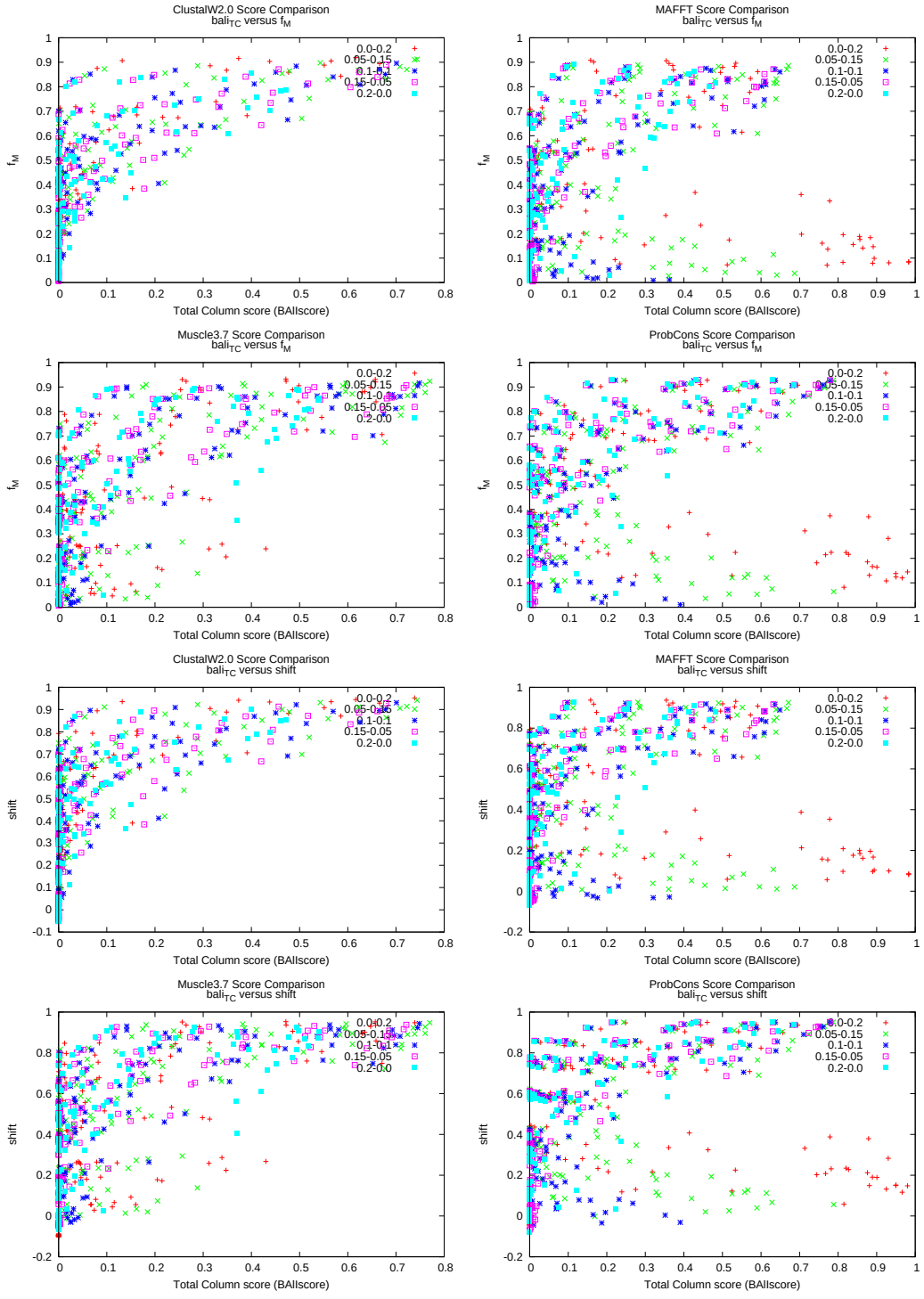

ClustalW2.0 Score Comparison
bali_TC versus f_M
MAFFT Score Comparison
bali_TC versus f_M
Muscle3.7 Score Comparison
bali_TC versus f_M
ProbCons Score Comparison
bali_TC versus f_M
ClustalW2.0 Score Comparison
bali_TC versus shift
MAFFT Score Comparison
bali_TC versus shift
Muscle3.7 Score Comparison
bali_TC versus shift
ProbCons Score Comparison
bali_TC versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
f_M
shift
Total Column score (BAIIscore)

Figure B.2

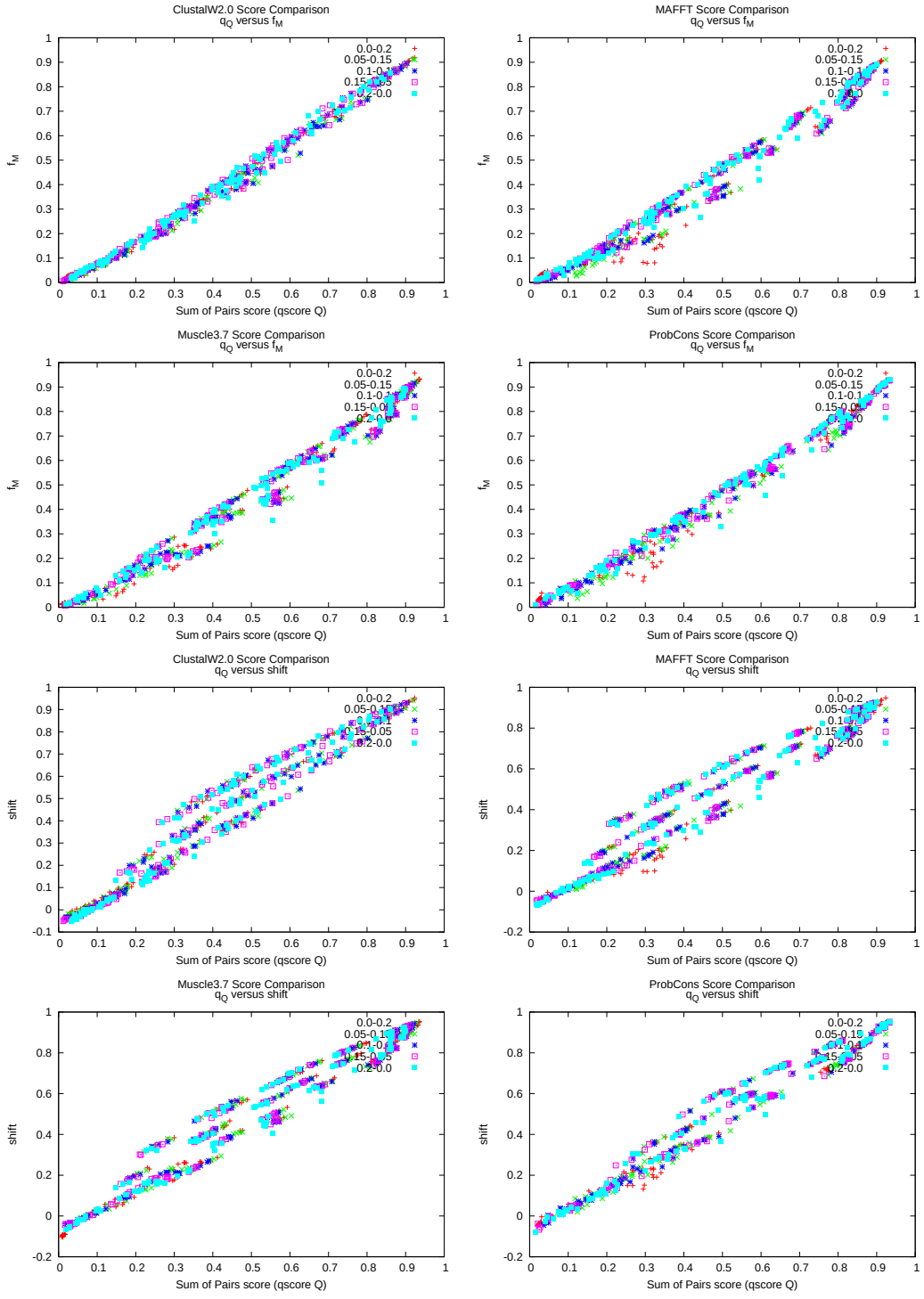

ClustalW2.0 Score Comparison
$q_Q$ versus $f_M$
MAFFT Score Comparison
$q_Q$ versus $f_M$
Muscle3.7 Score Comparison
$q_Q$ versus $f_M$
ProbCons Score Comparison
$q_Q$ versus $f_M$
ClustalW2.0 Score Comparison
$q_Q$ versus shift
MAFFT Score Comparison
$q_Q$ versus shift
Muscle3.7 Score Comparison
$q_Q$ versus shift
ProbCons Score Comparison
$q_Q$ versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
$f_M$
shift
Sum of Pairs score (qscore Q)

Figure B.2

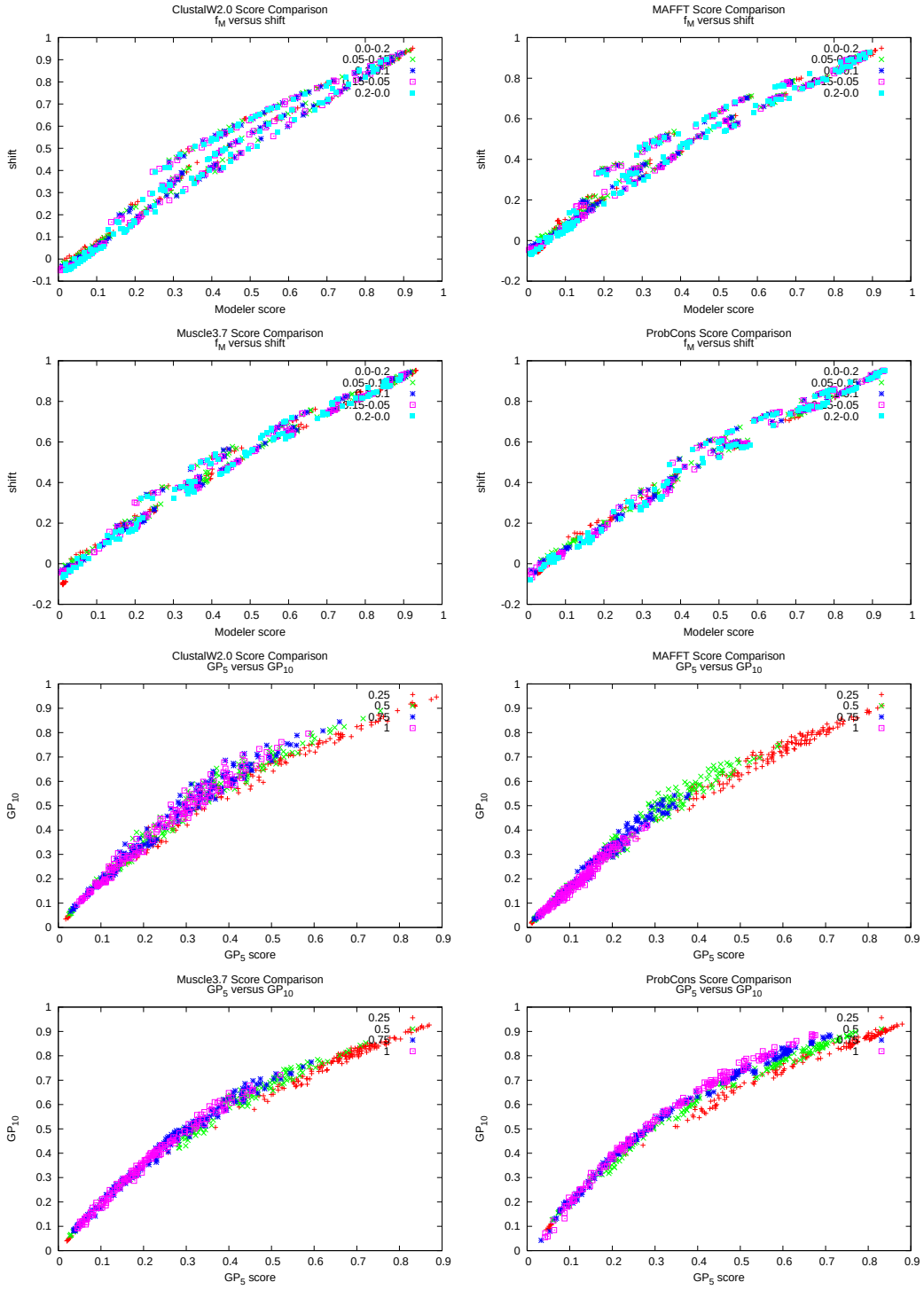

ClustalW2.0 Score Comparison
$f_M$ versus shift
MAFFT Score Comparison
$f_M$ versus shift
Muscle3.7 Score Comparison
$f_M$ versus shift
ProbCons Score Comparison
$f_M$ versus shift
0.0-0.2
0.05-0.15
0.1-0.1
0.15-0.05
0.2-0.0
shift
Modeler score
ClustalW2.0 Score Comparison
$GP_5$ versus $GP_{10}$
MAFFT Score Comparison
$GP_5$ versus $GP_{10}$
Muscle3.7 Score Comparison
$GP_5$ versus $GP_{10}$
ProbCons Score Comparison
$GP_5$ versus $GP_{10}$
0.25
0.5
0.75
1
$GP_{10}$
$GP_5$ score

Figure B.2

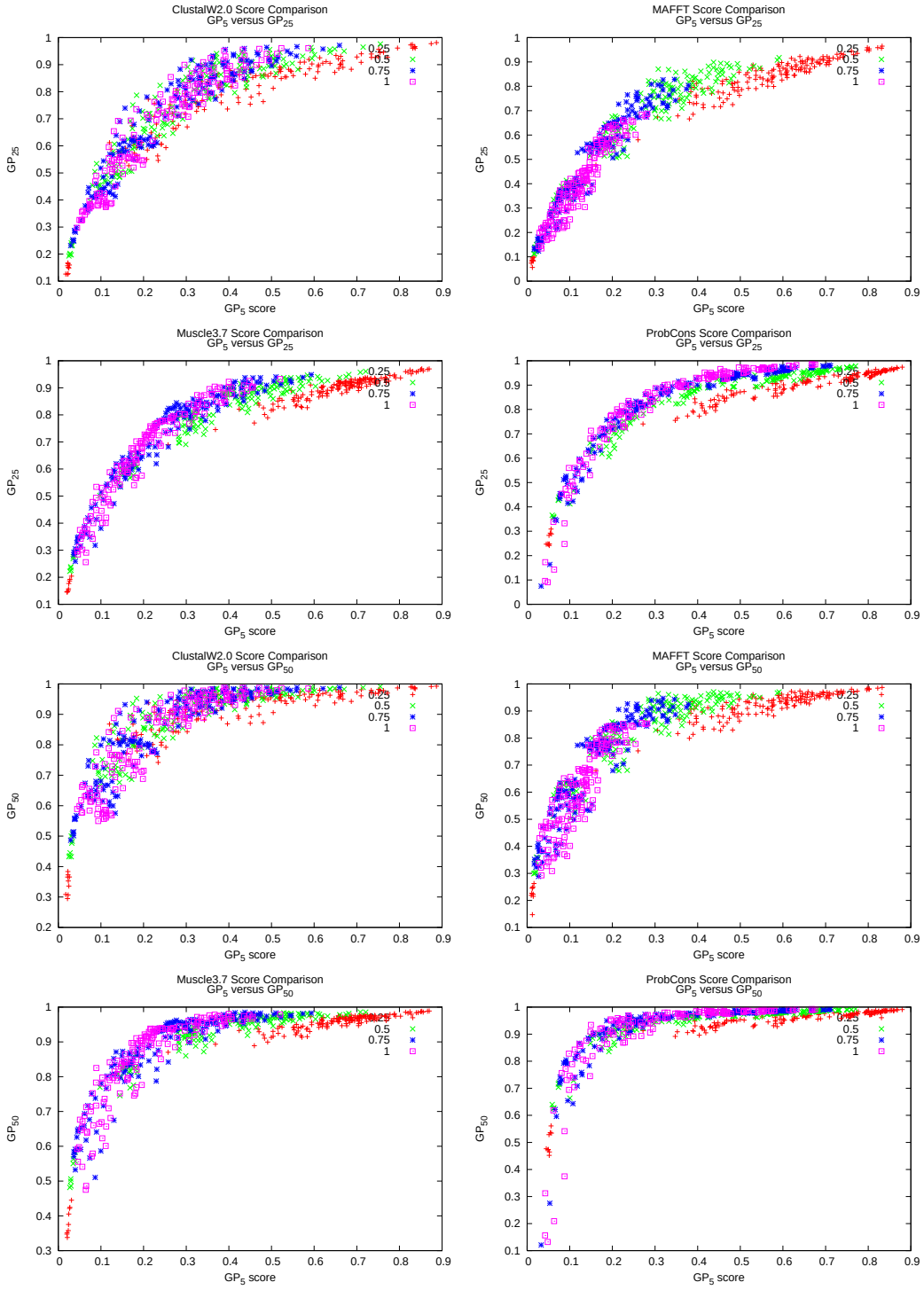

ClustalW2.0 Score Comparison
GP5 versus GP25
MAFFT Score Comparison
GP5 versus GP25
Muscle3.7 Score Comparison
GP5 versus GP25
ProbCons Score Comparison
GP5 versus GP25
ClustalW2.0 Score Comparison
GP5 versus GP50
MAFFT Score Comparison
GP5 versus GP50
Muscle3.7 Score Comparison
GP5 versus GP50
ProbCons Score Comparison
GP5 versus GP50
GP25
GP50
GP5 score
0.25
0.5
0.75
1

Figure B.2

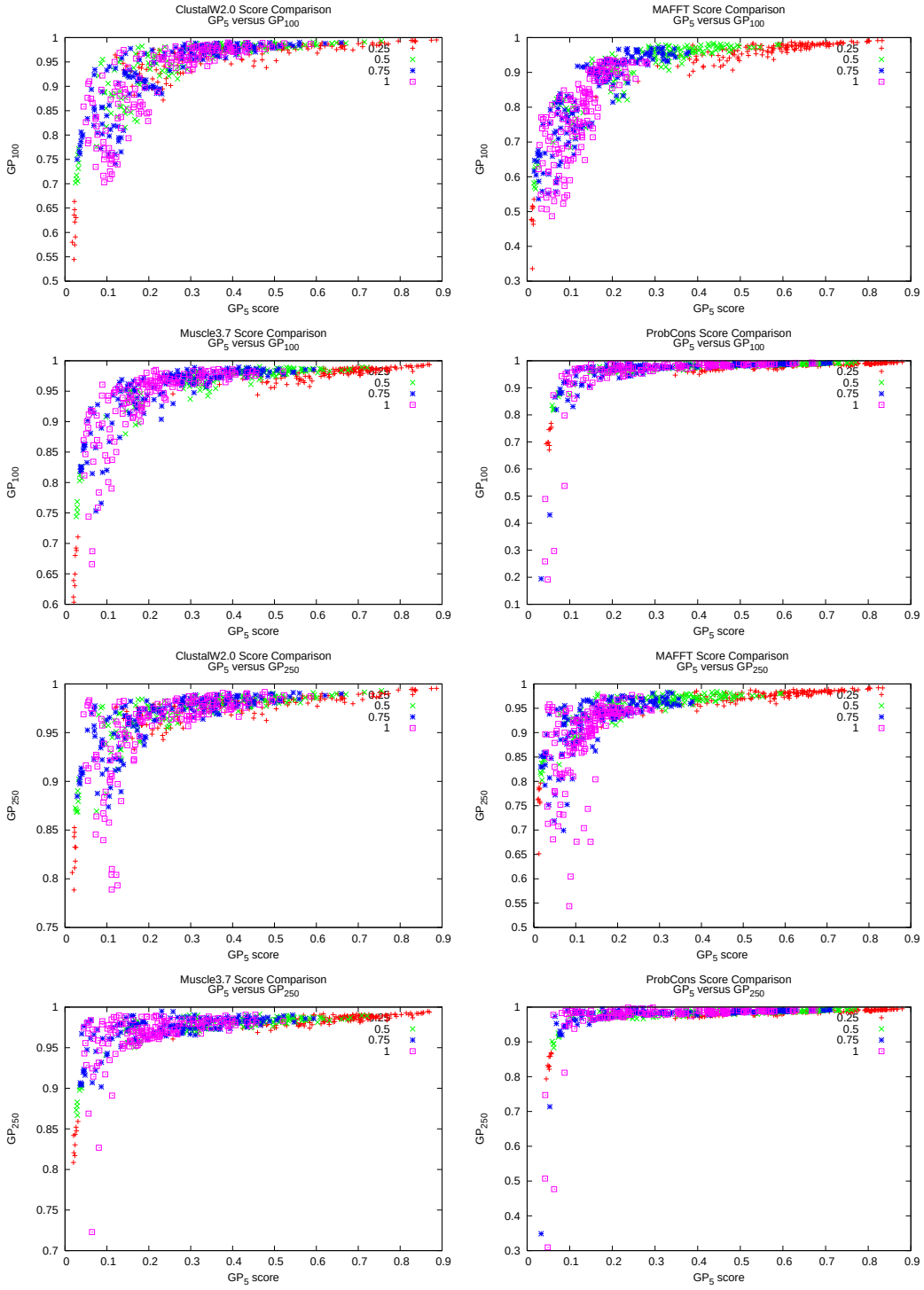

ClustalW2.0 Score Comparison
GP5 versus GP100
MAFFT Score Comparison
GP5 versus GP100
Muscle3.7 Score Comparison
GP5 versus GP100
ProbCons Score Comparison
GP5 versus GP100
ClustalW2.0 Score Comparison
GP5 versus GP250
MAFFT Score Comparison
GP5 versus GP250
Muscle3.7 Score Comparison
GP5 versus GP250
ProbCons Score Comparison
GP5 versus GP250
GP100
GP250
GP5 score
0.25
0.5
0.75
1

Figure B.2

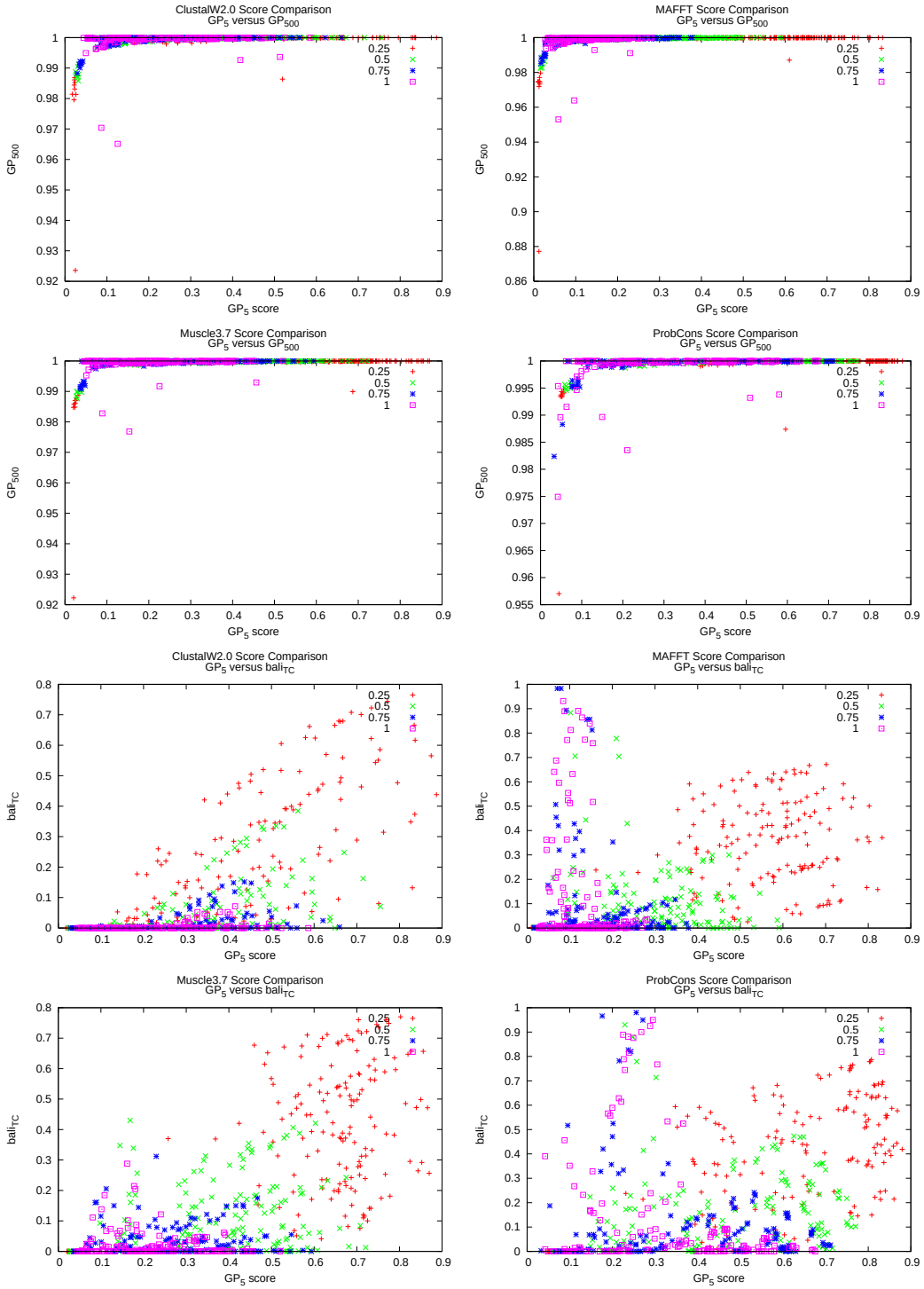

ClustalW2.0 Score Comparison
GP5 versus GP500
MAFFT Score Comparison
GP5 versus GP500
Muscle3.7 Score Comparison
GP5 versus GP500
ProbCons Score Comparison
GP5 versus GP500
ClustalW2.0 Score Comparison
GP5 versus balitc
MAFFT Score Comparison
GP5 versus balitc
Muscle3.7 Score Comparison
GP5 versus balitc
ProbCons Score Comparison
GP5 versus balitc
GP500
balitc
GP5 score
0.25
0.5
0.75
1

Figure B.2

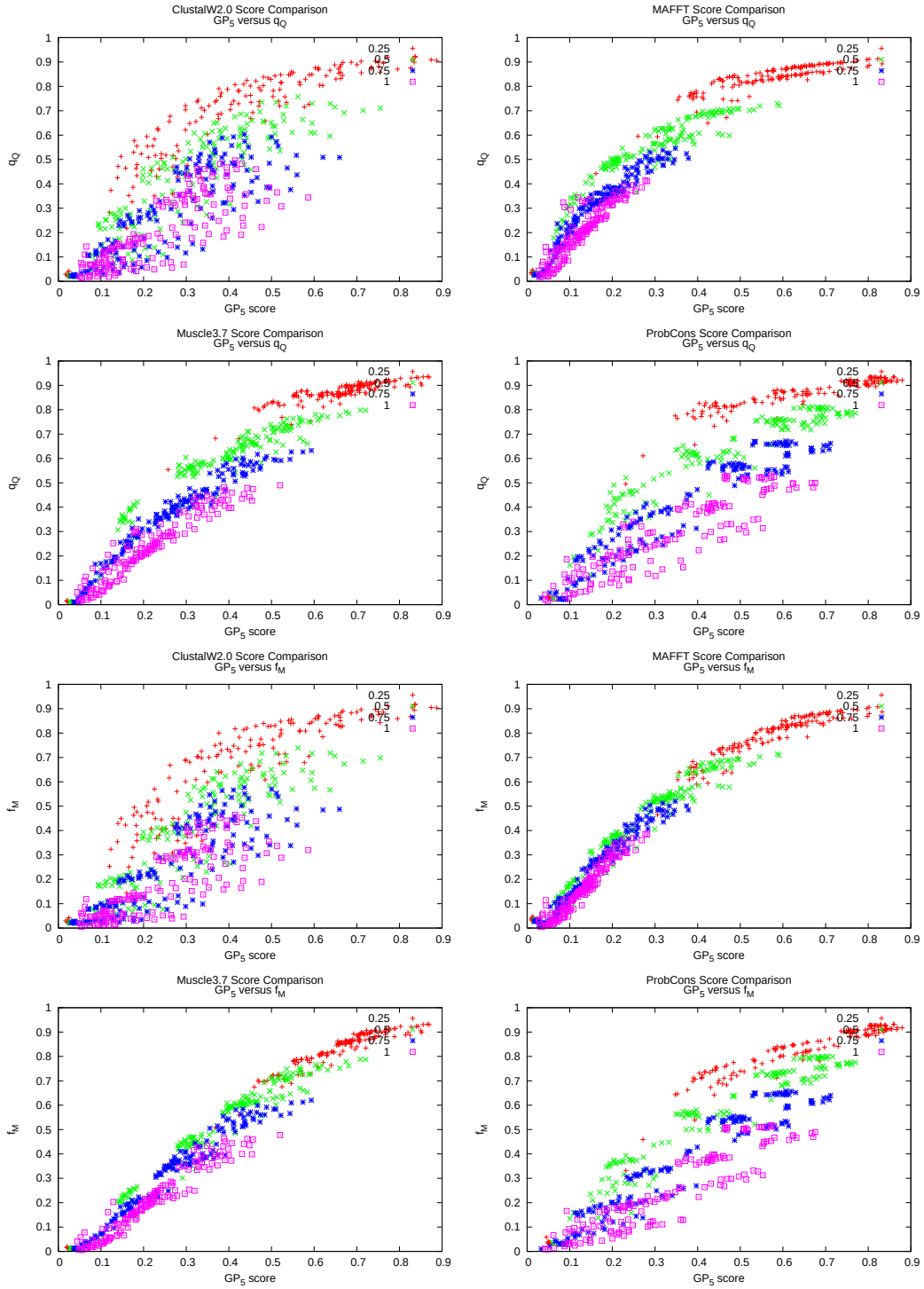

ClustalW2.0 Score Comparison
GP5 versus qQ
MAFFT Score Comparison
GP5 versus qQ
Muscle3.7 Score Comparison
GP5 versus qQ
ProbCons Score Comparison
GP5 versus qQ
ClustalW2.0 Score Comparison
GP5 versus fM
MAFFT Score Comparison
GP5 versus fM
Muscle3.7 Score Comparison
GP5 versus fM
ProbCons Score Comparison
GP5 versus fM
qQ
fM
GP5 score
0.25
0.5
0.75
1

Figure B.2

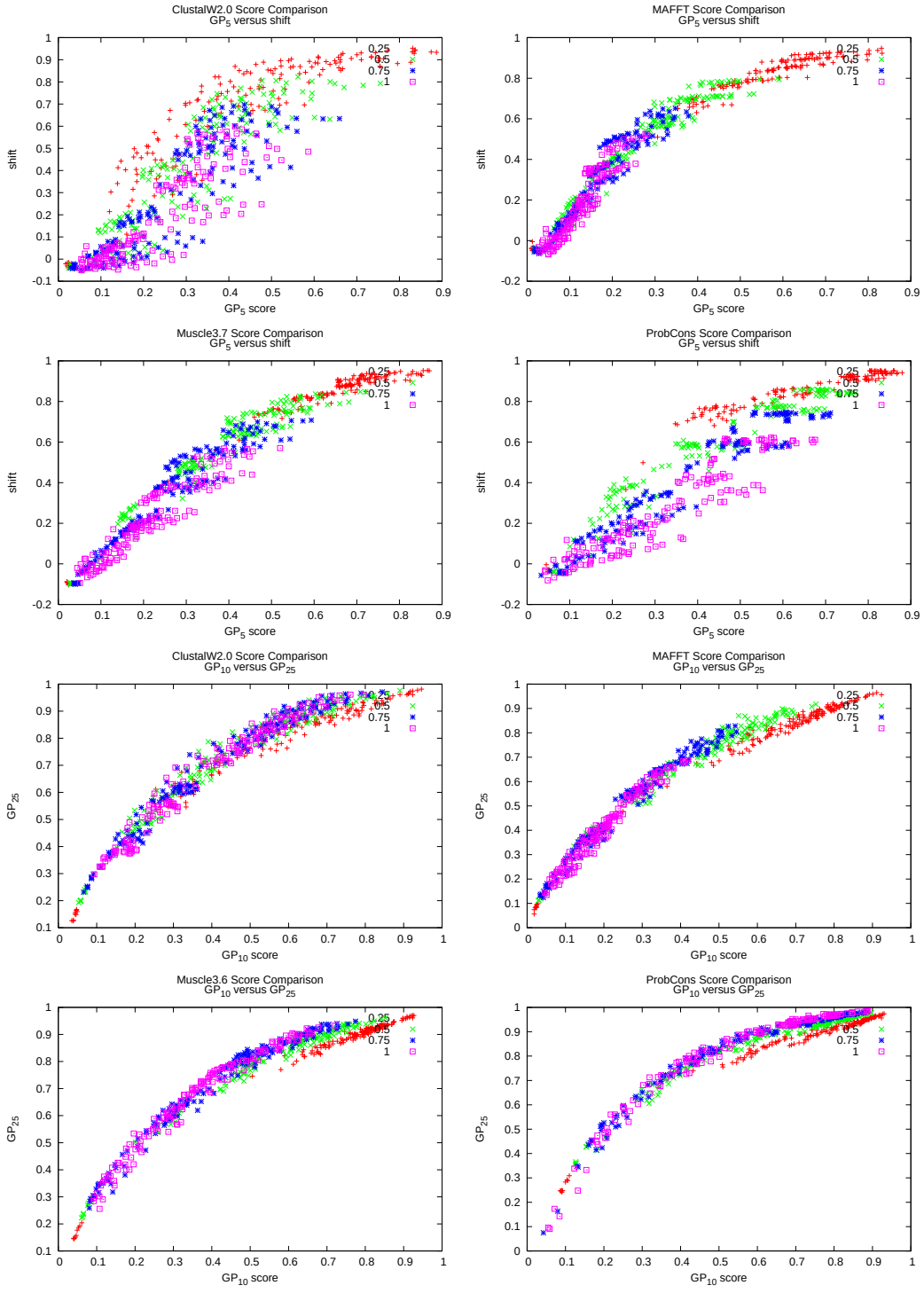

ClustalW2.0 Score Comparison
GP5 versus shift
MAFFT Score Comparison
GP5 versus shift
Muscle3.7 Score Comparison
GP5 versus shift
ProbCons Score Comparison
GP5 versus shift
ClustalW2.0 Score Comparison
GP10 versus GP25
MAFFT Score Comparison
GP10 versus GP25
Muscle3.6 Score Comparison
GP10 versus GP25
ProbCons Score Comparison
GP10 versus GP25
shift
GP5 score
GP25
GP10 score
0.25
0.5
0.75
1

Figure B.2

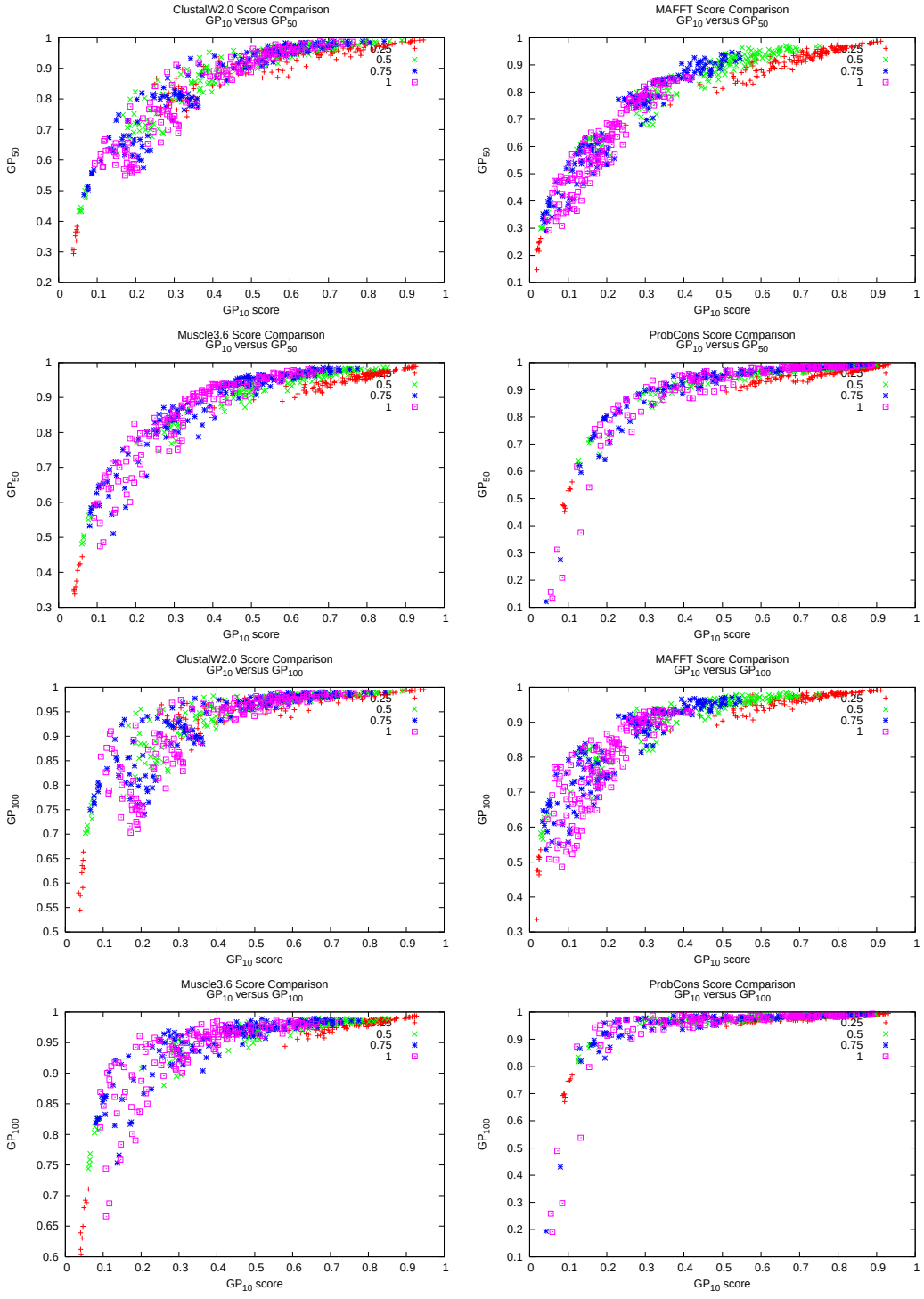

ClustalW2.0 Score Comparison
GP10 versus GP50
MAFFT Score Comparison
GP10 versus GP50
Muscle3.6 Score Comparison
GP10 versus GP50
ProbCons Score Comparison
GP10 versus GP50
ClustalW2.0 Score Comparison
GP10 versus GP100
MAFFT Score Comparison
GP10 versus GP100
Muscle3.6 Score Comparison
GP10 versus GP100
ProbCons Score Comparison
GP10 versus GP100
GP50
GP100
GP10 score
0.25
0.5
0.75
1

Figure B.2

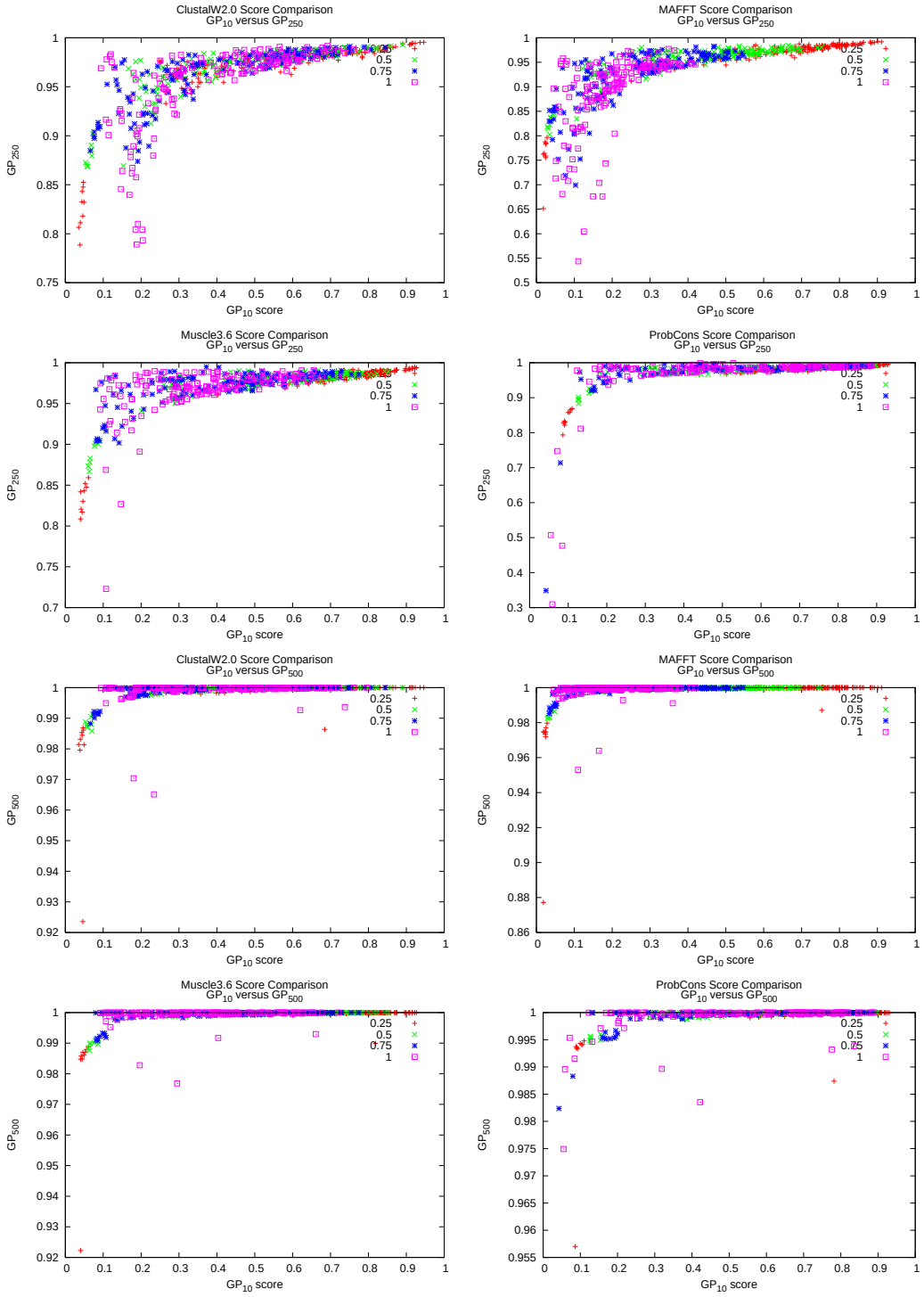

ClustalW2.0 Score Comparison
GP10 versus GP250
MAFFT Score Comparison
GP10 versus GP250
Muscle3.6 Score Comparison
GP10 versus GP250
ProbCons Score Comparison
GP10 versus GP250
ClustalW2.0 Score Comparison
GP10 versus GP500
MAFFT Score Comparison
GP10 versus GP500
Muscle3.6 Score Comparison
GP10 versus GP500
ProbCons Score Comparison
GP10 versus GP500
GP250
GP500
GP10 score
0.25
0.5
0.75
1

Figure B.2

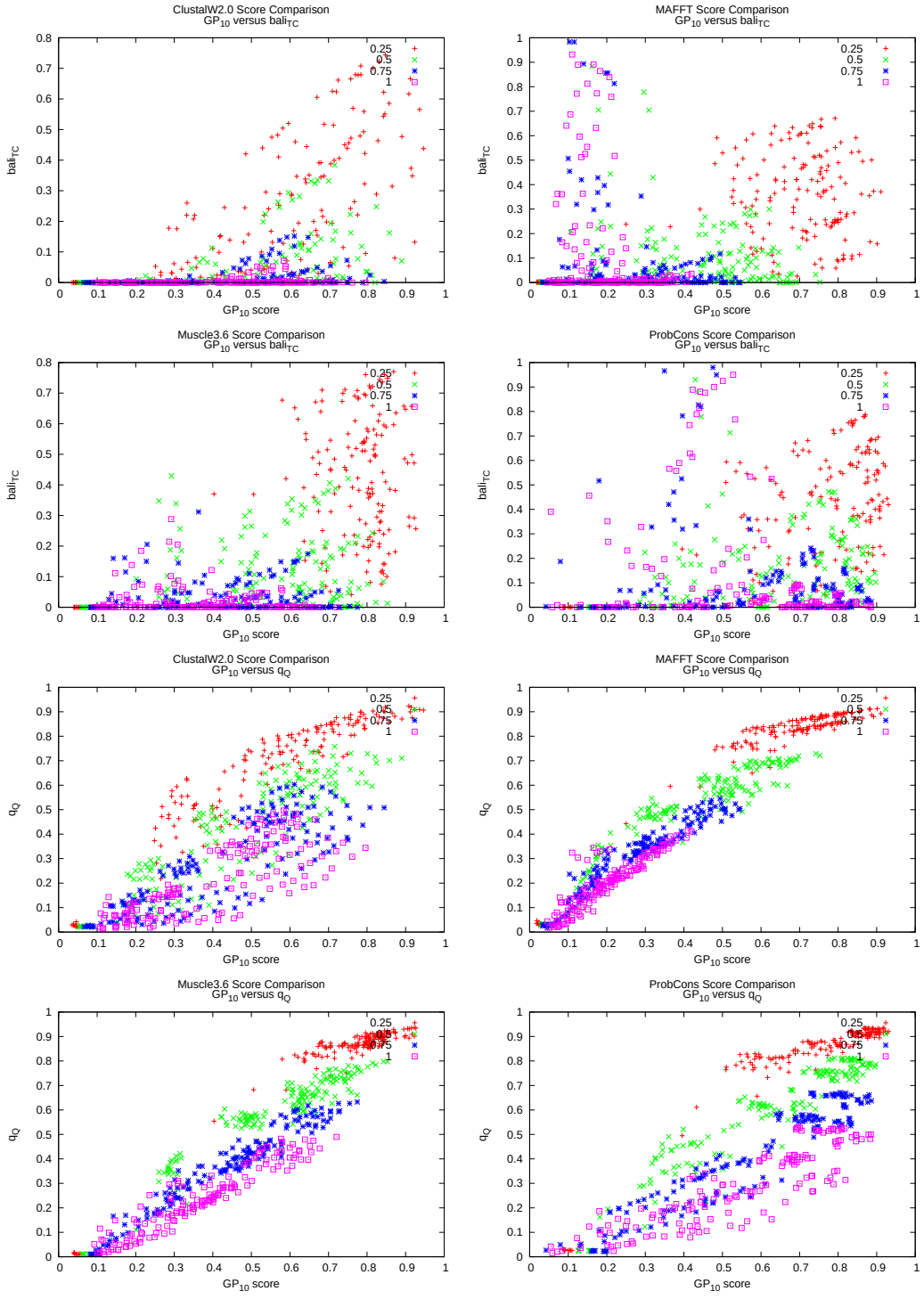

ClustalW2.0 Score Comparison
GP10 versus baliTC
MAFFT Score Comparison
GP10 versus baliTC
Muscle3.6 Score Comparison
GP10 versus baliTC
ProbCons Score Comparison
GP10 versus baliTC
ClustalW2.0 Score Comparison
GP10 versus qQ
MAFFT Score Comparison
GP10 versus qQ
Muscle3.6 Score Comparison
GP10 versus qQ
ProbCons Score Comparison
GP10 versus qQ
baliTC
qQ
GP10 score
0.25
0.5
0.75
1

Figure B.2

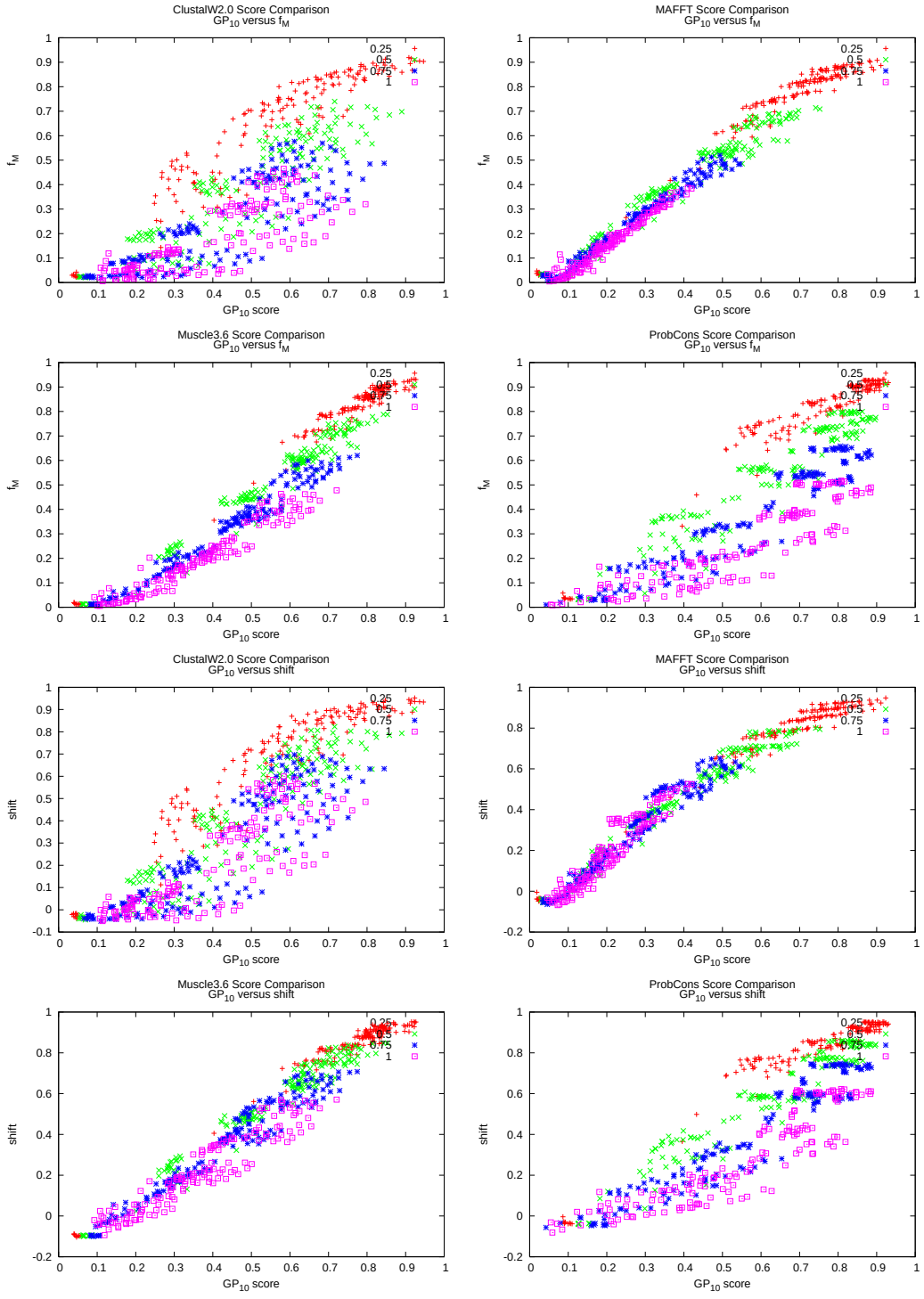

ClustalW2.0 Score Comparison
GP10 versus fM
MAFFT Score Comparison
GP10 versus fM
Muscle3.6 Score Comparison
GP10 versus fM
ProbCons Score Comparison
GP10 versus fM
ClustalW2.0 Score Comparison
GP10 versus shift
MAFFT Score Comparison
GP10 versus shift
Muscle3.6 Score Comparison
GP10 versus shift
ProbCons Score Comparison
GP10 versus shift
fM
shift
GP10 score
0.25
0.5
0.75
1

Figure B.2

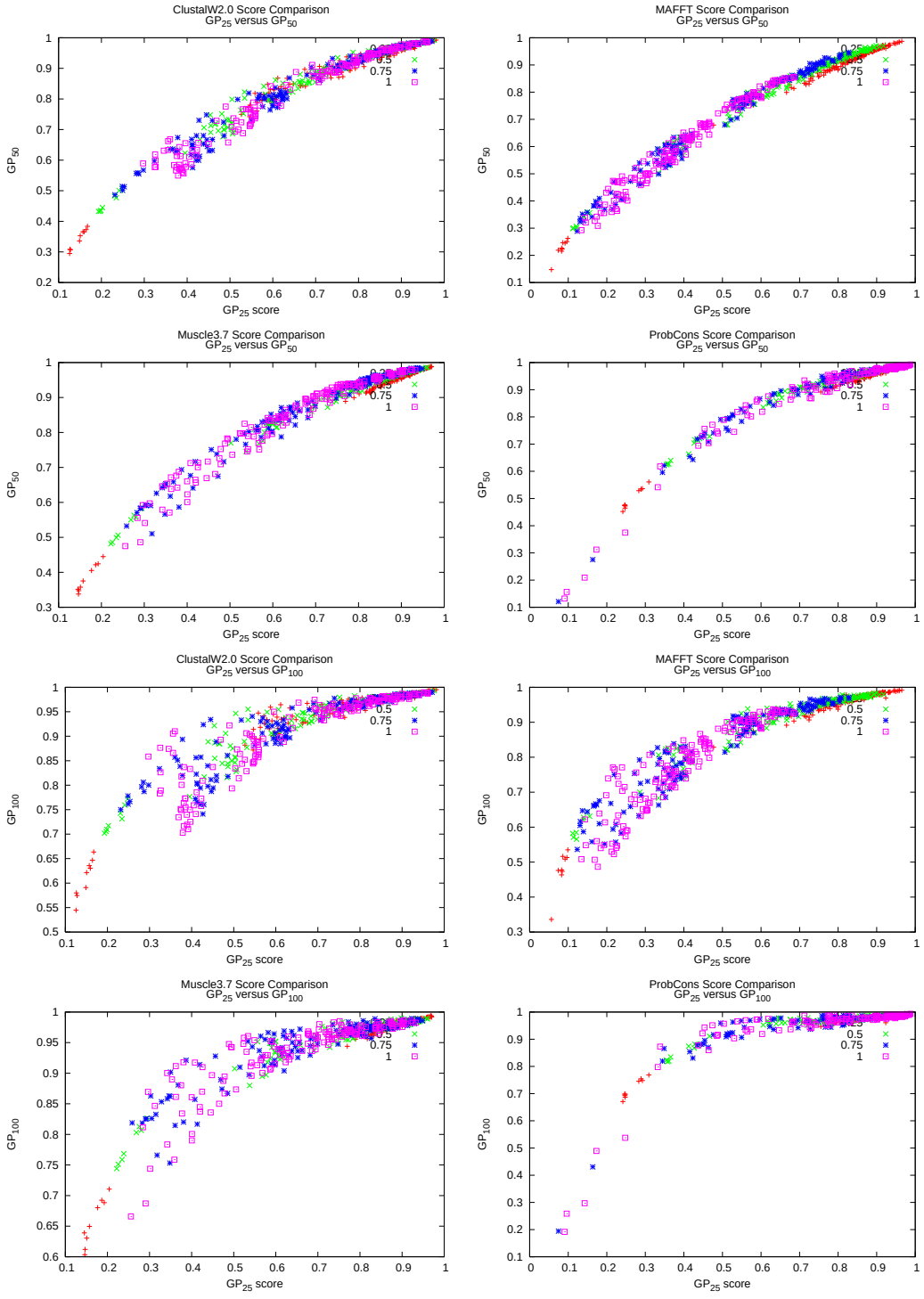

ClustalW2.0 Score Comparison
GP25 versus GP50
MAFFT Score Comparison
GP25 versus GP50
Muscle3.7 Score Comparison
GP25 versus GP50
ProbCons Score Comparison
GP25 versus GP50
ClustalW2.0 Score Comparison
GP25 versus GP100
MAFFT Score Comparison
GP25 versus GP100
Muscle3.7 Score Comparison
GP25 versus GP100
ProbCons Score Comparison
GP25 versus GP100
GP50
GP100
GP25 score
0.25
0.5
0.75
1

Figure B.2

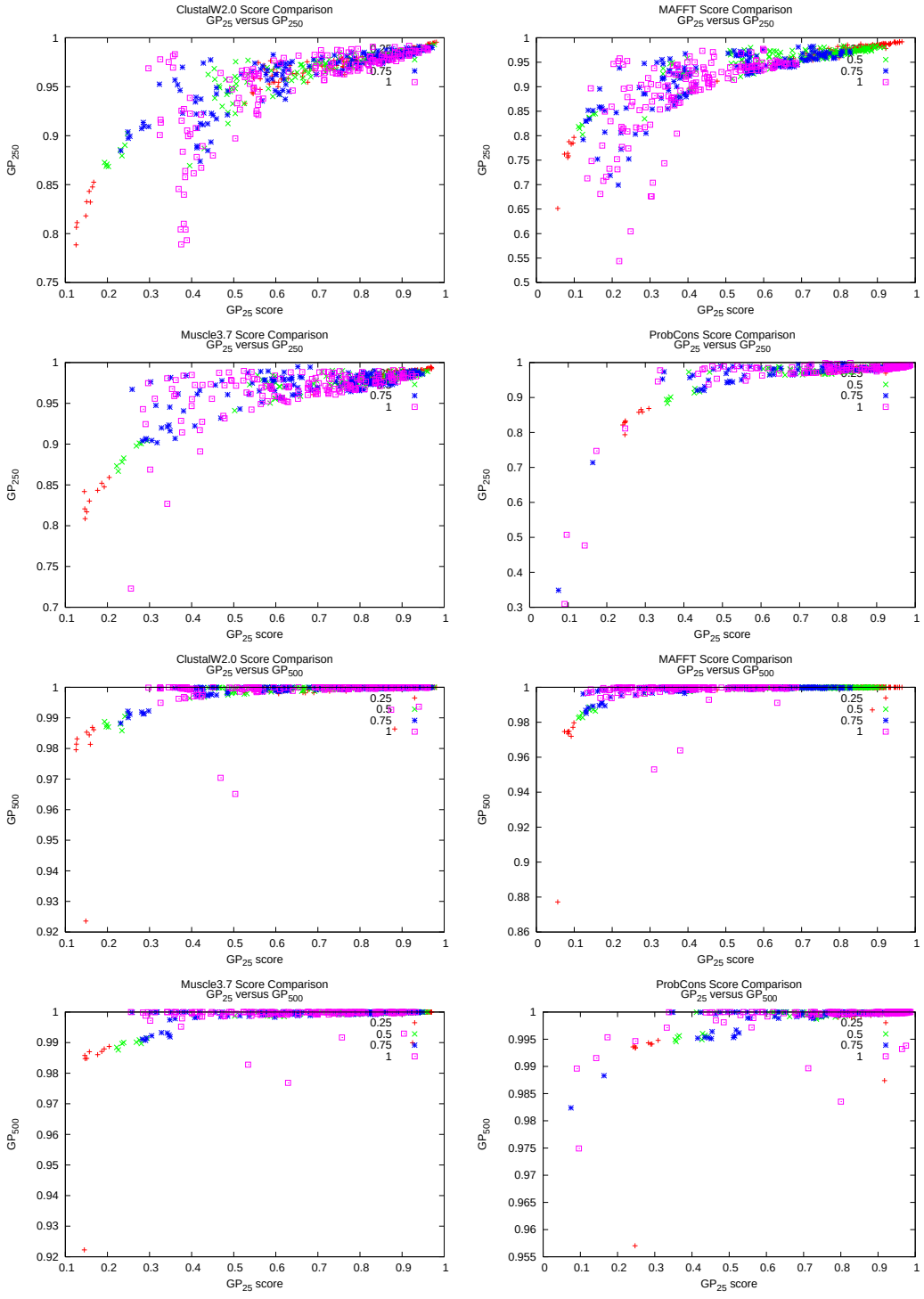

ClustalW2.0 Score Comparison
GP25 versus GP250
MAFFT Score Comparison
GP25 versus GP250
Muscle3.7 Score Comparison
GP25 versus GP250
ProbCons Score Comparison
GP25 versus GP250
ClustalW2.0 Score Comparison
GP25 versus GP500
MAFFT Score Comparison
GP25 versus GP500
Muscle3.7 Score Comparison
GP25 versus GP500
ProbCons Score Comparison
GP25 versus GP500
GP250
GP500
GP25 score
0.25
0.5
0.75
1

Figure B.2

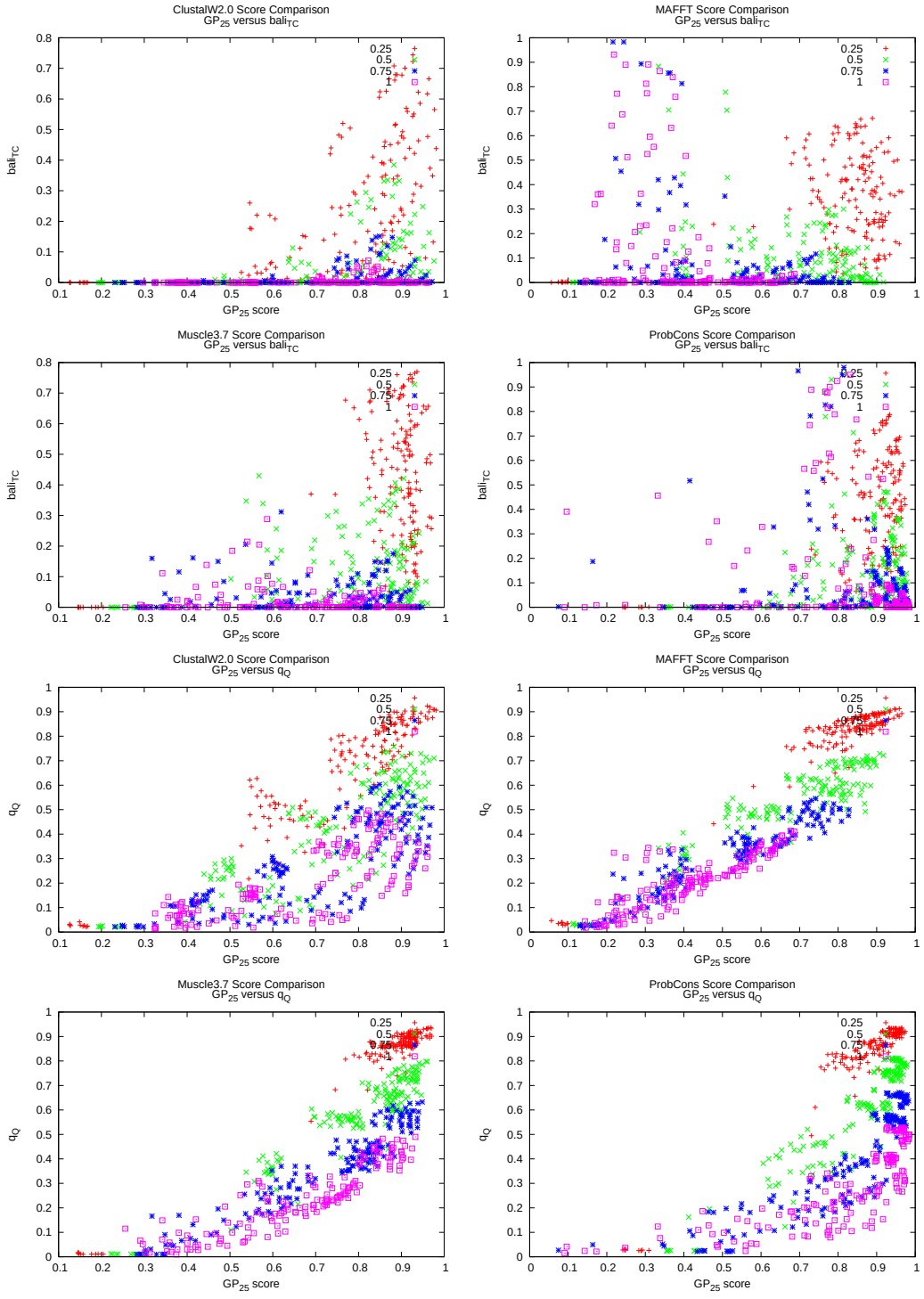

ClustalW2.0 Score Comparison
GP25 versus baliTC
MAFFT Score Comparison
GP25 versus baliTC
Muscle3.7 Score Comparison
GP25 versus baliTC
ProbCons Score Comparison
GP25 versus baliTC
ClustalW2.0 Score Comparison
GP25 versus qQ
MAFFT Score Comparison
GP25 versus qQ
Muscle3.7 Score Comparison
GP25 versus qQ
ProbCons Score Comparison
GP25 versus qQ
baliTC
qQ
GP25 score
0.25
0.5
0.75
1

Figure B.2

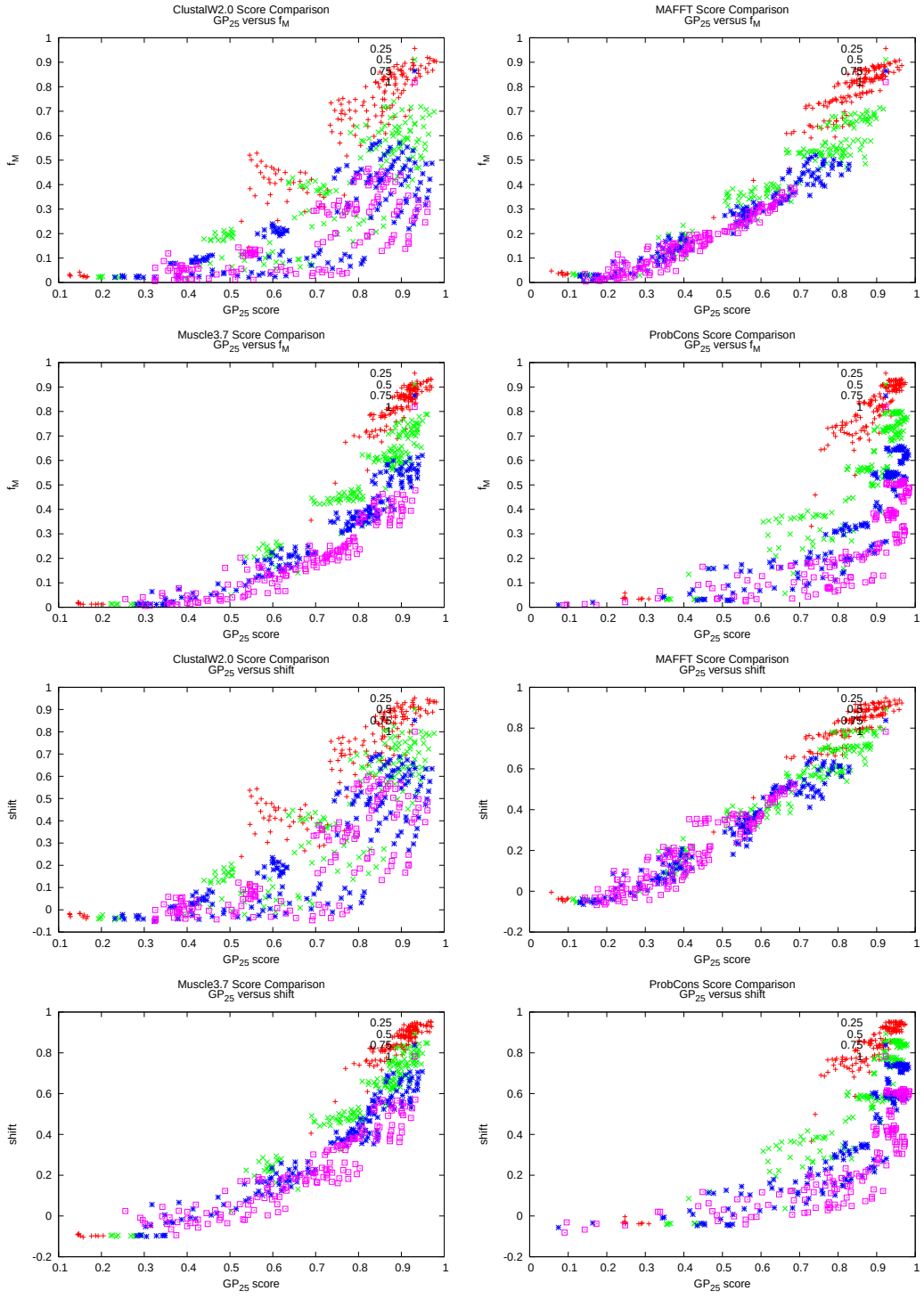

ClustalW2.0 Score Comparison
GP25 versus fM
MAFFT Score Comparison
GP25 versus fM
Muscle3.7 Score Comparison
GP25 versus fM
ProbCons Score Comparison
GP25 versus fM
ClustalW2.0 Score Comparison
GP25 versus shift
MAFFT Score Comparison
GP25 versus shift
Muscle3.7 Score Comparison
GP25 versus shift
ProbCons Score Comparison
GP25 versus shift
fM
shift
GP25 score
0.25
0.5
0.75
1

Figure B.2

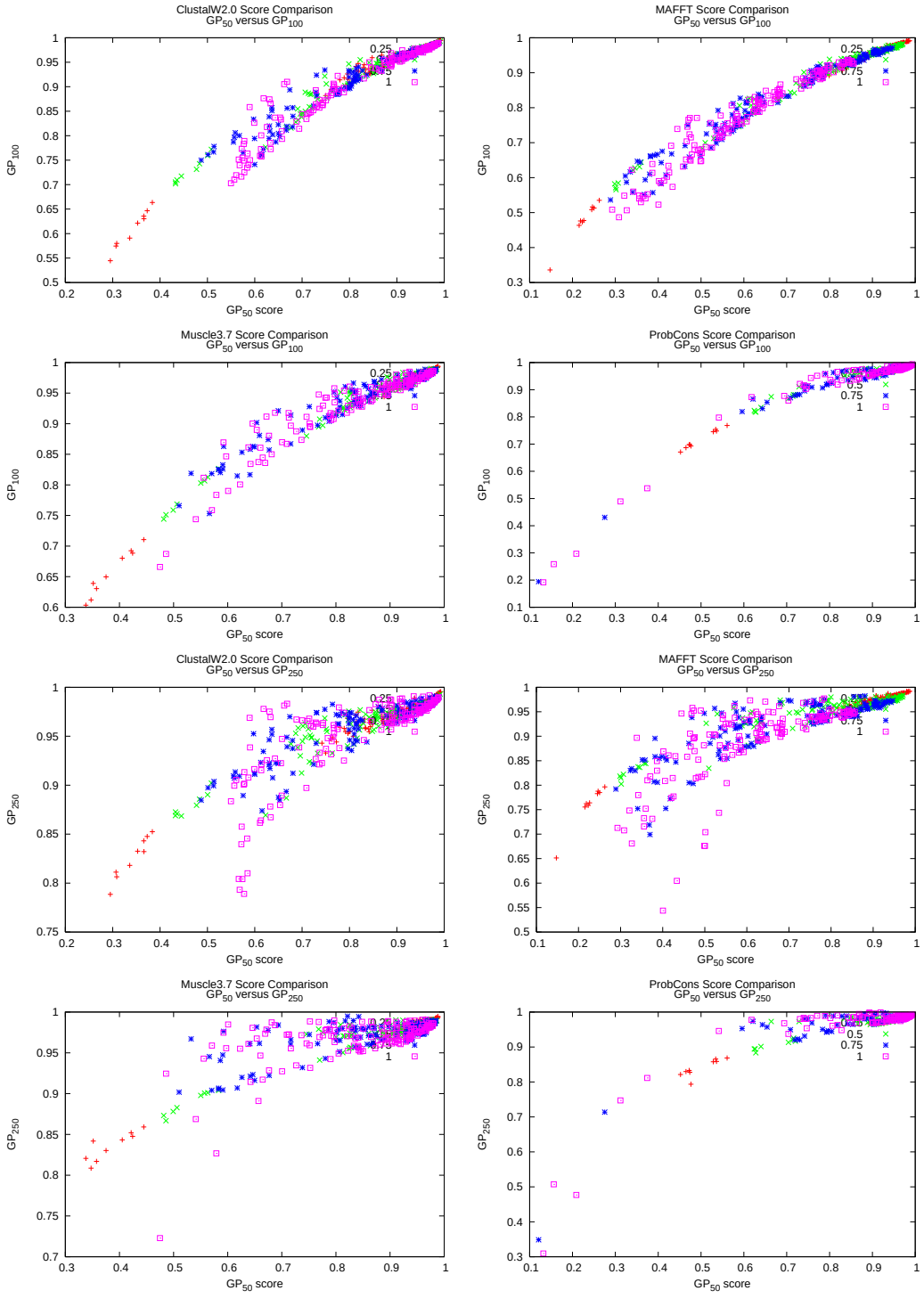

ClustalW2.0 Score Comparison
GP50 versus GP100
MAFFT Score Comparison
GP50 versus GP100
Muscle3.7 Score Comparison
GP50 versus GP100
ProbCons Score Comparison
GP50 versus GP100
ClustalW2.0 Score Comparison
GP50 versus GP250
MAFFT Score Comparison
GP50 versus GP250
Muscle3.7 Score Comparison
GP50 versus GP250
ProbCons Score Comparison
GP50 versus GP250
GP100
GP250
GP50 score
0.25
0.5
0.75
1

Figure B.2

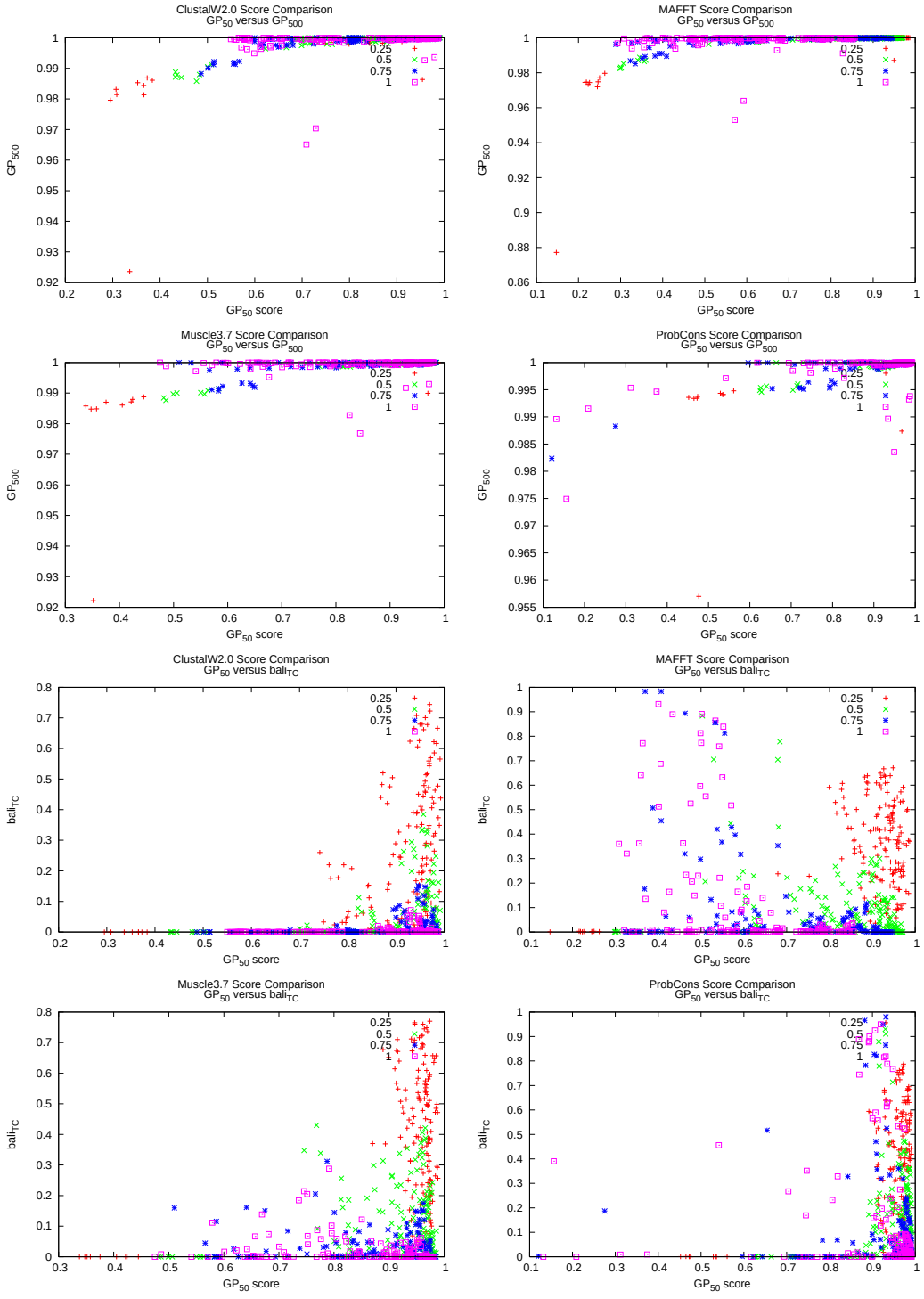

ClustalW2.0 Score Comparison
GP50 versus GP500
MAFFT Score Comparison
GP50 versus GP500
Muscle3.7 Score Comparison
GP50 versus GP500
ProbCons Score Comparison
GP50 versus GP500
ClustalW2.0 Score Comparison
GP50 versus baliTC
MAFFT Score Comparison
GP50 versus baliTC
Muscle3.7 Score Comparison
GP50 versus baliTC
ProbCons Score Comparison
GP50 versus baliTC
GP500
baliTC
GP50 score
0.25
0.5
0.75
1

Figure B.2

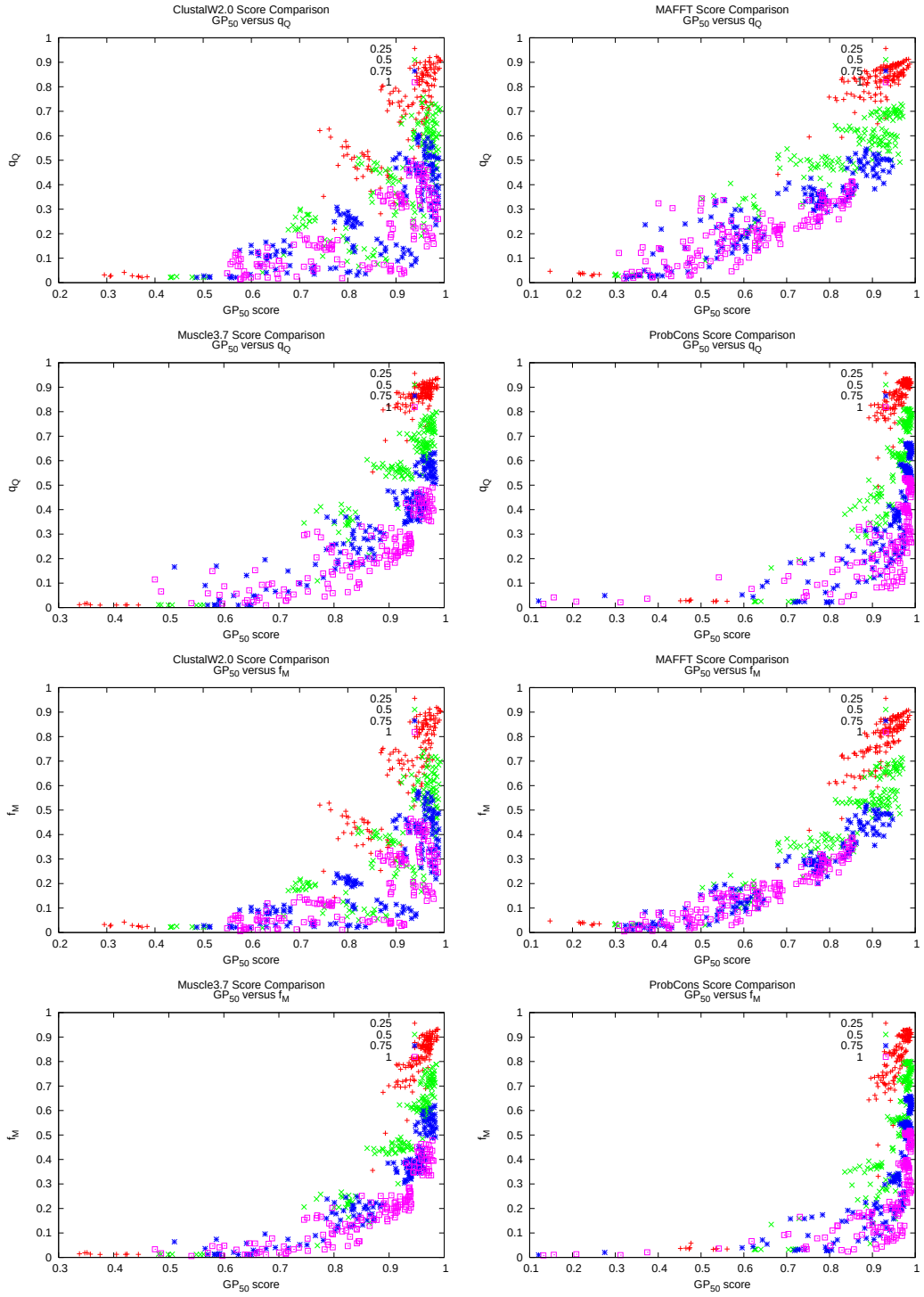

ClustalW2.0 Score Comparison
GP50 versus qQ
MAFFT Score Comparison
GP50 versus qQ
Muscle3.7 Score Comparison
GP50 versus qQ
ProbCons Score Comparison
GP50 versus qQ
ClustalW2.0 Score Comparison
GP50 versus fM
MAFFT Score Comparison
GP50 versus fM
Muscle3.7 Score Comparison
GP50 versus fM
ProbCons Score Comparison
GP50 versus fM
qQ
fM
GP50 score
0.25
0.5
0.75
1

Figure B.2

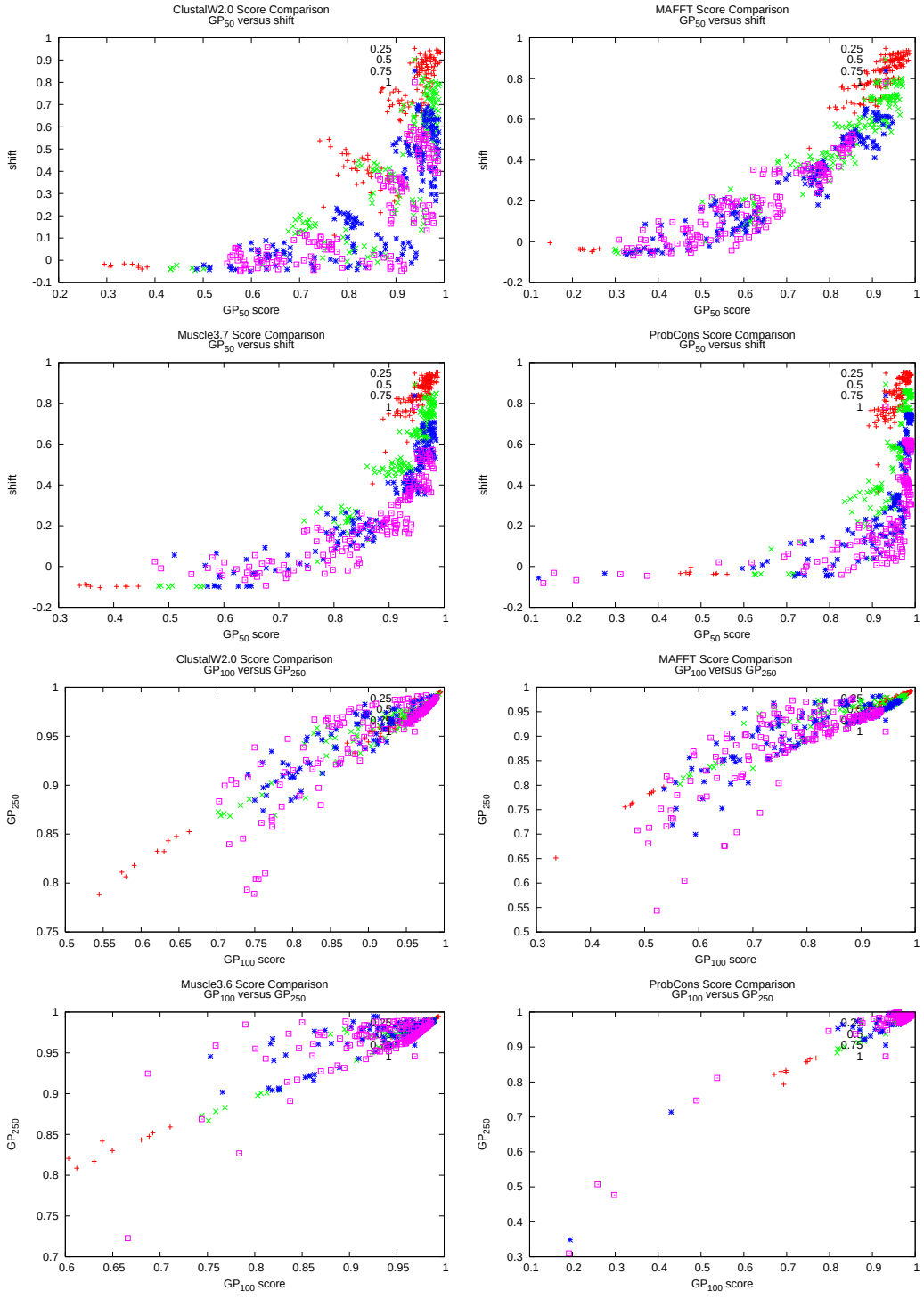

ClustalW2.0 Score Comparison
GP50 versus shift
MAFFT Score Comparison
GP50 versus shift
Muscle3.7 Score Comparison
GP50 versus shift
ProbCons Score Comparison
GP50 versus shift
ClustalW2.0 Score Comparison
GP100 versus GP250
MAFFT Score Comparison
GP100 versus GP250
Muscle3.6 Score Comparison
GP100 versus GP250
ProbCons Score Comparison
GP100 versus GP250
shift
GP50 score
GP250
GP100 score
0.25
0.5
0.75
1

Figure B.2

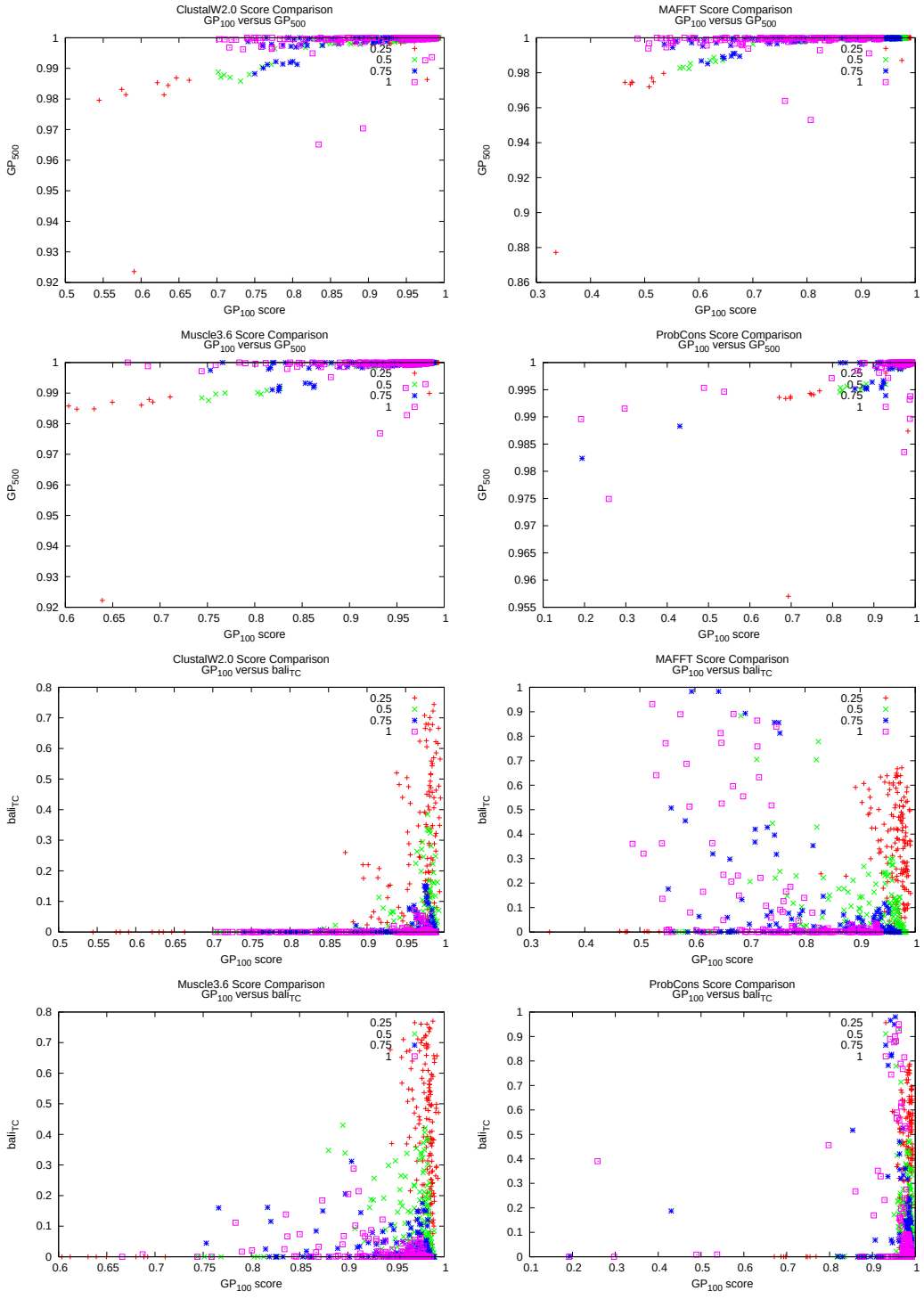

ClustalW2.0 Score Comparison
GP100 versus GP500
MAFFT Score Comparison
GP100 versus GP500
Muscle3.6 Score Comparison
GP100 versus GP500
ProbCons Score Comparison
GP100 versus GP500
ClustalW2.0 Score Comparison
GP100 versus balitc
MAFFT Score Comparison
GP100 versus balitc
Muscle3.6 Score Comparison
GP100 versus balitc
ProbCons Score Comparison
GP100 versus balitc
GP500
balitc
GP100 score
0.25
0.5
0.75
1

Figure B.2

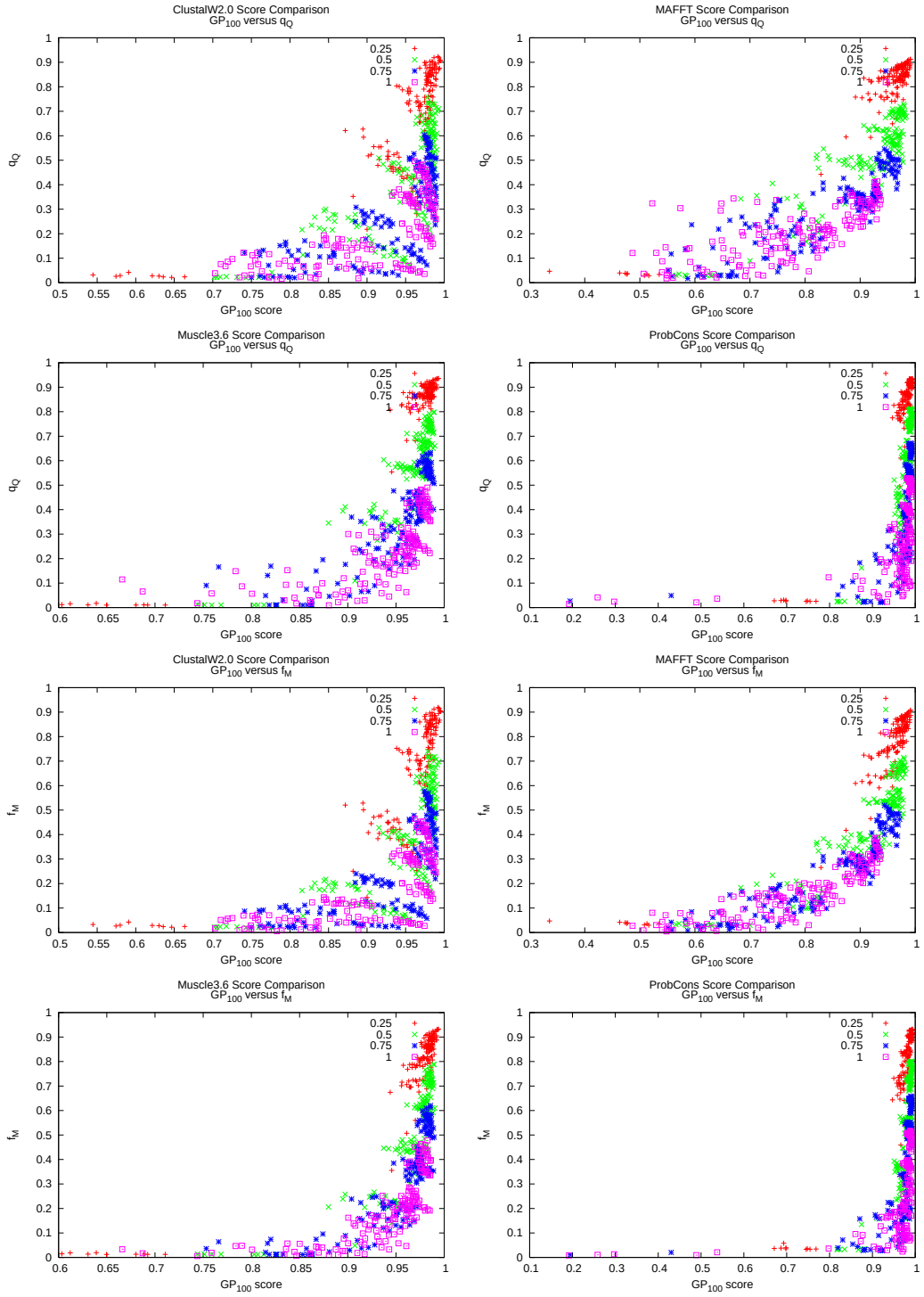

ClustalW2.0 Score Comparison
GP100 versus qQ
MAFFT Score Comparison
GP100 versus qQ
Muscle3.6 Score Comparison
GP100 versus qQ
ProbCons Score Comparison
GP100 versus qQ
ClustalW2.0 Score Comparison
GP100 versus fM
MAFFT Score Comparison
GP100 versus fM
Muscle3.6 Score Comparison
GP100 versus fM
ProbCons Score Comparison
GP100 versus fM
0.25
0.5
0.75
1
qQ
fM
GP100 score

Figure B.2

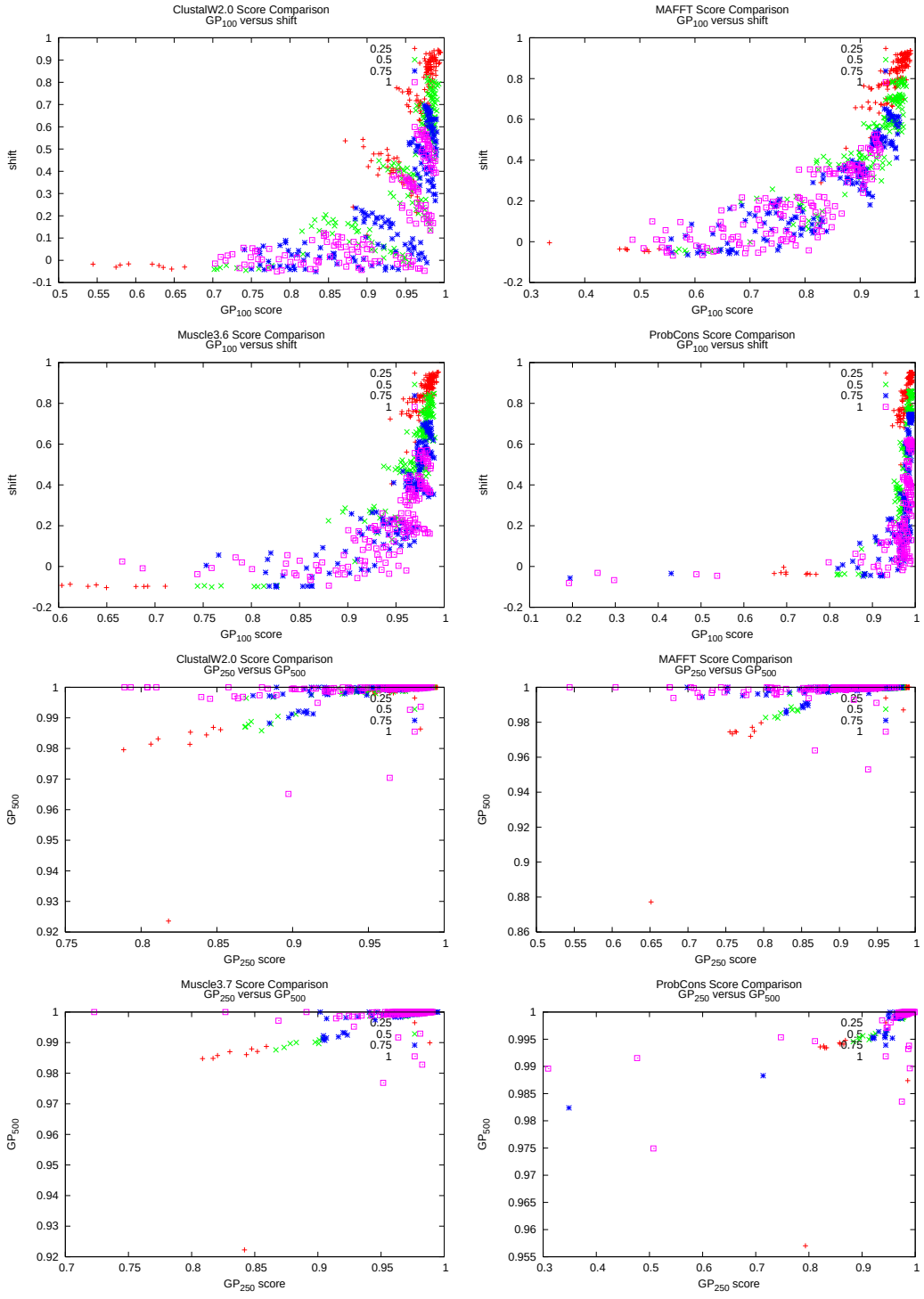

ClustalW2.0 Score Comparison
GP100 versus shift
MAFFT Score Comparison
GP100 versus shift
Muscle3.6 Score Comparison
GP100 versus shift
ProbCons Score Comparison
GP100 versus shift
ClustalW2.0 Score Comparison
GP250 versus GP500
MAFFT Score Comparison
GP250 versus GP500
Muscle3.7 Score Comparison
GP250 versus GP500
ProbCons Score Comparison
GP250 versus GP500
shift
GP100 score
GP500
GP250 score
0.25
0.5
0.75
1

Figure B.2

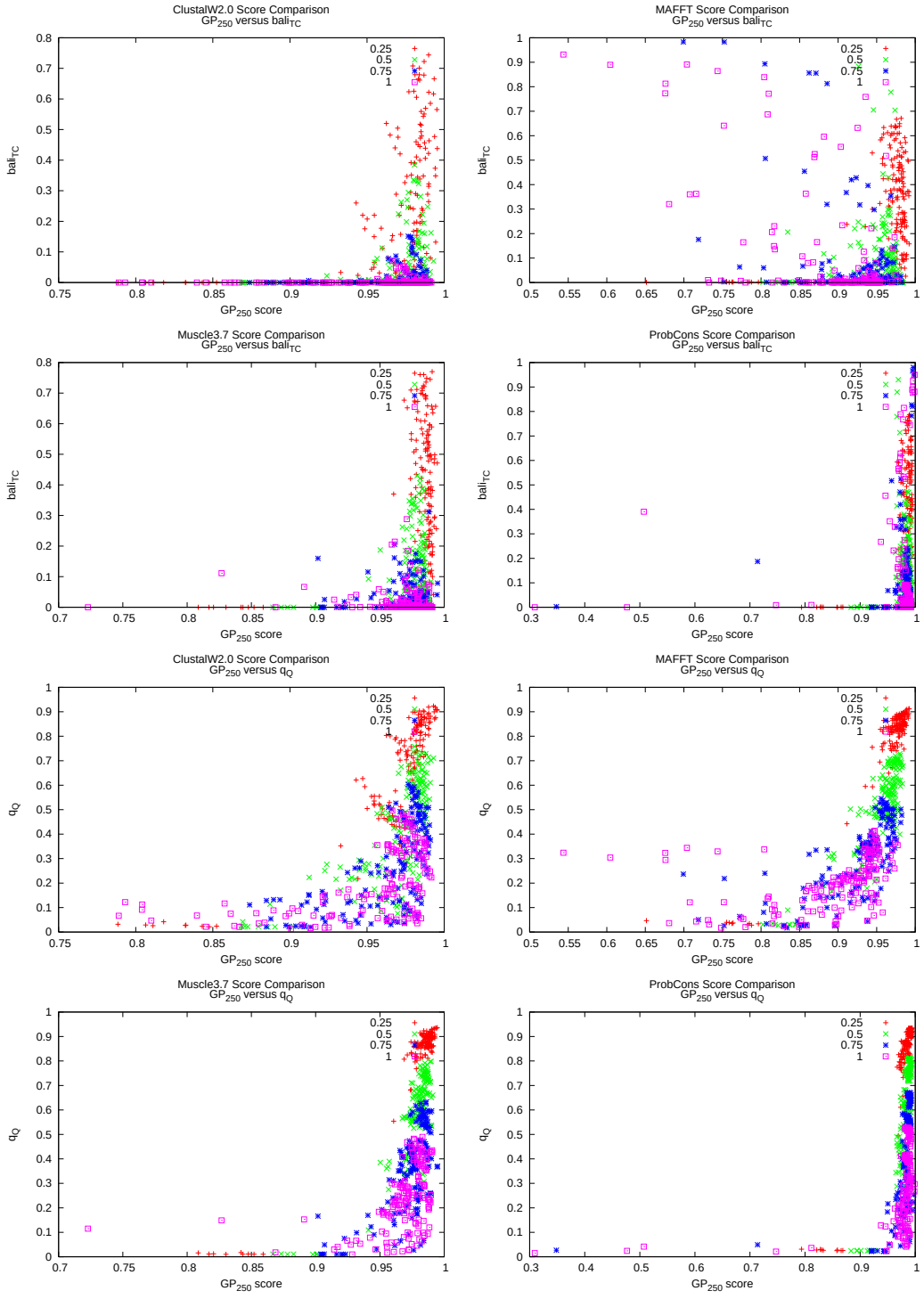

ClustalW2.0 Score Comparison
GP250 versus baliTC
MAFFT Score Comparison
GP250 versus baliTC
Muscle3.7 Score Comparison
GP250 versus baliTC
ProbCons Score Comparison
GP250 versus baliTC
ClustalW2.0 Score Comparison
GP250 versus qQ
MAFFT Score Comparison
GP250 versus qQ
Muscle3.7 Score Comparison
GP250 versus qQ
ProbCons Score Comparison
GP250 versus qQ
baliTC
qQ
GP250 score
0.25
0.5
0.75
1

Figure B.2

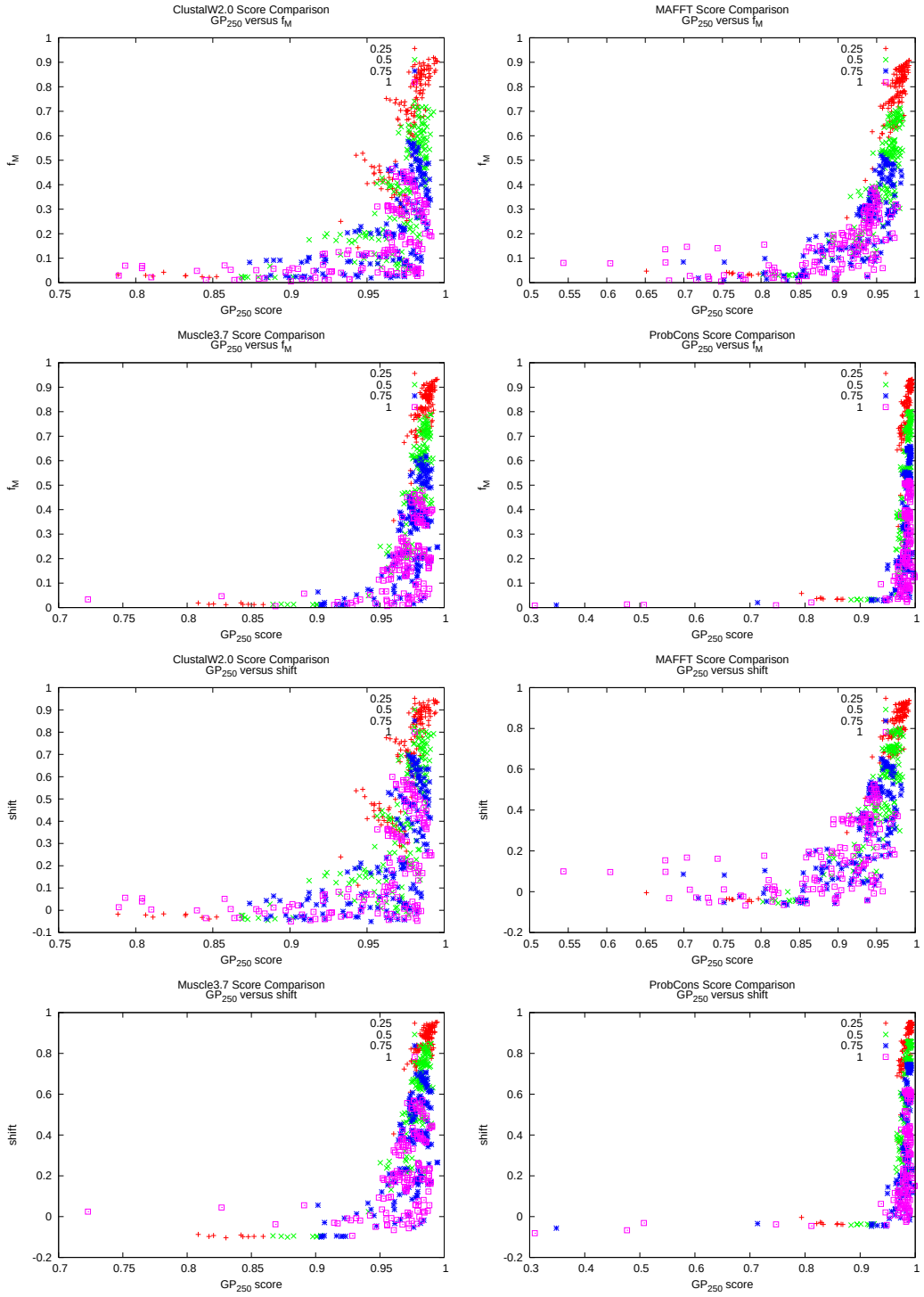

ClustalW2.0 Score Comparison
GP250 versus fM
MAFFT Score Comparison
GP250 versus fM
Muscle3.7 Score Comparison
GP250 versus fM
ProbCons Score Comparison
GP250 versus fM
ClustalW2.0 Score Comparison
GP250 versus shift
MAFFT Score Comparison
GP250 versus shift
Muscle3.7 Score Comparison
GP250 versus shift
ProbCons Score Comparison
GP250 versus shift
0.25
0.5
0.75
1
fM
shift
GP250 score

Figure B.2

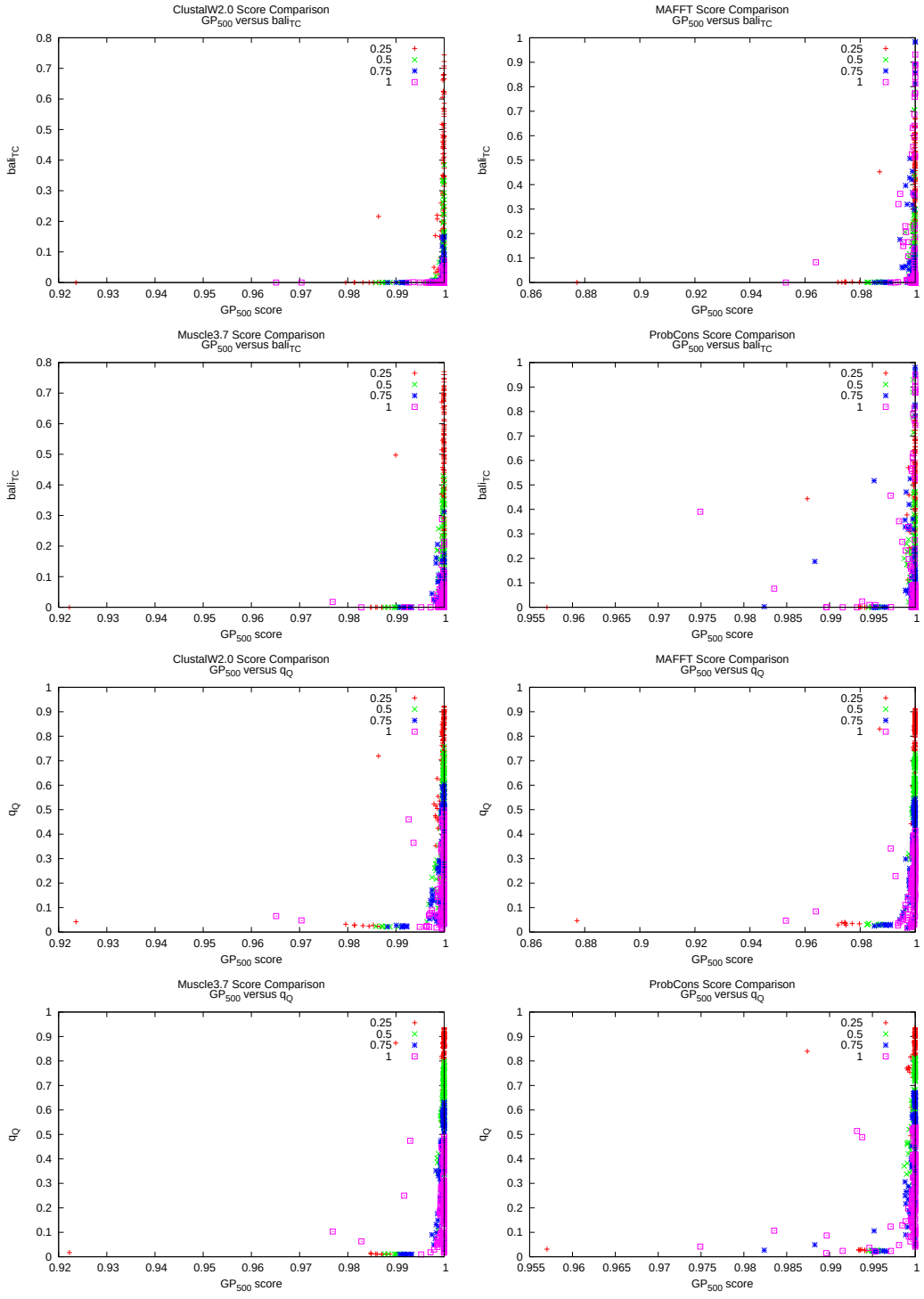

ClustalW2.0 Score Comparison
GP500 versus baliTC
MAFFT Score Comparison
GP500 versus baliTC
Muscle3.7 Score Comparison
GP500 versus baliTC
ProbCons Score Comparison
GP500 versus baliTC
ClustalW2.0 Score Comparison
GP500 versus qQ
MAFFT Score Comparison
GP500 versus qQ
Muscle3.7 Score Comparison
GP500 versus qQ
ProbCons Score Comparison
GP500 versus qQ
0.25
0.5
0.75
1
baliTC
qQ
GP500 score

Figure B.2

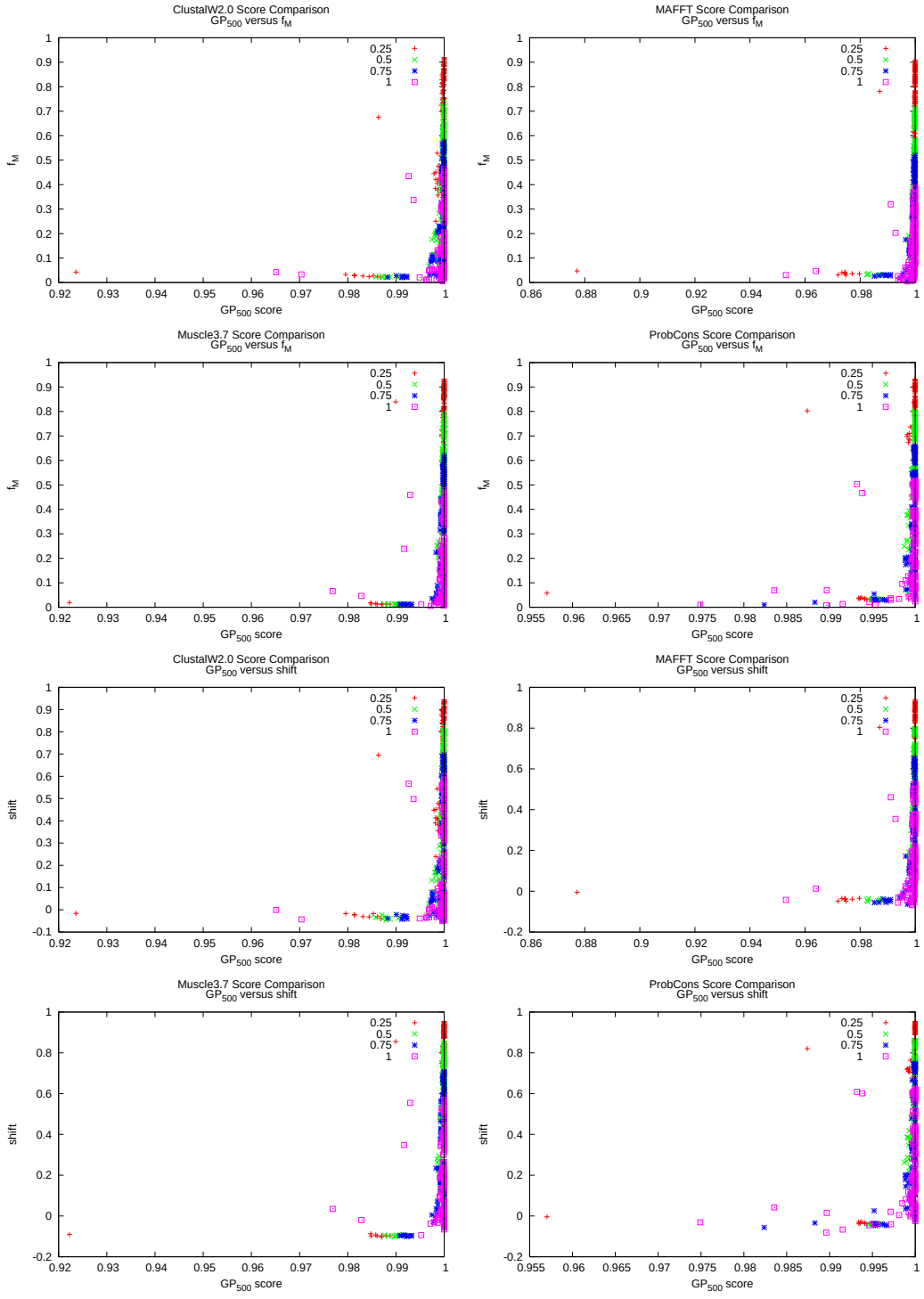

ClustalW2.0 Score Comparison
GP500 versus fM
MAFFT Score Comparison
GP500 versus fM
Muscle3.7 Score Comparison
GP500 versus fM
ProbCons Score Comparison
GP500 versus fM
ClustalW2.0 Score Comparison
GP500 versus shift
MAFFT Score Comparison
GP500 versus shift
Muscle3.7 Score Comparison
GP500 versus shift
ProbCons Score Comparison
GP500 versus shift
0.25
0.5
0.75
1
fM
shift
GP500 score

Figure B.2

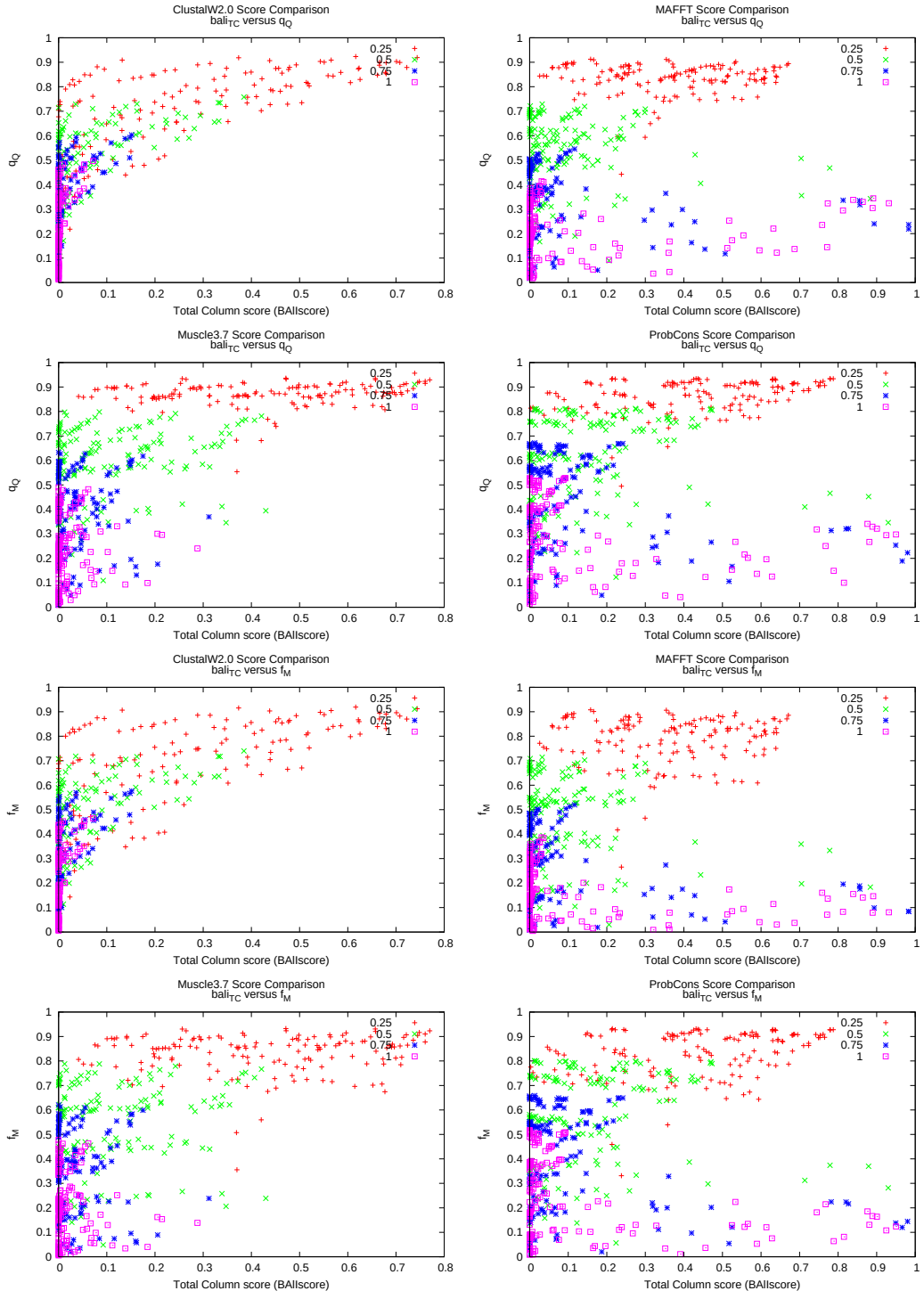

ClustalW2.0 Score Comparison
bali_TC versus q_Q
MAFFT Score Comparison
bali_TC versus q_Q
Muscle3.7 Score Comparison
bali_TC versus q_Q
ProbCons Score Comparison
bali_TC versus q_Q
ClustalW2.0 Score Comparison
bali_TC versus f_M
MAFFT Score Comparison
bali_TC versus f_M
Muscle3.7 Score Comparison
bali_TC versus f_M
ProbCons Score Comparison
bali_TC versus f_M
q_Q
f_M
Total Column score (BAIIscore)
0.25
0.5
0.75
1

Figure B.2

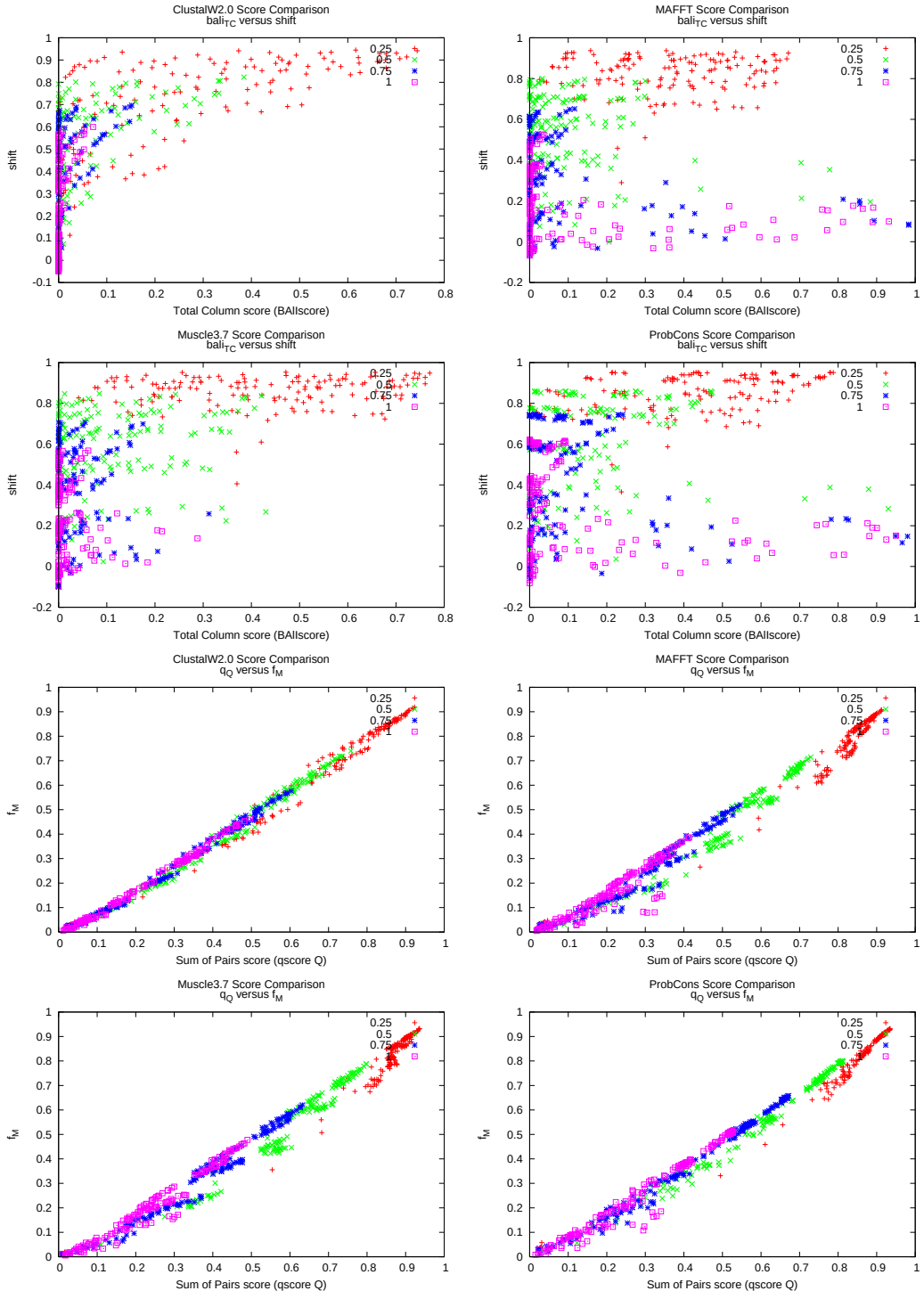

ClustalW2.0 Score Comparison
bali_TC versus shift
MAFFT Score Comparison
bali_TC versus shift
Muscle3.7 Score Comparison
bali_TC versus shift
ProbCons Score Comparison
bali_TC versus shift
ClustalW2.0 Score Comparison
q_Q versus f_M
MAFFT Score Comparison
q_Q versus f_M
Muscle3.7 Score Comparison
q_Q versus f_M
ProbCons Score Comparison
q_Q versus f_M
shift
f_M
Total Column score (BAIlscore)
Sum of Pairs score (qscore Q)
0.25
0.5
0.75
1

Figure B.2

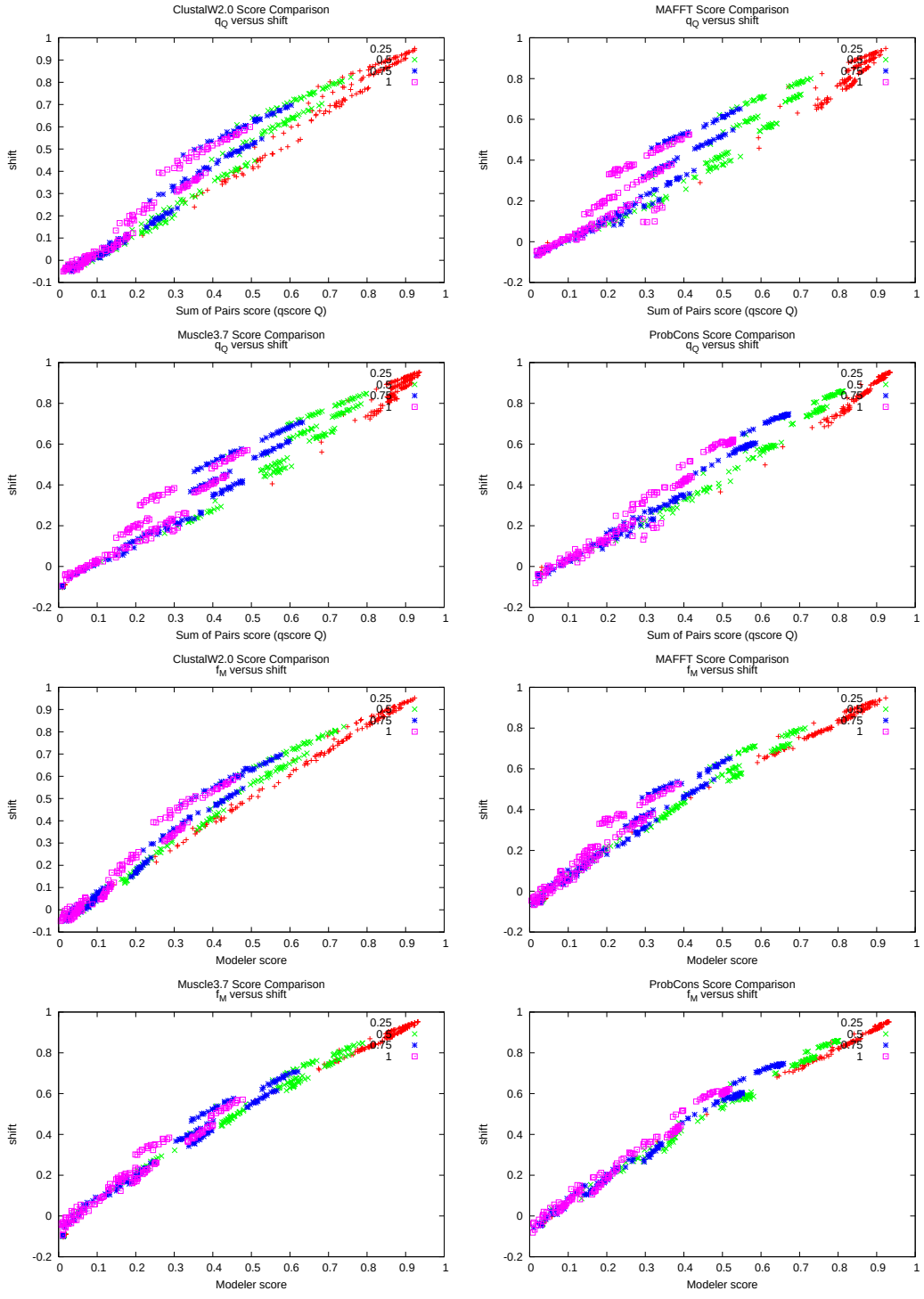

ClustalW2.0 Score Comparison
$q_Q$ versus shift
MAFFT Score Comparison
$q_Q$ versus shift
Muscle3.7 Score Comparison
$q_Q$ versus shift
ProbCons Score Comparison
$q_Q$ versus shift
ClustalW2.0 Score Comparison
$f_M$ versus shift
MAFFT Score Comparison
$f_M$ versus shift
Muscle3.7 Score Comparison
$f_M$ versus shift
ProbCons Score Comparison
$f_M$ versus shift
shift
Sum of Pairs score (qscore Q)
Modeler score
0.25
0.5
0.75
1

Figure B.2

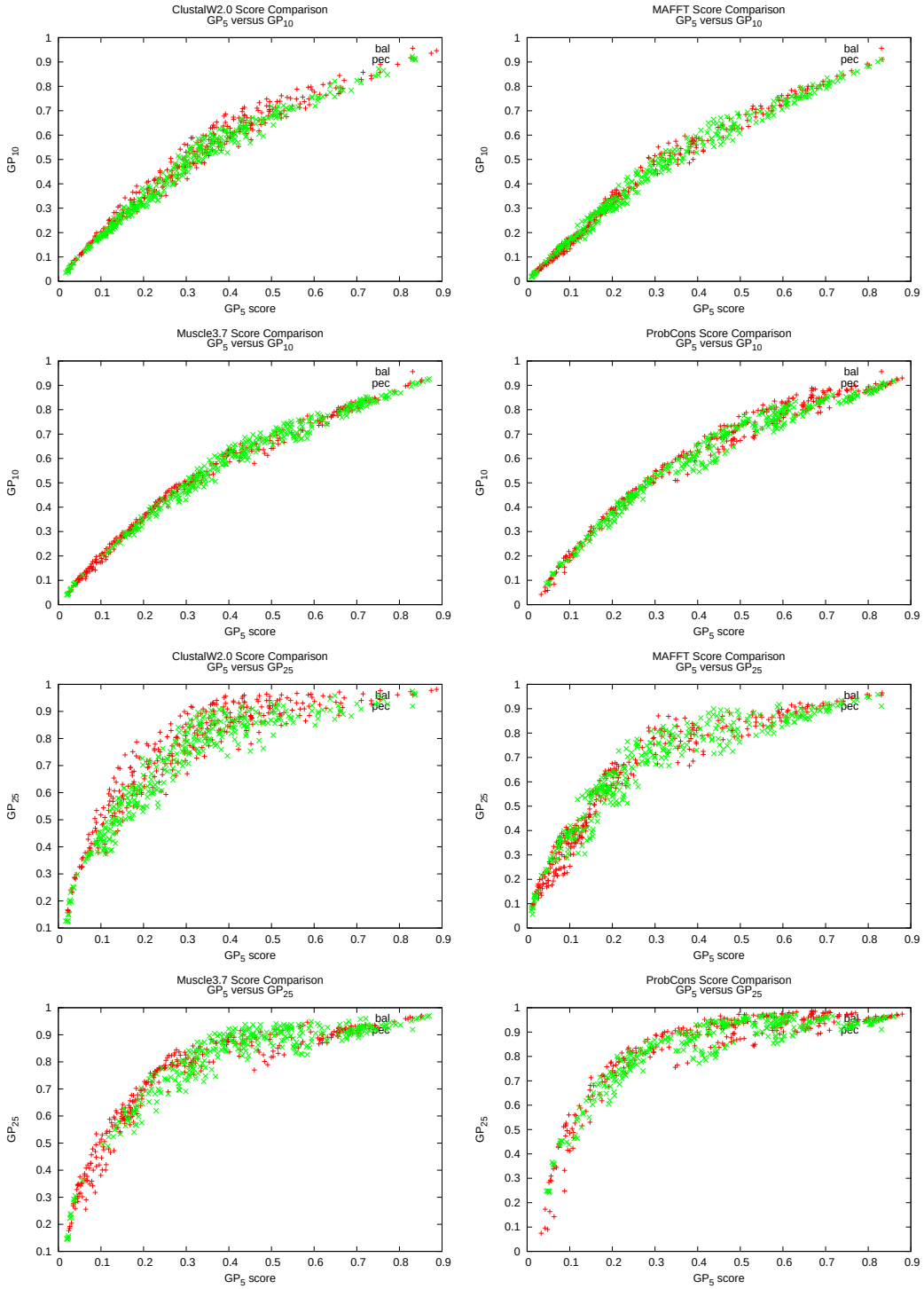

ClustalW2.0 Score Comparison
GP5 versus GP10
MAFFT Score Comparison
GP5 versus GP10
Muscle3.7 Score Comparison
GP5 versus GP10
ProbCons Score Comparison
GP5 versus GP10
ClustalW2.0 Score Comparison
GP5 versus GP25
MAFFT Score Comparison
GP5 versus GP25
Muscle3.7 Score Comparison
GP5 versus GP25
ProbCons Score Comparison
GP5 versus GP25
bal
pec
GP10
GP25
GP5 score

Figure B.2

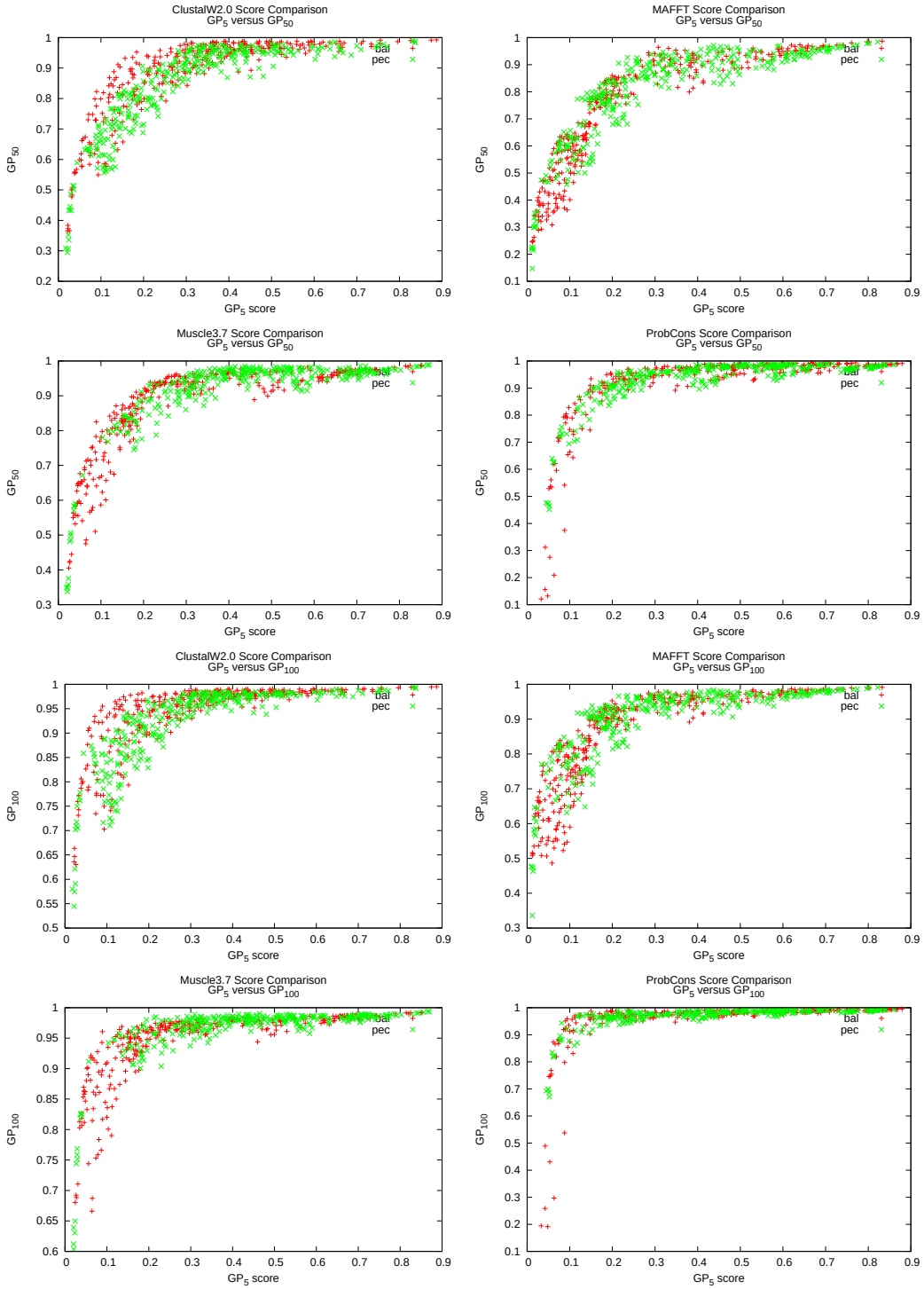

ClustalW2.0 Score Comparison
GP5 versus GP50
MAFFT Score Comparison
GP5 versus GP50
Muscle3.7 Score Comparison
GP5 versus GP50
ProbCons Score Comparison
GP5 versus GP50
ClustalW2.0 Score Comparison
GP5 versus GP100
MAFFT Score Comparison
GP5 versus GP100
Muscle3.7 Score Comparison
GP5 versus GP100
ProbCons Score Comparison
GP5 versus GP100
bal
pec
GP50
GP100
GP5 score

Figure B.2

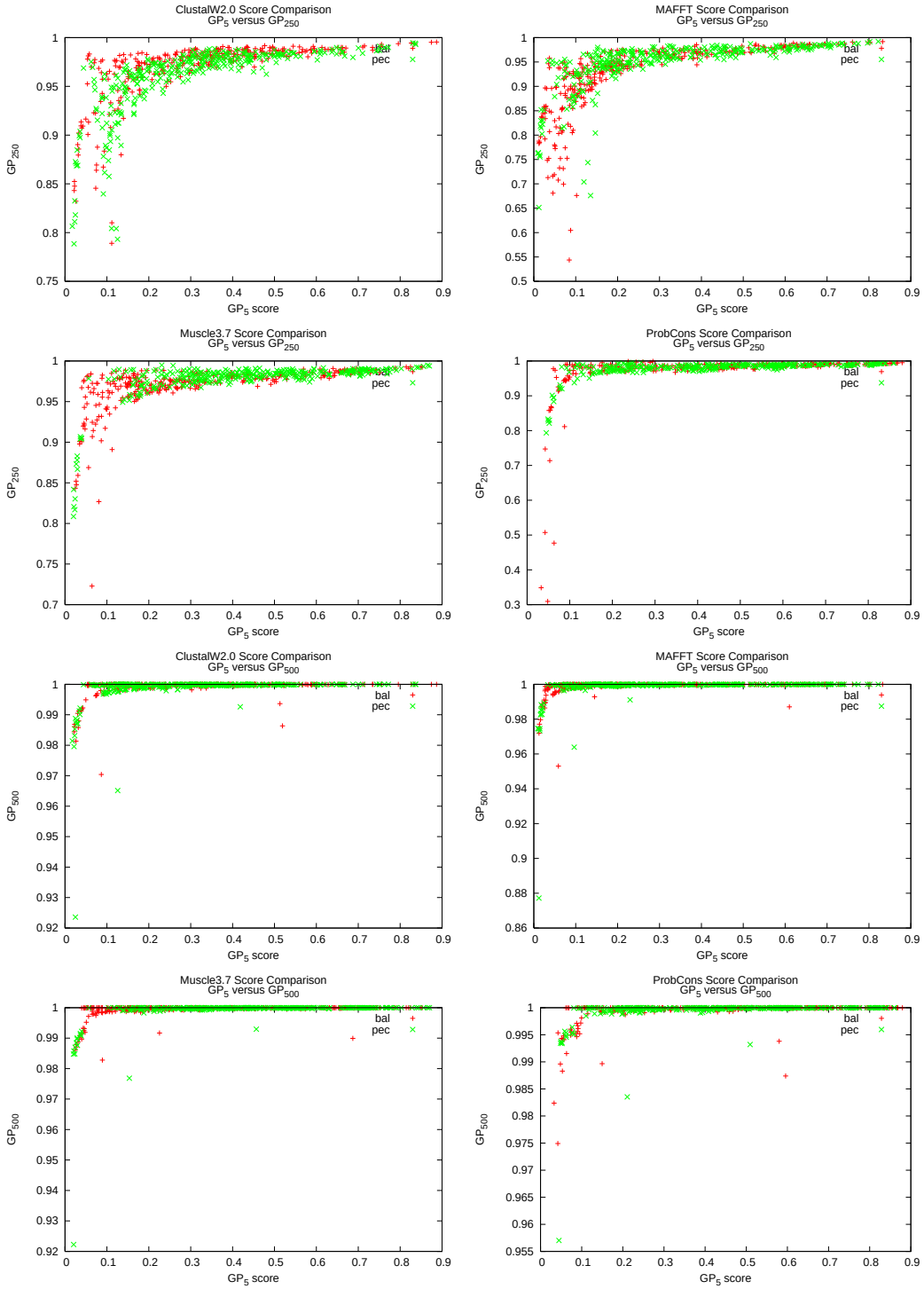

ClustalW2.0 Score Comparison
GP5 versus GP250
MAFFT Score Comparison
GP5 versus GP250
Muscle3.7 Score Comparison
GP5 versus GP250
ProbCons Score Comparison
GP5 versus GP250
ClustalW2.0 Score Comparison
GP5 versus GP500
MAFFT Score Comparison
GP5 versus GP500
Muscle3.7 Score Comparison
GP5 versus GP500
ProbCons Score Comparison
GP5 versus GP500
bal
pec
GP250
GP500
GP5 score

Figure B.2

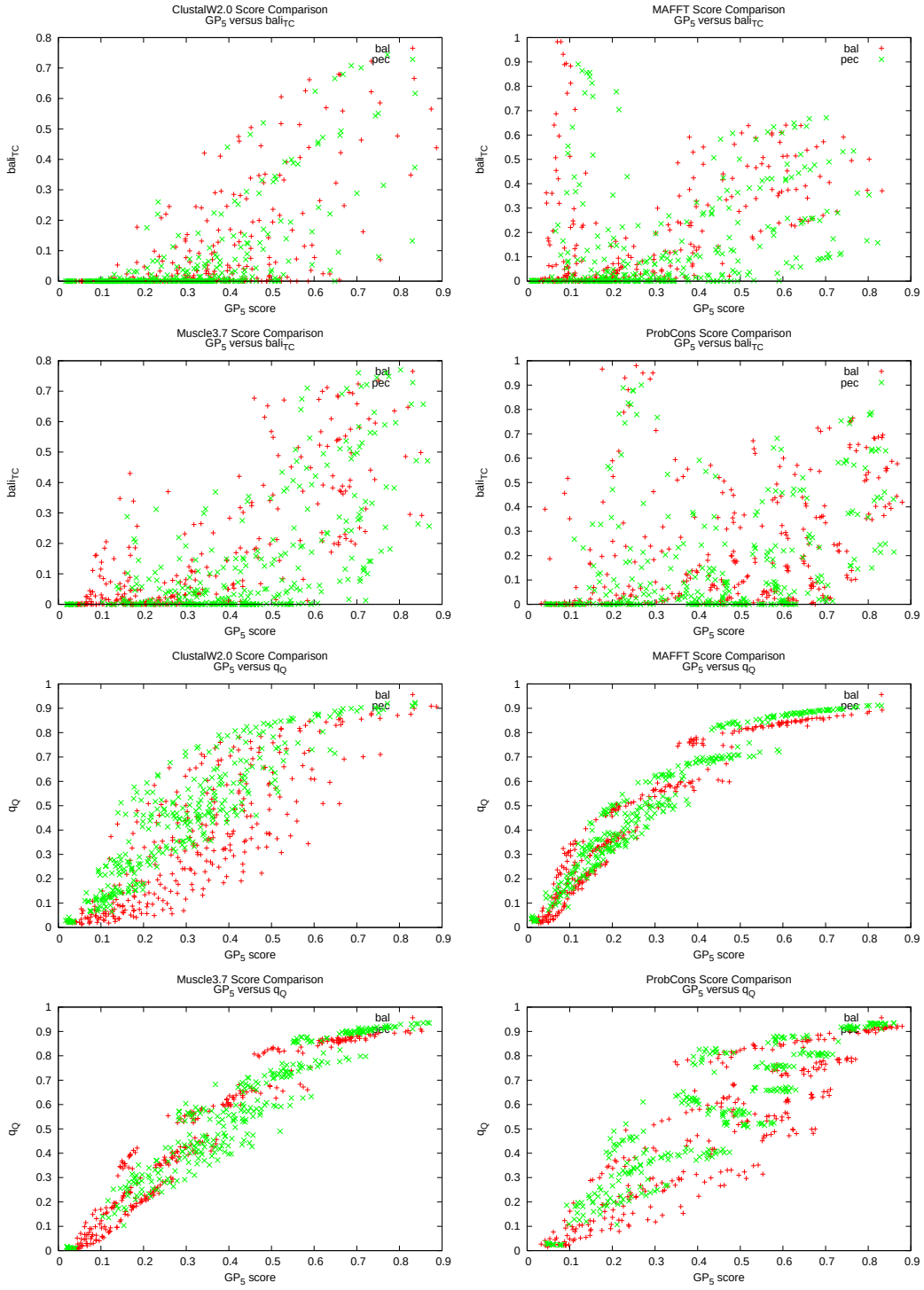

ClustalW2.0 Score Comparison
GP5 versus balTC
MAFFT Score Comparison
GP5 versus balTC
Muscle3.7 Score Comparison
GP5 versus balTC
ProbCons Score Comparison
GP5 versus balTC
ClustalW2.0 Score Comparison
GP5 versus qQ
MAFFT Score Comparison
GP5 versus qQ
Muscle3.7 Score Comparison
GP5 versus qQ
ProbCons Score Comparison
GP5 versus qQ
bal
pec
balTC
qQ
GP5 score

Figure B.2

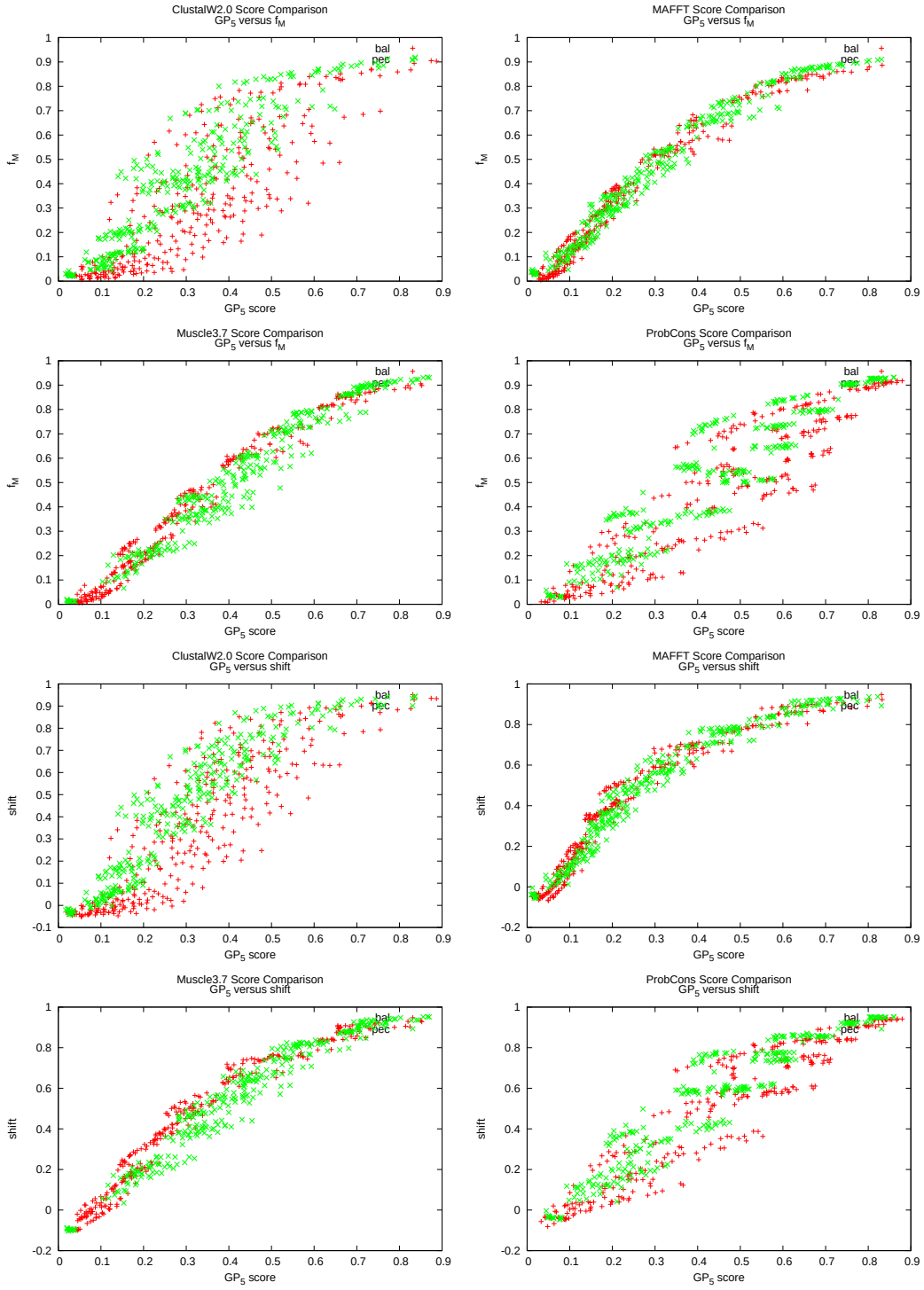

ClustalW2.0 Score Comparison
GP5 versus fM
MAFFT Score Comparison
GP5 versus fM
Muscle3.7 Score Comparison
GP5 versus fM
ProbCons Score Comparison
GP5 versus fM
ClustalW2.0 Score Comparison
GP5 versus shift
MAFFT Score Comparison
GP5 versus shift
Muscle3.7 Score Comparison
GP5 versus shift
ProbCons Score Comparison
GP5 versus shift
bal
pec
fM
shift
GP5 score

Figure B.2

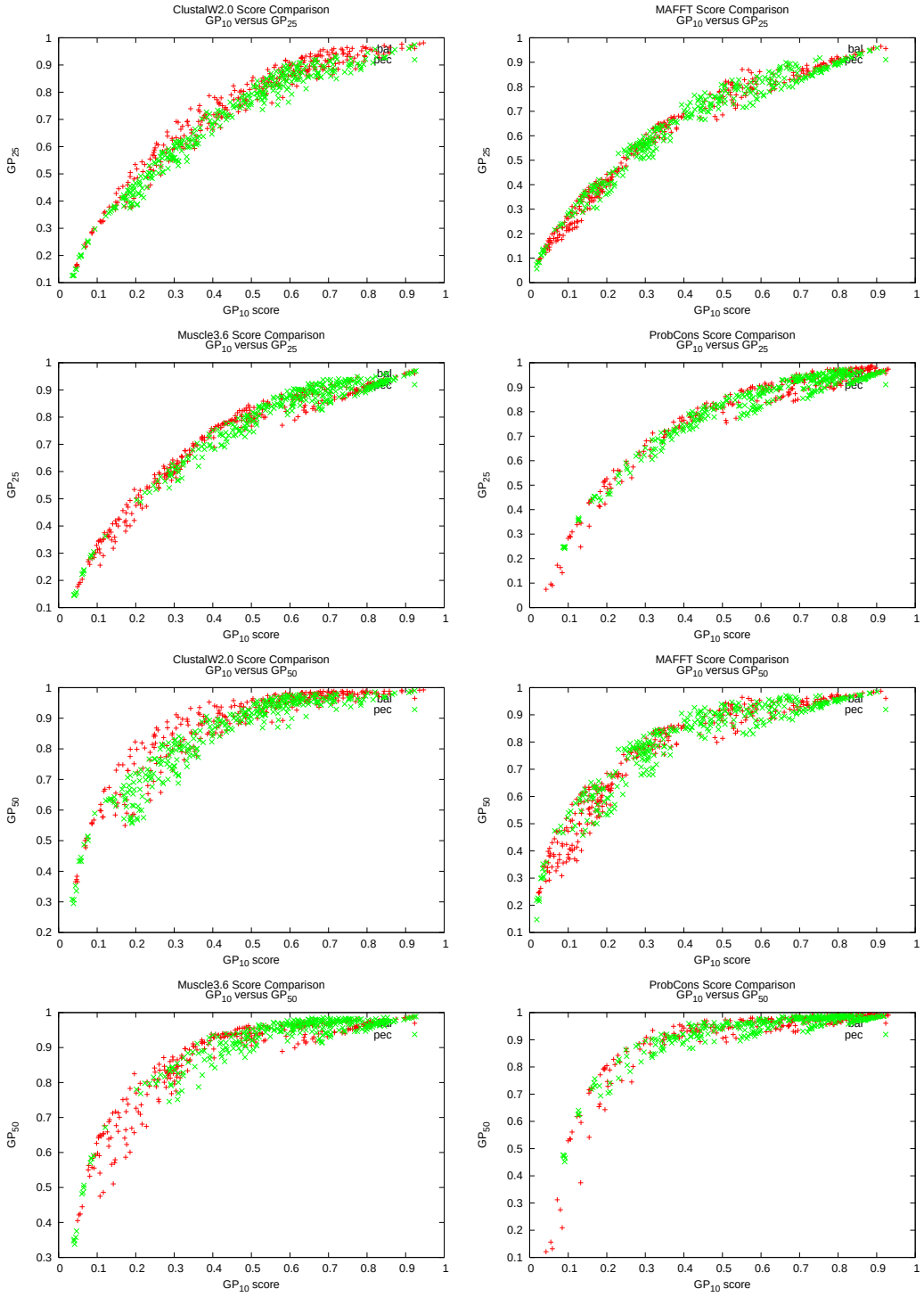

ClustalW2.0 Score Comparison
GP10 versus GP25
MAFFT Score Comparison
GP10 versus GP25
Muscle3.6 Score Comparison
GP10 versus GP25
ProbCons Score Comparison
GP10 versus GP25
ClustalW2.0 Score Comparison
GP10 versus GP50
MAFFT Score Comparison
GP10 versus GP50
Muscle3.6 Score Comparison
GP10 versus GP50
ProbCons Score Comparison
GP10 versus GP50
bal
pec
GP25
GP50
GP10 score

Figure B.2

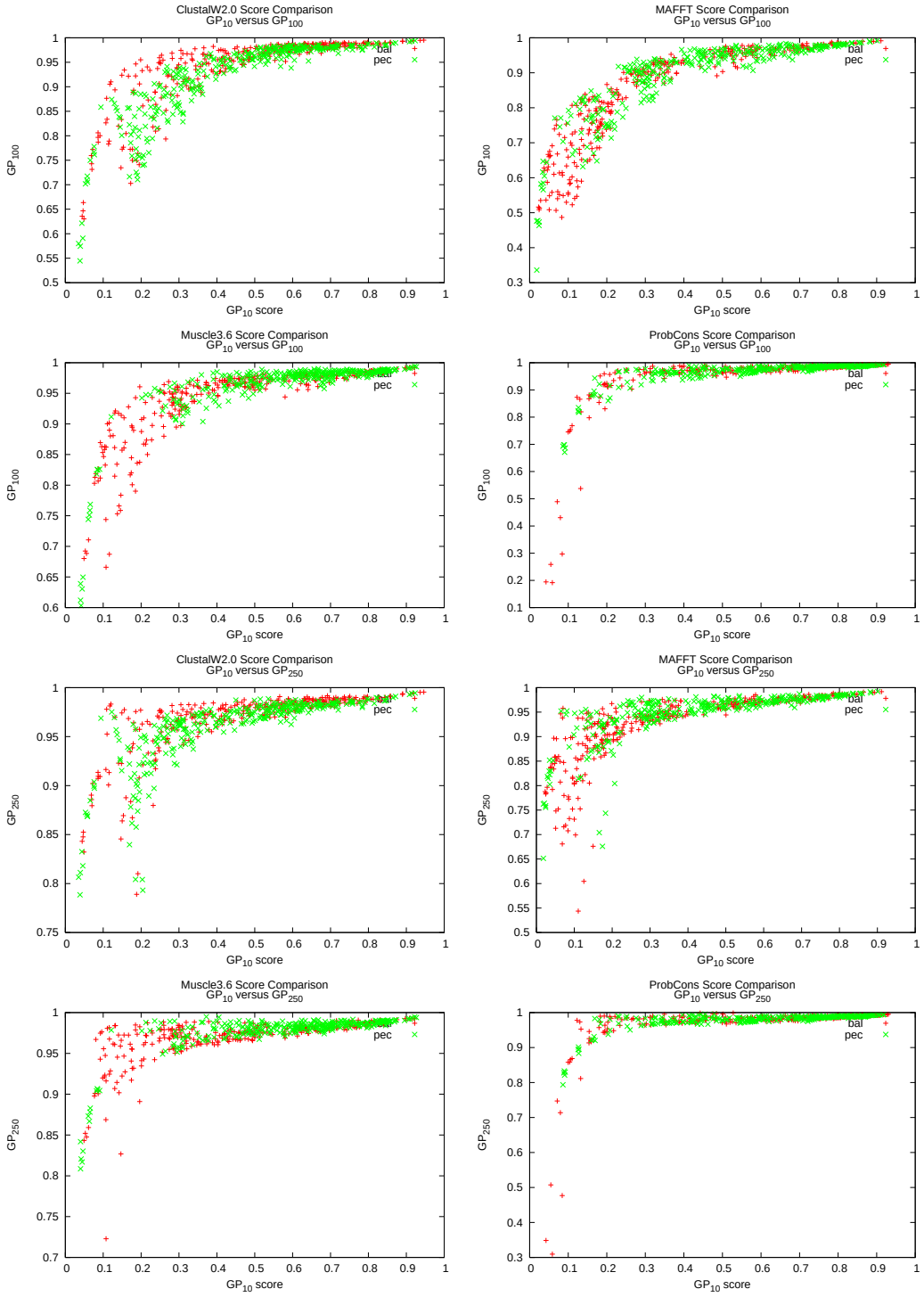

ClustalW2.0 Score Comparison
GP10 versus GP100
MAFFT Score Comparison
GP10 versus GP100
Muscle3.6 Score Comparison
GP10 versus GP100
ProbCons Score Comparison
GP10 versus GP100
ClustalW2.0 Score Comparison
GP10 versus GP250
MAFFT Score Comparison
GP10 versus GP250
Muscle3.6 Score Comparison
GP10 versus GP250
ProbCons Score Comparison
GP10 versus GP250
bal
pec
GP100
GP250
GP10 score

Figure B.2

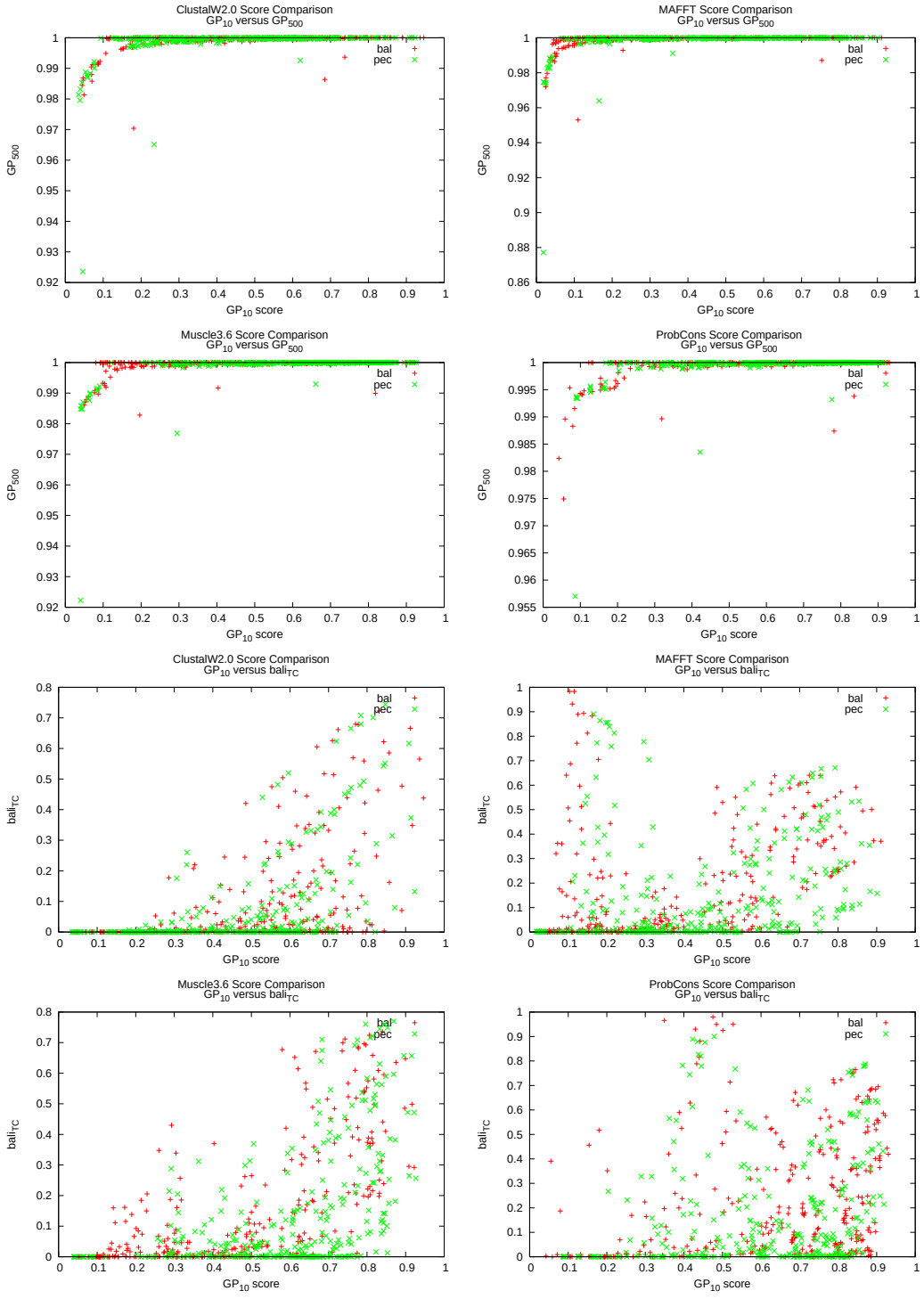

ClustalW2.0 Score Comparison
GP10 versus GP500
MAFFT Score Comparison
GP10 versus GP500
Muscle3.6 Score Comparison
GP10 versus GP500
ProbCons Score Comparison
GP10 versus GP500
ClustalW2.0 Score Comparison
GP10 versus baliTC
MAFFT Score Comparison
GP10 versus baliTC
Muscle3.6 Score Comparison
GP10 versus baliTC
ProbCons Score Comparison
GP10 versus baliTC
bal
pec
GP500
baliTC
GP10 score

Figure B.2

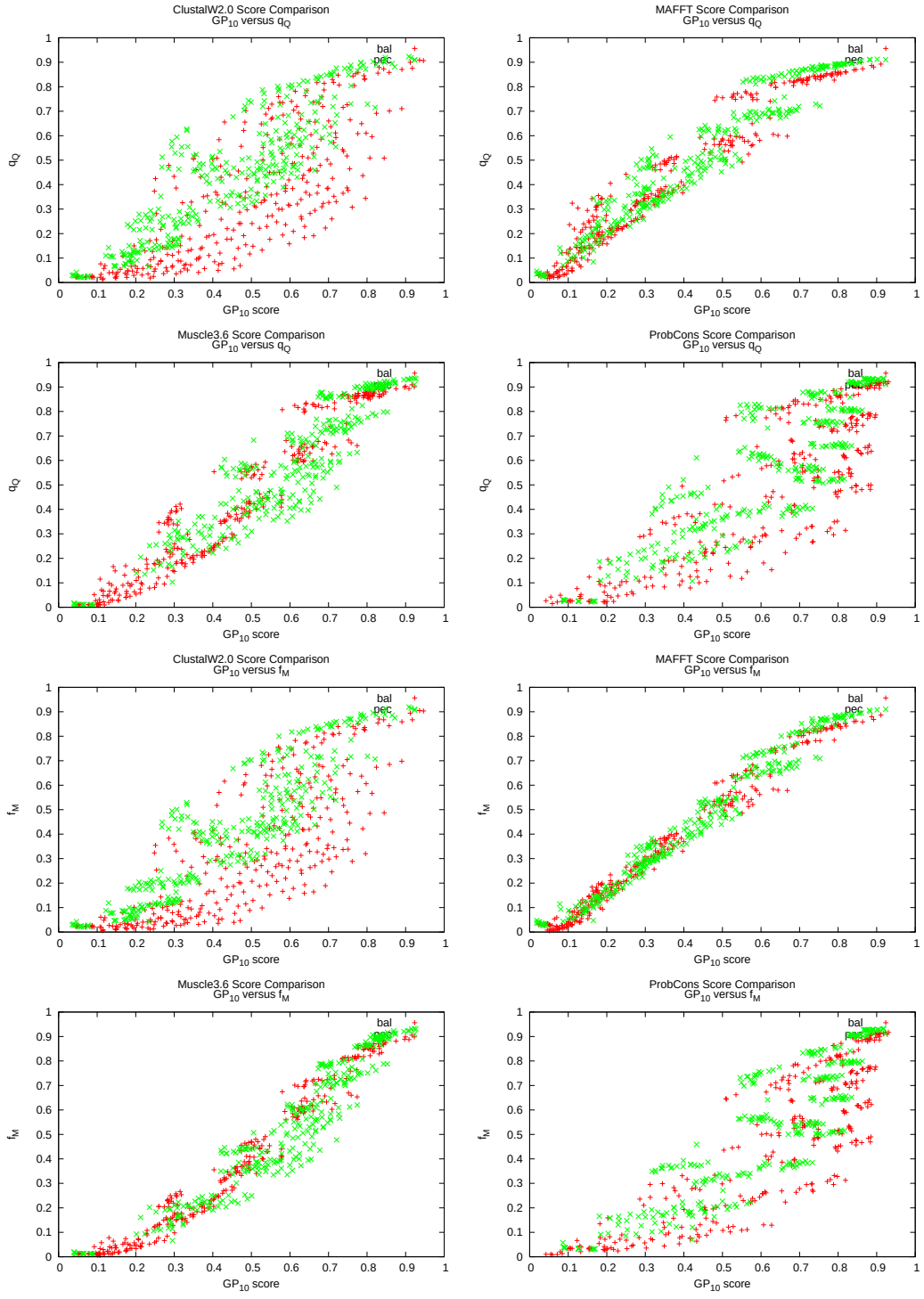

ClustalW2.0 Score Comparison
GP10 versus qQ
MAFFT Score Comparison
GP10 versus qQ
Muscle3.6 Score Comparison
GP10 versus qQ
ProbCons Score Comparison
GP10 versus qQ
ClustalW2.0 Score Comparison
GP10 versus fM
MAFFT Score Comparison
GP10 versus fM
Muscle3.6 Score Comparison
GP10 versus fM
ProbCons Score Comparison
GP10 versus fM
bal
pec
qQ
fM
GP10 score

Figure B.2

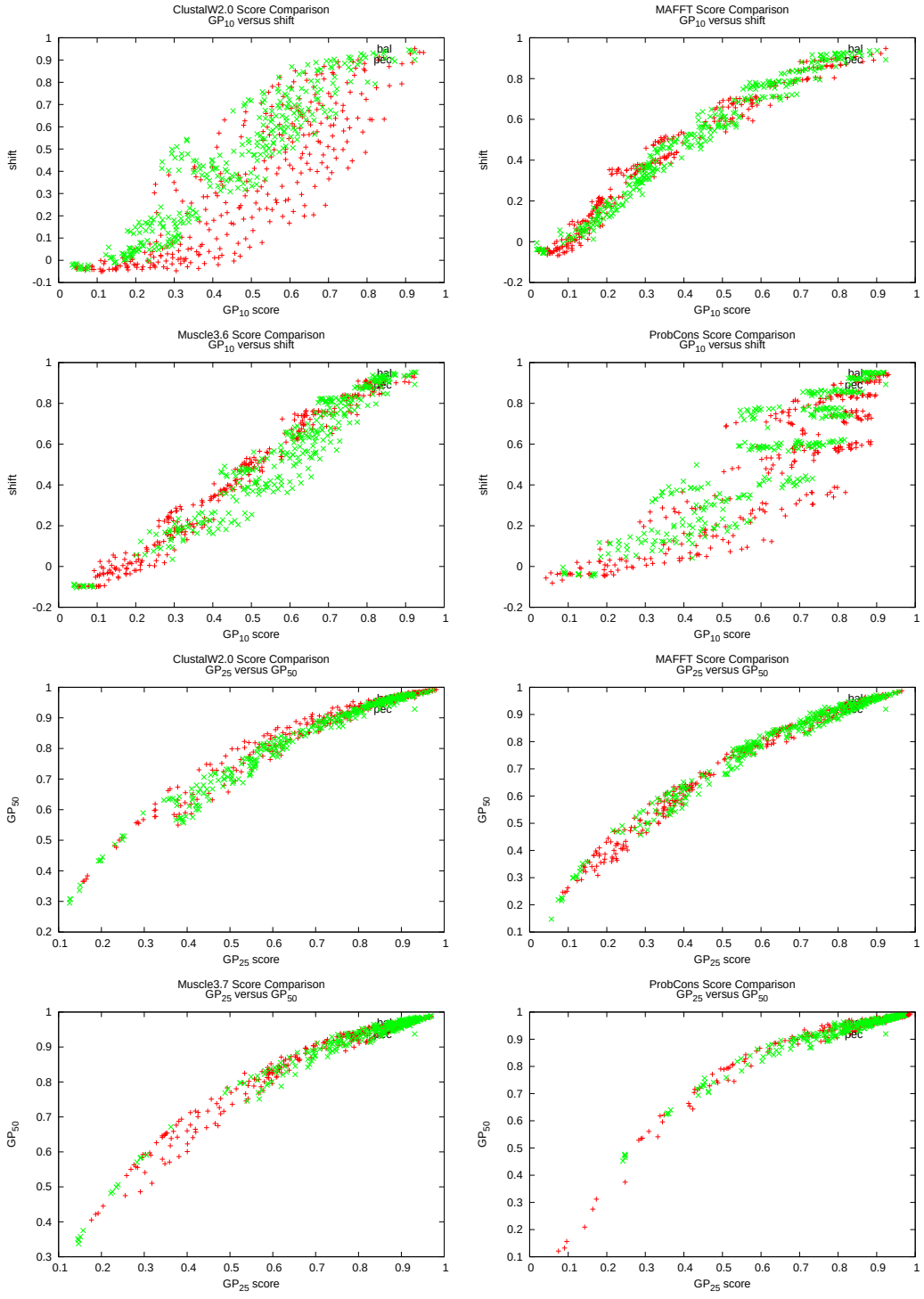

ClustalW2.0 Score Comparison
GP10 versus shift
MAFFT Score Comparison
GP10 versus shift
Muscle3.6 Score Comparison
GP10 versus shift
ProbCons Score Comparison
GP10 versus shift
ClustalW2.0 Score Comparison
GP25 versus GP50
MAFFT Score Comparison
GP25 versus GP50
Muscle3.7 Score Comparison
GP25 versus GP50
ProbCons Score Comparison
GP25 versus GP50
shift
GP50
GP10 score
GP25 score
bal
pec

Figure B.2

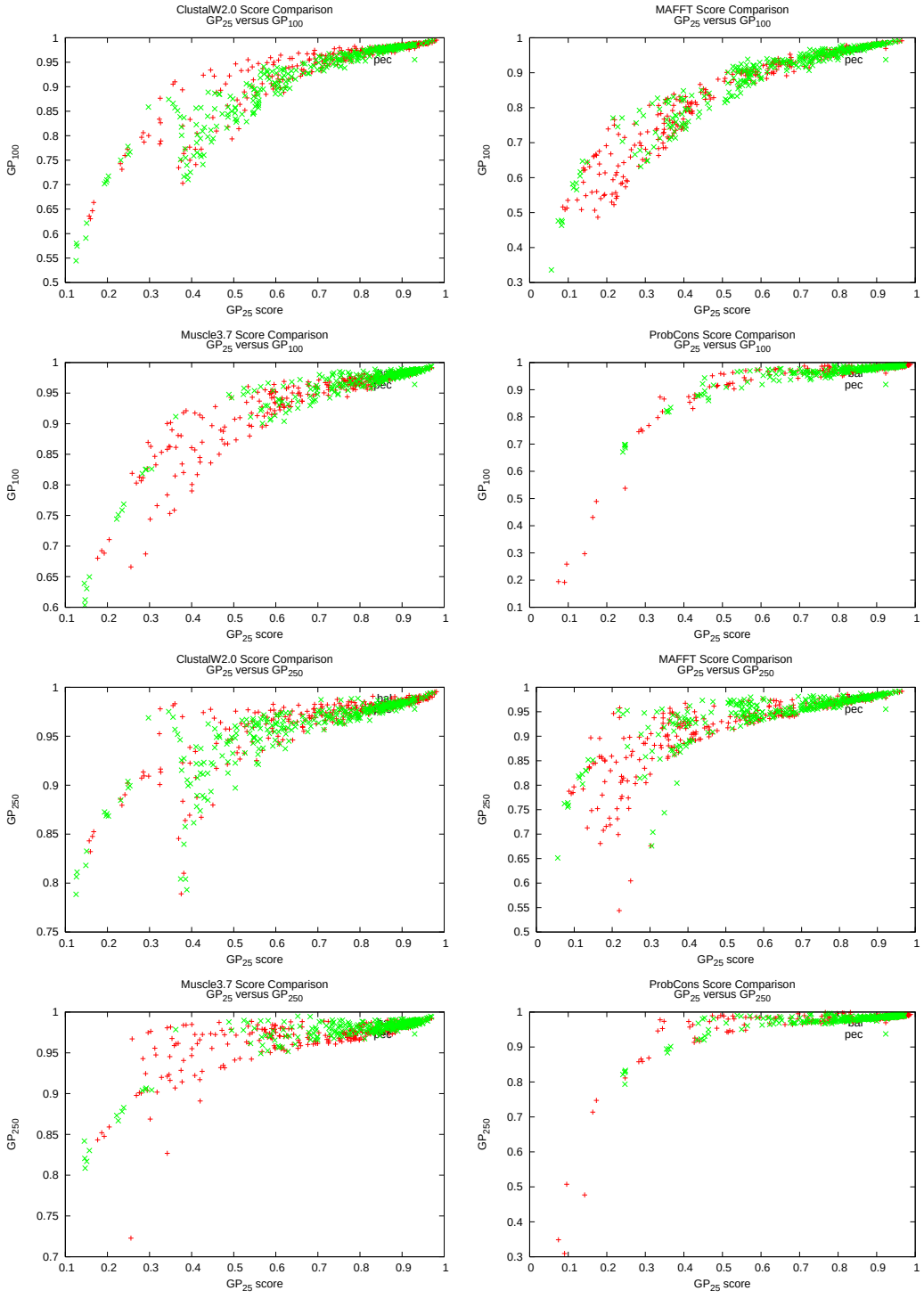

ClustalW2.0 Score Comparison
GP25 versus GP100
MAFFT Score Comparison
GP25 versus GP100
Muscle3.7 Score Comparison
GP25 versus GP100
ProbCons Score Comparison
GP25 versus GP100
ClustalW2.0 Score Comparison
GP25 versus GP250
MAFFT Score Comparison
GP25 versus GP250
Muscle3.7 Score Comparison
GP25 versus GP250
ProbCons Score Comparison
GP25 versus GP250
GP100
GP250
GP25 score
pec
bali

Figure B.2

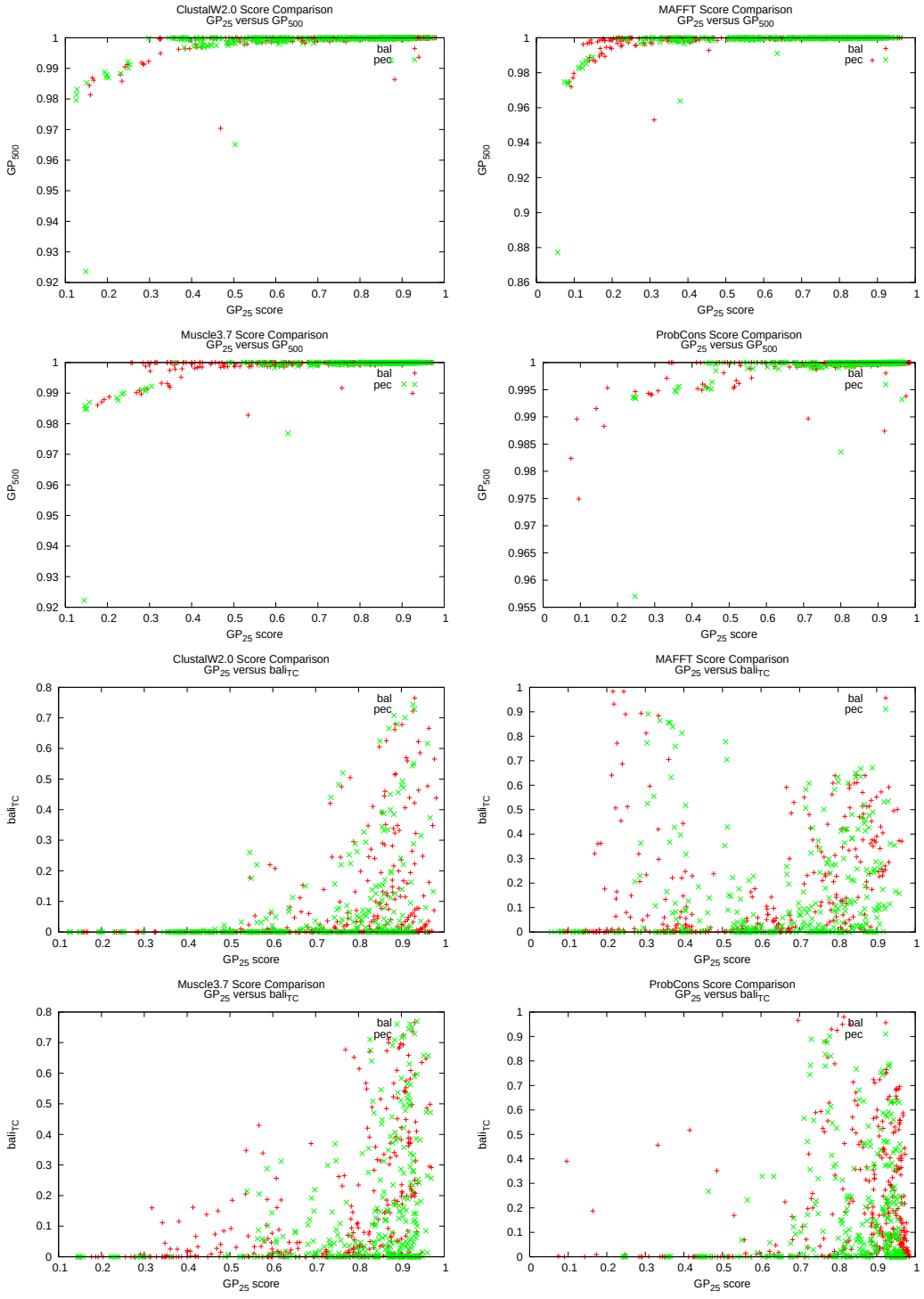

ClustalW2.0 Score Comparison
GP25 versus GP500
MAFFT Score Comparison
GP25 versus GP500
Muscle3.7 Score Comparison
GP25 versus GP500
ProbCons Score Comparison
GP25 versus GP500
ClustalW2.0 Score Comparison
GP25 versus balitc
MAFFT Score Comparison
GP25 versus balitc
Muscle3.7 Score Comparison
GP25 versus balitc
ProbCons Score Comparison
GP25 versus balitc
bal
pec
GP500
balitc
GP25 score

Figure B.2

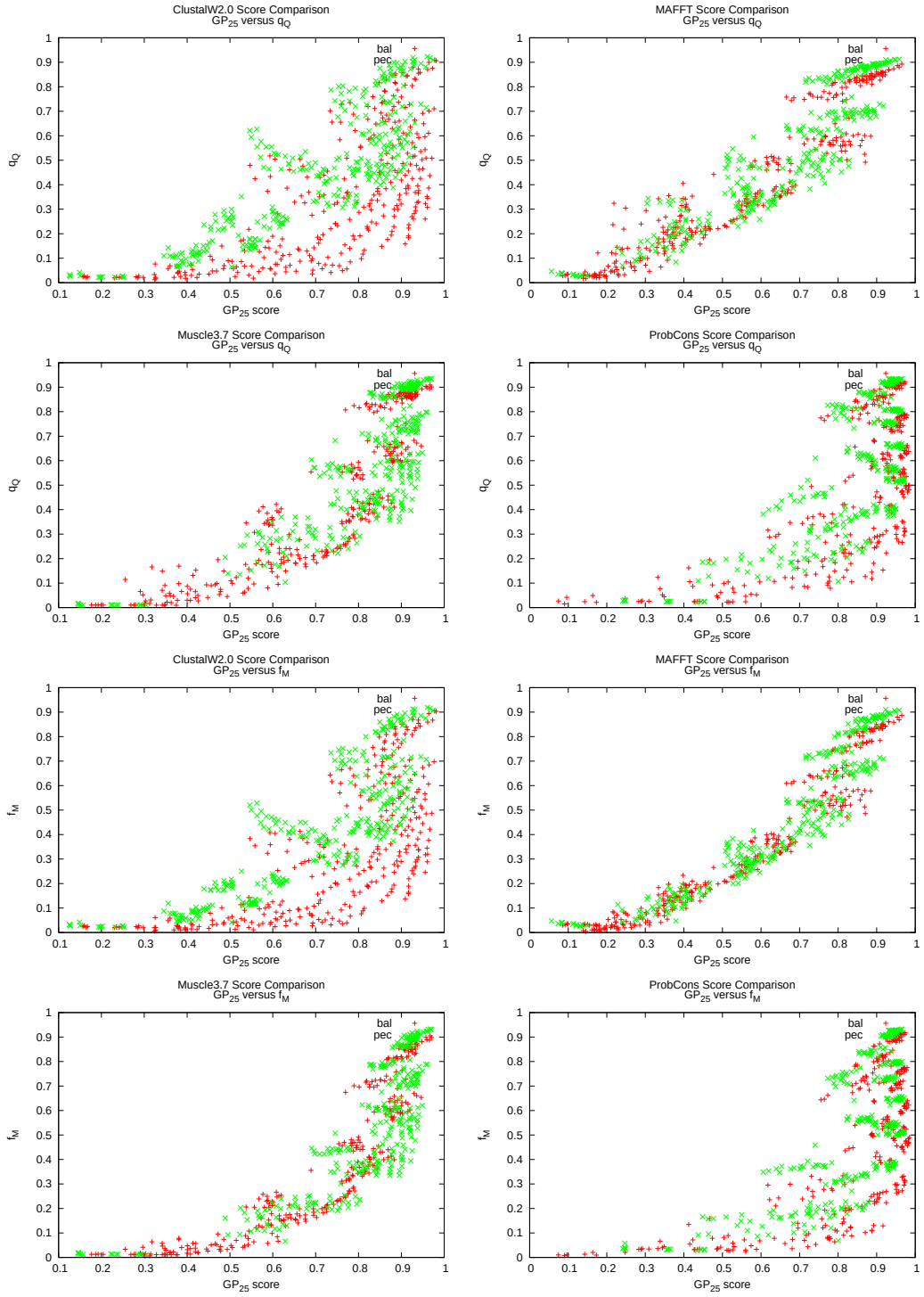

ClustalW2.0 Score Comparison
GP25 versus qQ
MAFFT Score Comparison
GP25 versus qQ
Muscle3.7 Score Comparison
GP25 versus qQ
ProbCons Score Comparison
GP25 versus qQ
ClustalW2.0 Score Comparison
GP25 versus fM
MAFFT Score Comparison
GP25 versus fM
Muscle3.7 Score Comparison
GP25 versus fM
ProbCons Score Comparison
GP25 versus fM
bal
pec
qQ
fM
GP25 score

Figure B.2

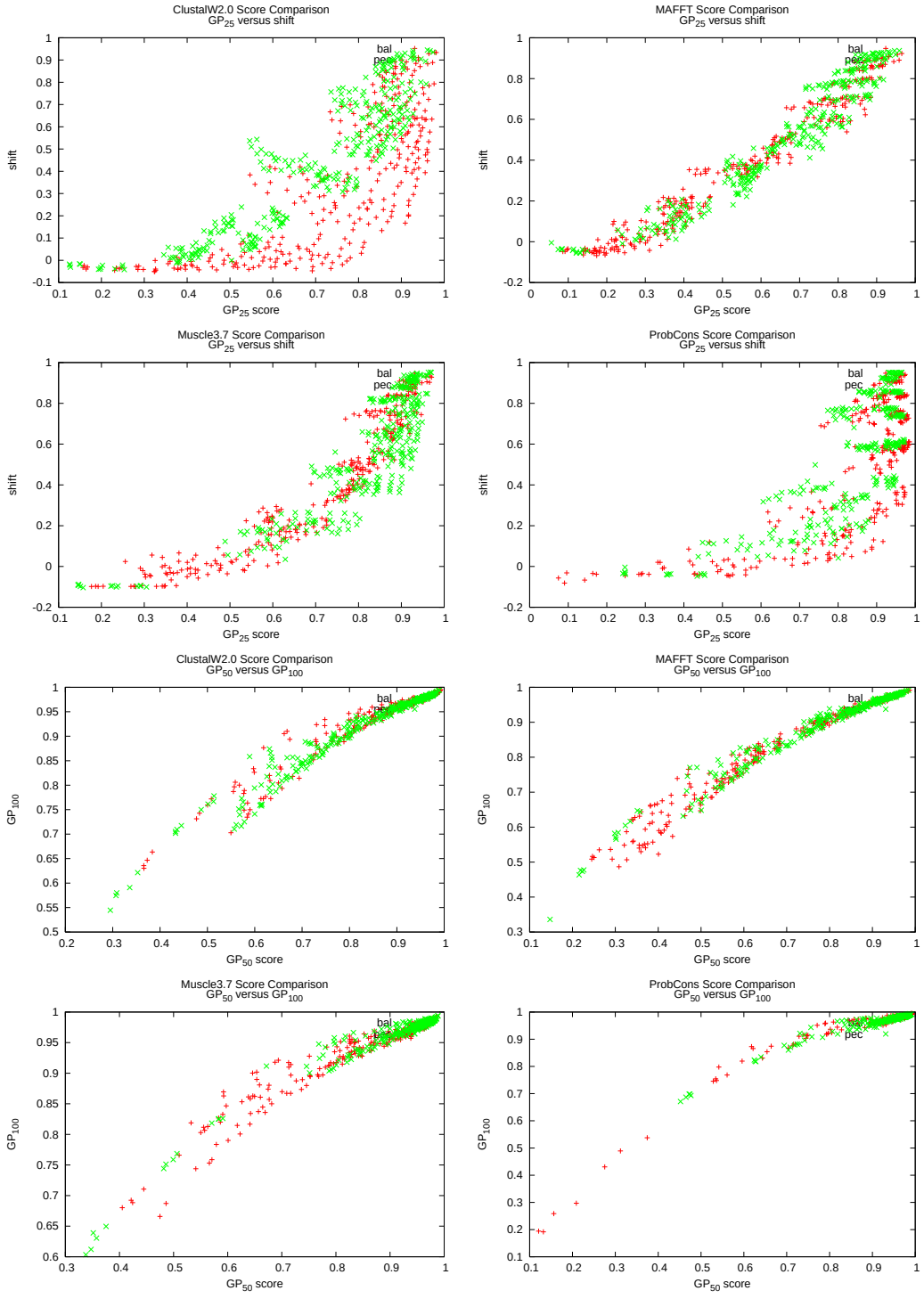

ClustalW2.0 Score Comparison
GP25 versus shift
MAFFT Score Comparison
GP25 versus shift
Muscle3.7 Score Comparison
GP25 versus shift
ProbCons Score Comparison
GP25 versus shift
ClustalW2.0 Score Comparison
GP50 versus GP100
MAFFT Score Comparison
GP50 versus GP100
Muscle3.7 Score Comparison
GP50 versus GP100
ProbCons Score Comparison
GP50 versus GP100
bal
pec
shift
GP25 score
GP100
GP50 score

Figure B.2

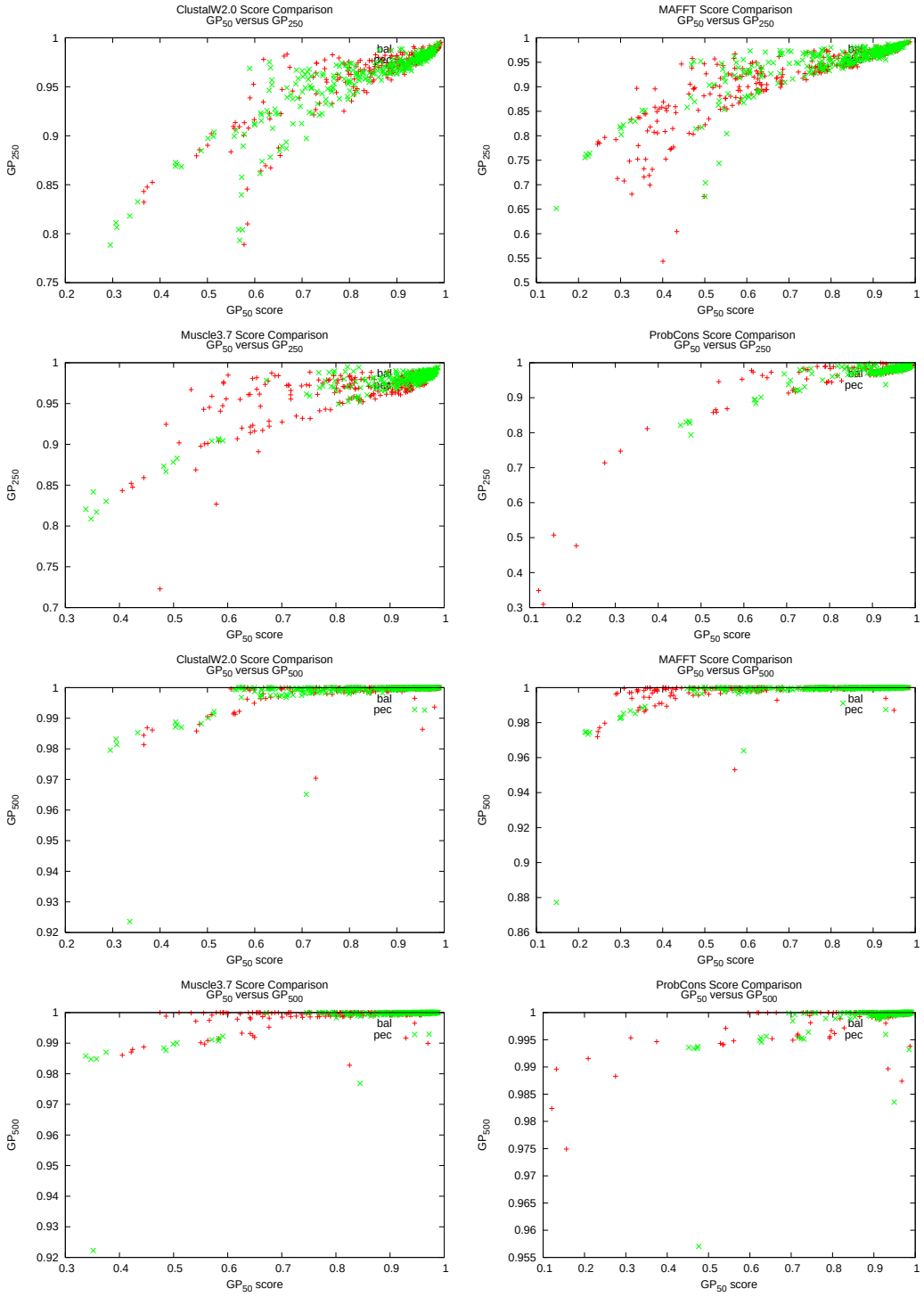

ClustalW2.0 Score Comparison
GP50 versus GP250
MAFFT Score Comparison
GP50 versus GP250
Muscle3.7 Score Comparison
GP50 versus GP250
ProbCons Score Comparison
GP50 versus GP250
ClustalW2.0 Score Comparison
GP50 versus GP500
MAFFT Score Comparison
GP50 versus GP500
Muscle3.7 Score Comparison
GP50 versus GP500
ProbCons Score Comparison
GP50 versus GP500
bal
pec
GP250
GP500
GP50 score

Figure B.2

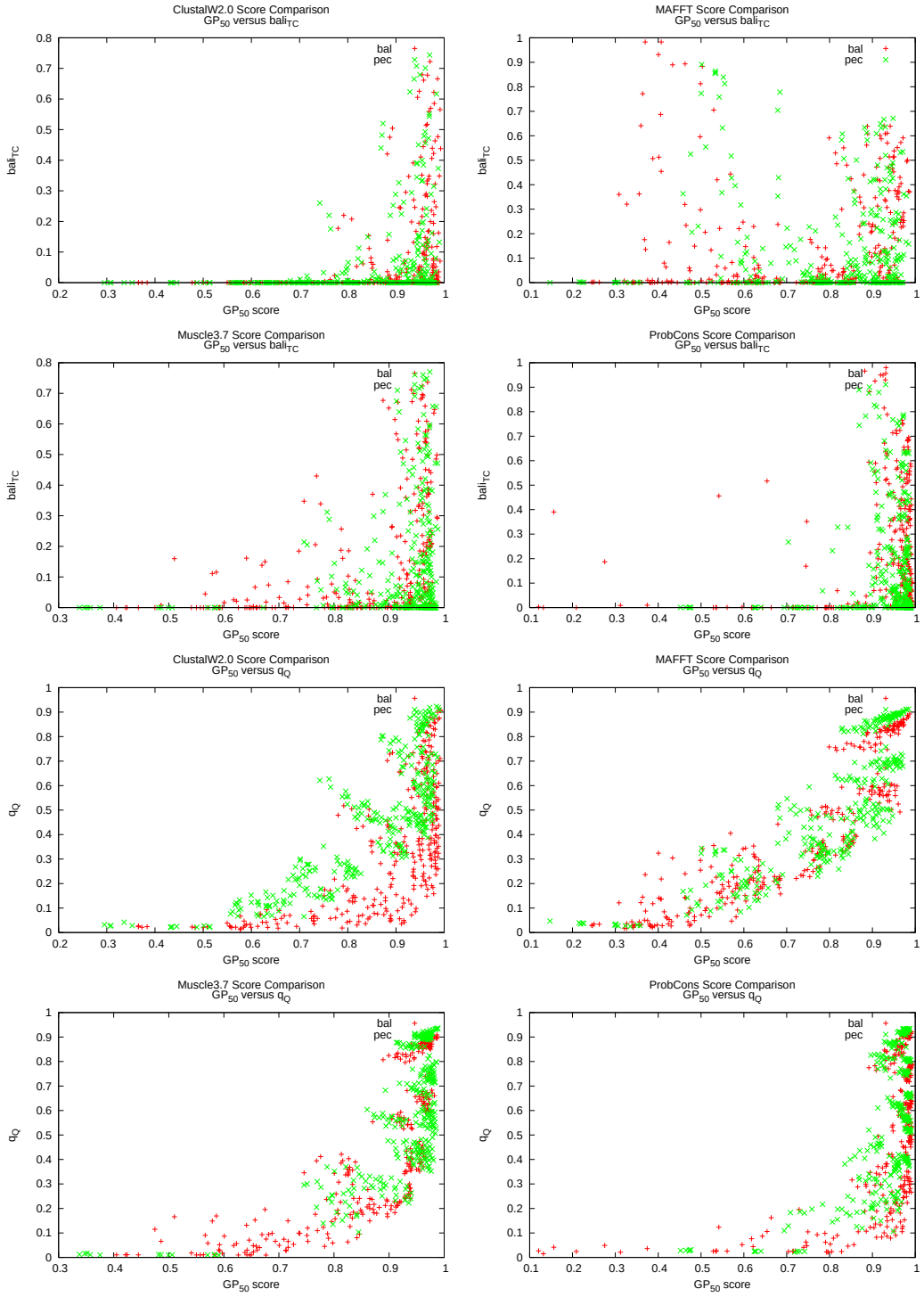

ClustalW2.0 Score Comparison
GP50 versus baliTC
MAFFT Score Comparison
GP50 versus baliTC
Muscle3.7 Score Comparison
GP50 versus baliTC
ProbCons Score Comparison
GP50 versus baliTC
ClustalW2.0 Score Comparison
GP50 versus qQ
MAFFT Score Comparison
GP50 versus qQ
Muscle3.7 Score Comparison
GP50 versus qQ
ProbCons Score Comparison
GP50 versus qQ
bal
pec
baliTC
qQ
GP50 score

Figure B.2

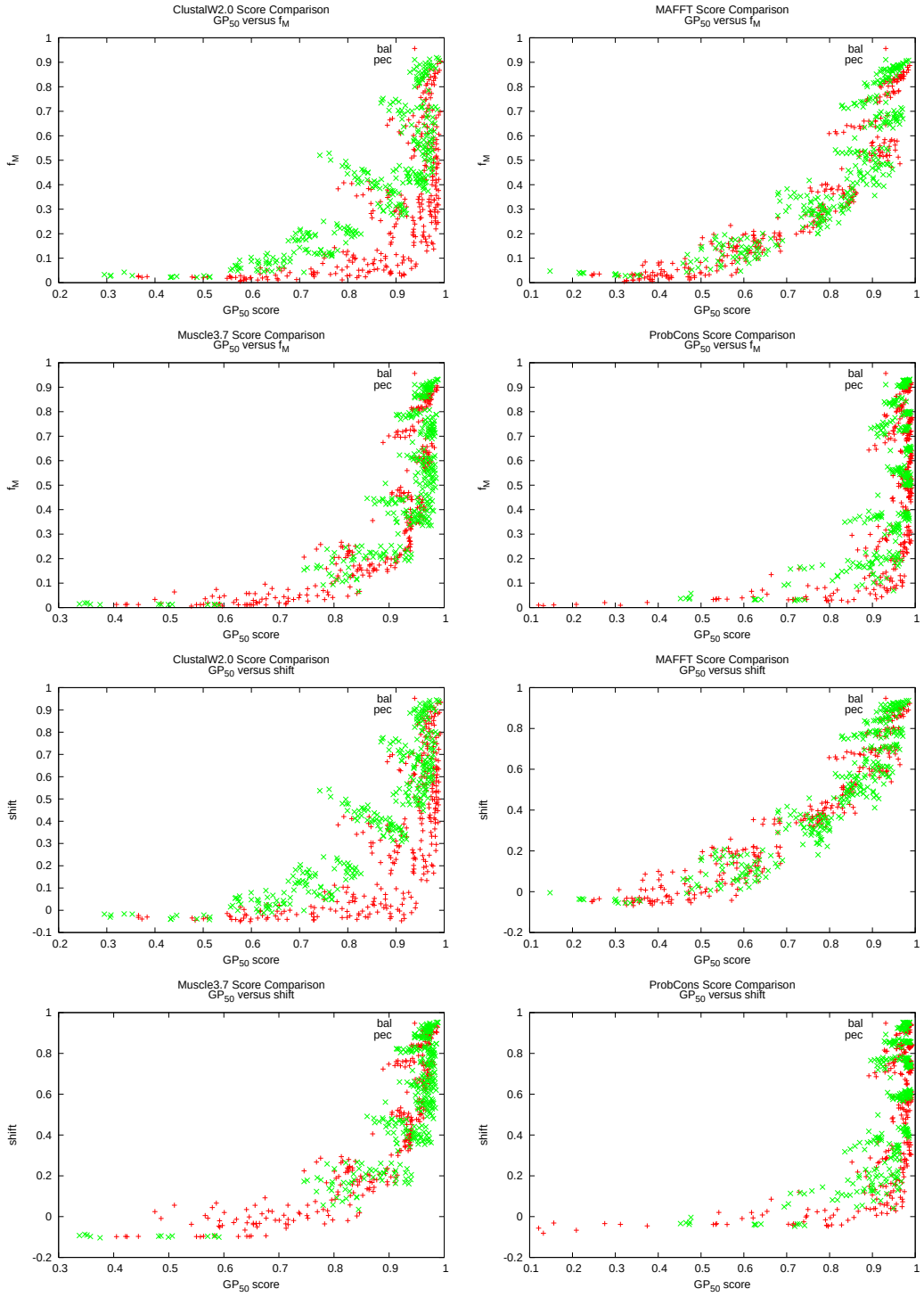

ClustalW2.0 Score Comparison
GP50 versus fM
MAFFT Score Comparison
GP50 versus fM
Muscle3.7 Score Comparison
GP50 versus fM
ProbCons Score Comparison
GP50 versus fM
ClustalW2.0 Score Comparison
GP50 versus shift
MAFFT Score Comparison
GP50 versus shift
Muscle3.7 Score Comparison
GP50 versus shift
ProbCons Score Comparison
GP50 versus shift
bal
pec
fM
shift
GP50 score

Figure B.2

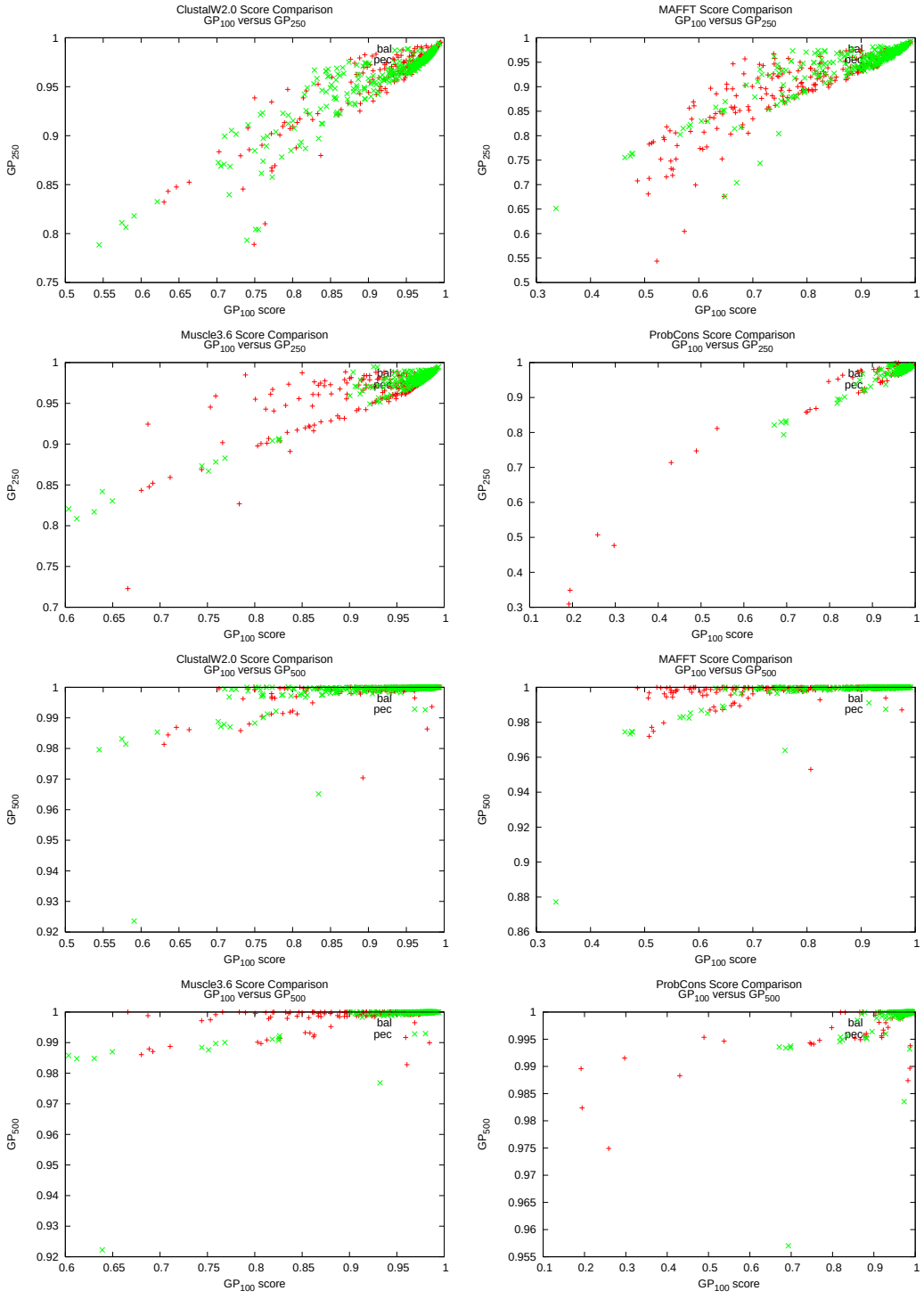

ClustalW2.0 Score Comparison
GP100 versus GP250
MAFFT Score Comparison
GP100 versus GP250
Muscle3.6 Score Comparison
GP100 versus GP250
ProbCons Score Comparison
GP100 versus GP250
ClustalW2.0 Score Comparison
GP100 versus GP500
MAFFT Score Comparison
GP100 versus GP500
Muscle3.6 Score Comparison
GP100 versus GP500
ProbCons Score Comparison
GP100 versus GP500
GP250
GP500
GP100 score
bal
pec

Figure B.2

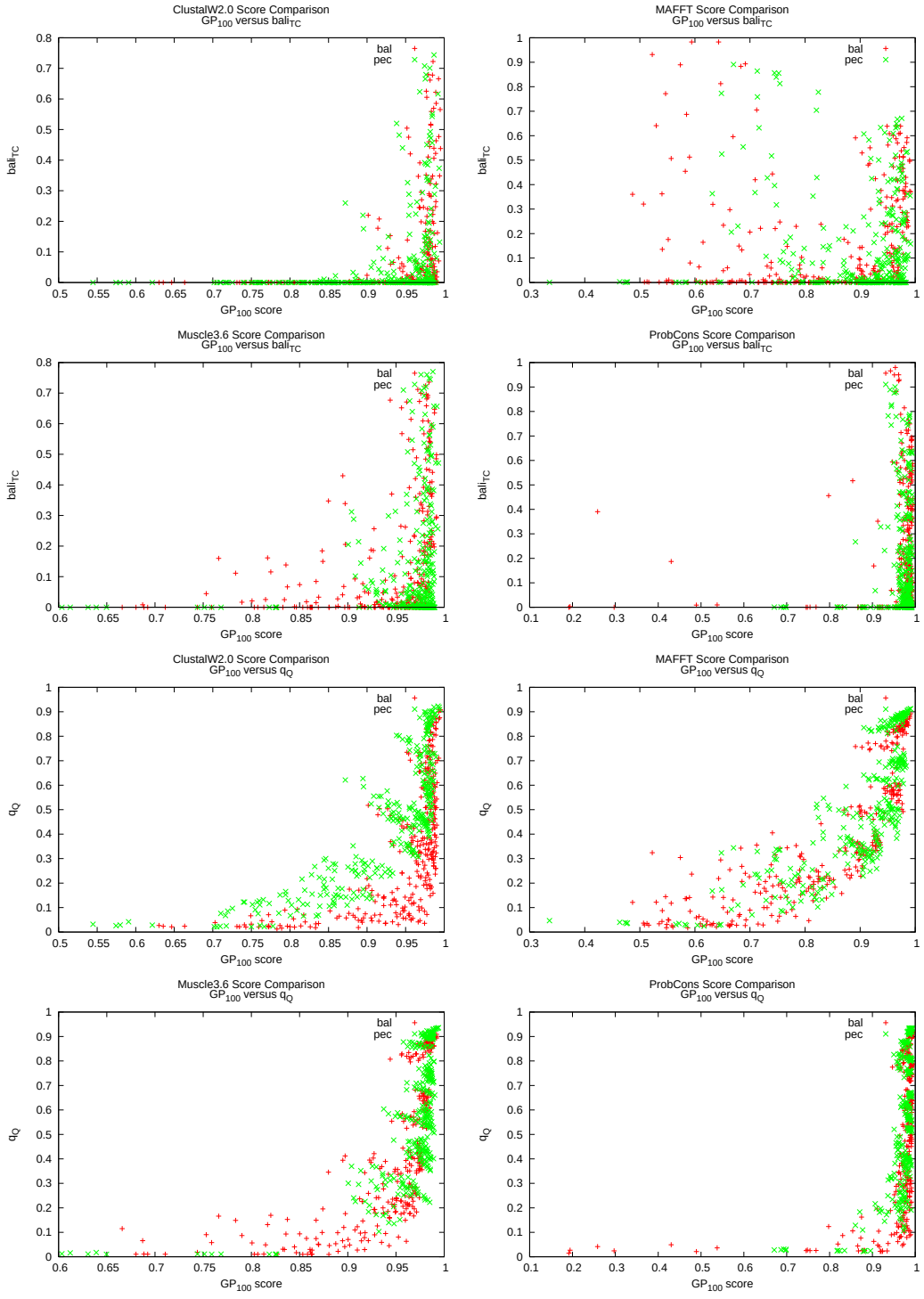

ClustalW2.0 Score Comparison
GP100 versus balTC
MAFFT Score Comparison
GP100 versus balTC
Muscle3.6 Score Comparison
GP100 versus balTC
ProbCons Score Comparison
GP100 versus balTC
ClustalW2.0 Score Comparison
GP100 versus qQ
MAFFT Score Comparison
GP100 versus qQ
Muscle3.6 Score Comparison
GP100 versus qQ
ProbCons Score Comparison
GP100 versus qQ
bal
pec
balTC
qQ
GP100 score

Figure B.2

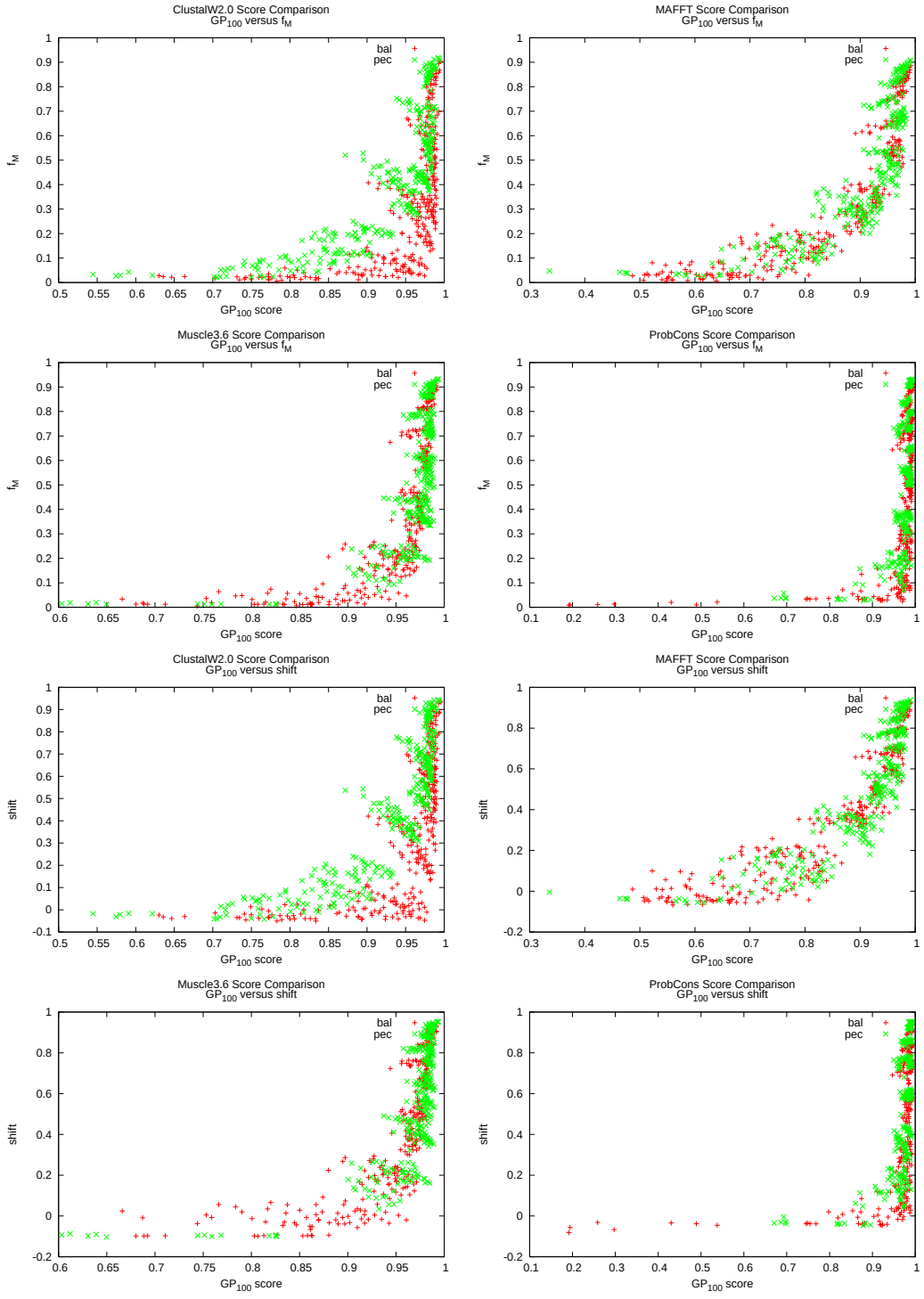

ClustalW2.0 Score Comparison
GP100 versus fM
MAFFT Score Comparison
GP100 versus fM
Muscle3.6 Score Comparison
GP100 versus fM
ProbCons Score Comparison
GP100 versus fM
ClustalW2.0 Score Comparison
GP100 versus shift
MAFFT Score Comparison
GP100 versus shift
Muscle3.6 Score Comparison
GP100 versus shift
ProbCons Score Comparison
GP100 versus shift
bal
pec
fM
shift
GP100 score

Figure B.2

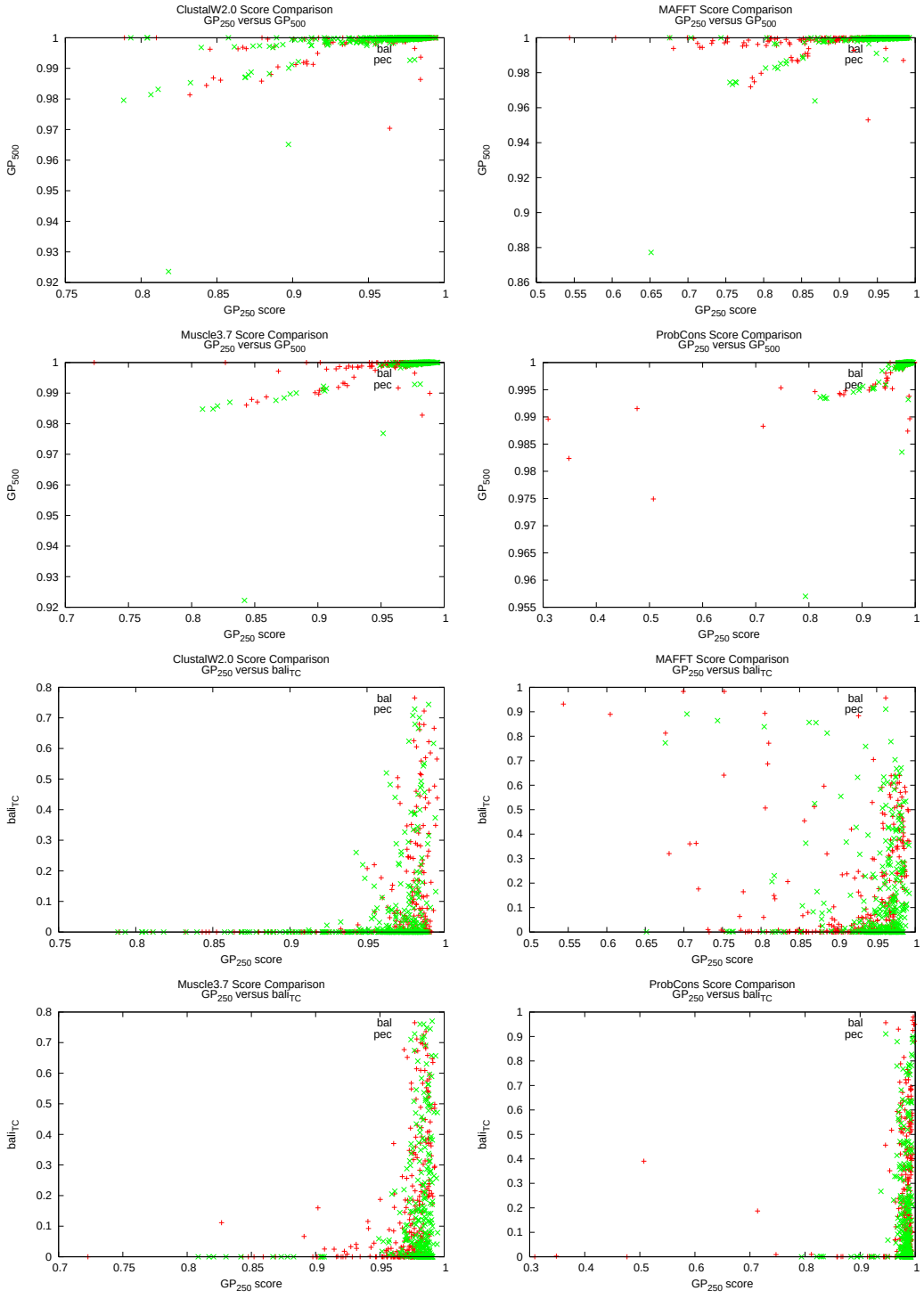

ClustalW2.0 Score Comparison
GP250 versus GP500
MAFFT Score Comparison
GP250 versus GP500
Muscle3.7 Score Comparison
GP250 versus GP500
ProbCons Score Comparison
GP250 versus GP500
ClustalW2.0 Score Comparison
GP250 versus balTC
MAFFT Score Comparison
GP250 versus balTC
Muscle3.7 Score Comparison
GP250 versus balTC
ProbCons Score Comparison
GP250 versus balTC
bal
pec
GP500
balTC
GP250 score

Figure B.2

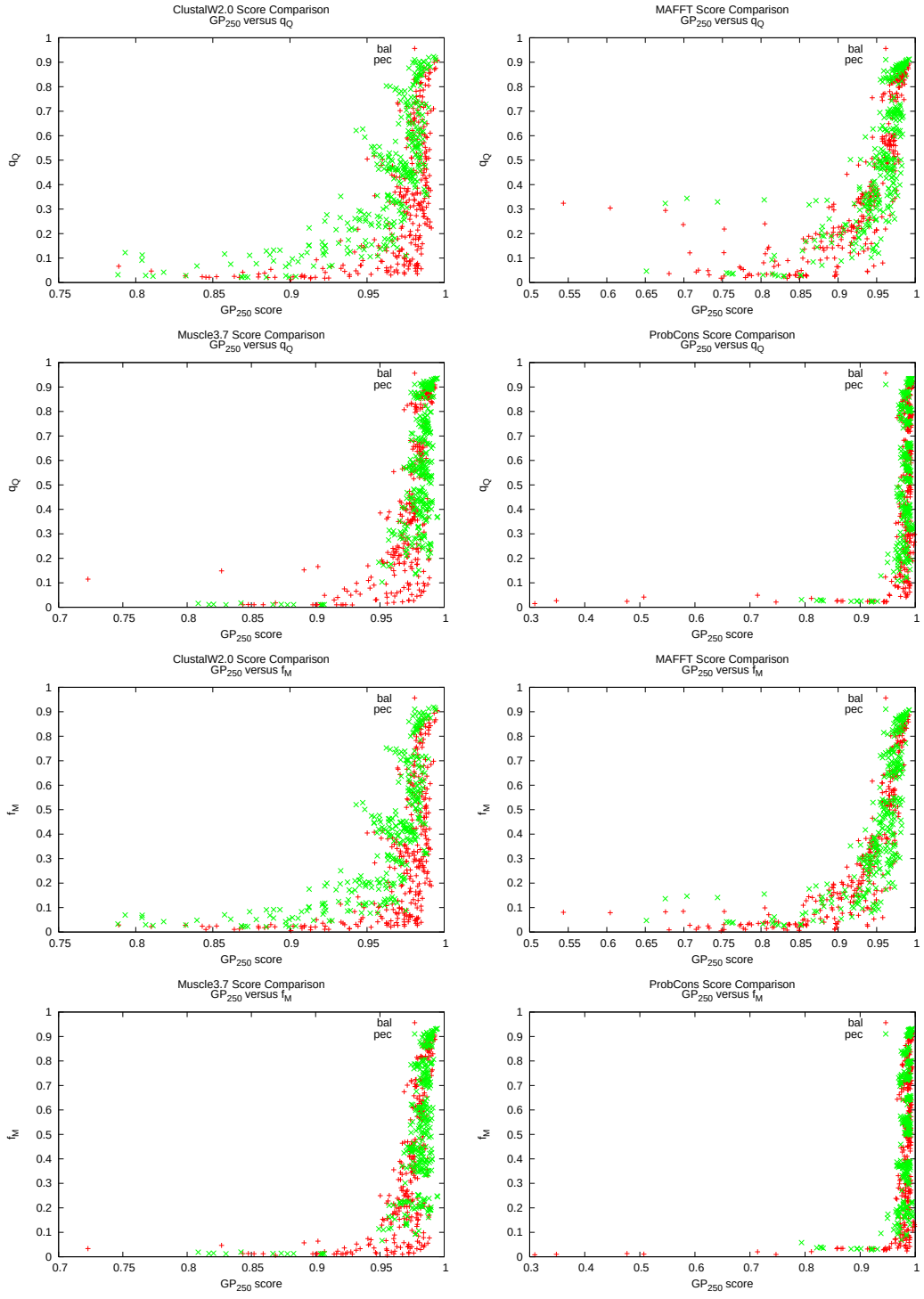

ClustalW2.0 Score Comparison
GP250 versus qQ
MAFFT Score Comparison
GP250 versus qQ
Muscle3.7 Score Comparison
GP250 versus qQ
ProbCons Score Comparison
GP250 versus qQ
ClustalW2.0 Score Comparison
GP250 versus fM
MAFFT Score Comparison
GP250 versus fM
Muscle3.7 Score Comparison
GP250 versus fM
ProbCons Score Comparison
GP250 versus fM
bal
pec
qQ
fM
GP250 score

Figure B.2

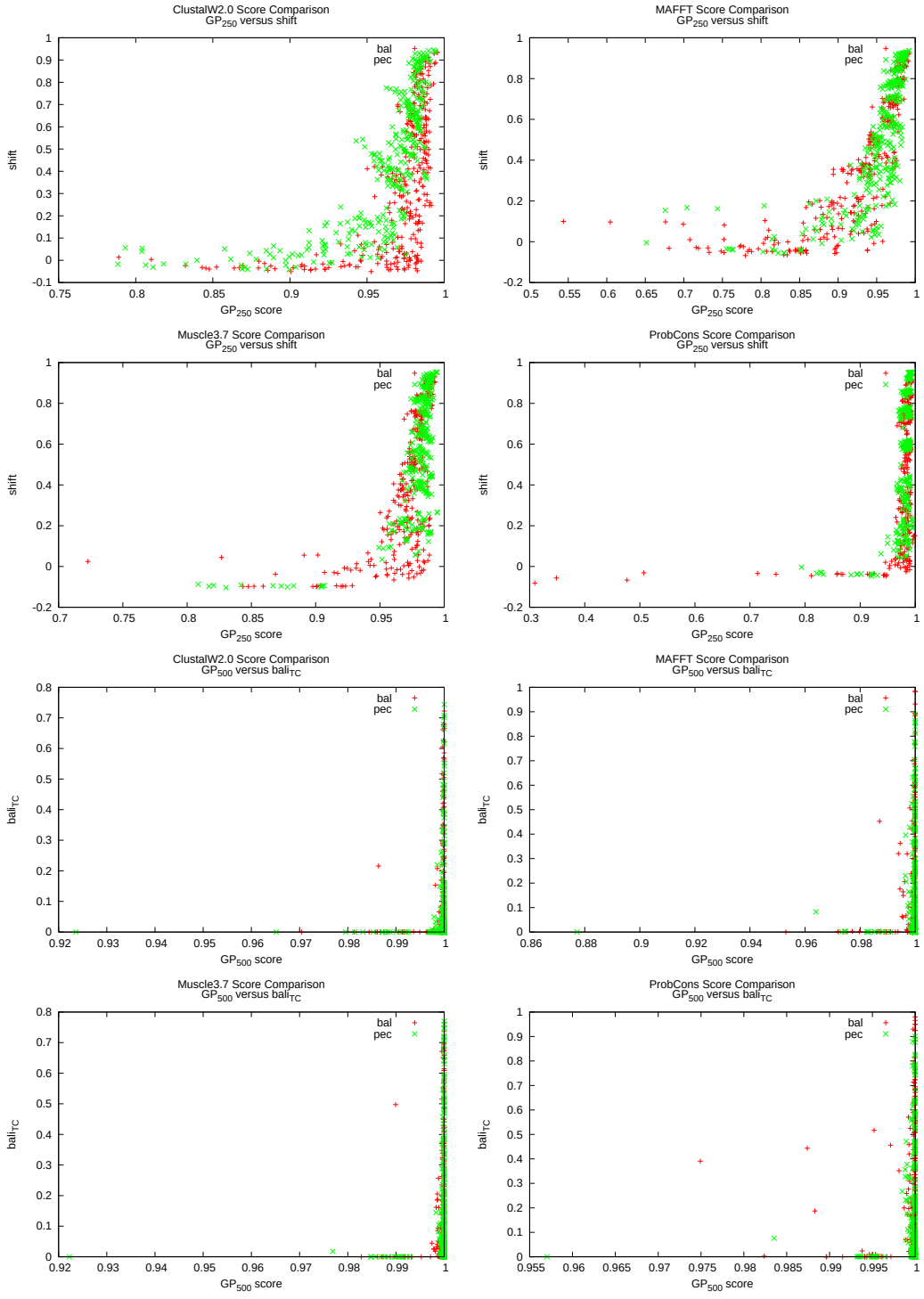

ClustalW2.0 Score Comparison
GP250 versus shift
MAFFT Score Comparison
GP250 versus shift
Muscle3.7 Score Comparison
GP250 versus shift
ProbCons Score Comparison
GP250 versus shift
ClustalW2.0 Score Comparison
GP500 versus balTC
MAFFT Score Comparison
GP500 versus balTC
Muscle3.7 Score Comparison
GP500 versus balTC
ProbCons Score Comparison
GP500 versus balTC
bal
pec
shift
balTC
GP250 score
GP500 score

Figure B.2

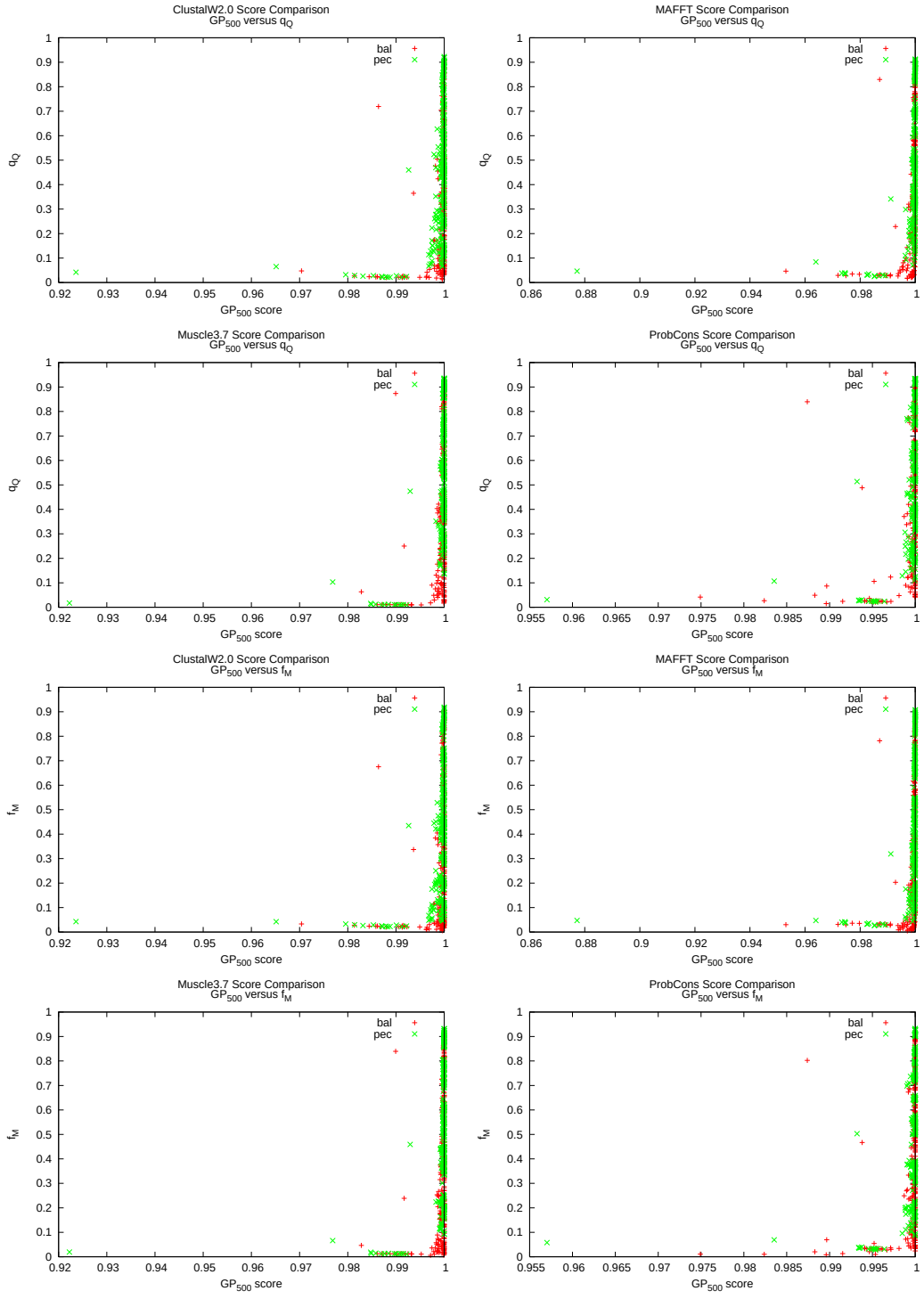

ClustalW2.0 Score Comparison
GP500 versus qQ
MAFFT Score Comparison
GP500 versus qQ
Muscle3.7 Score Comparison
GP500 versus qQ
ProbCons Score Comparison
GP500 versus qQ
ClustalW2.0 Score Comparison
GP500 versus fM
MAFFT Score Comparison
GP500 versus fM
Muscle3.7 Score Comparison
GP500 versus fM
ProbCons Score Comparison
GP500 versus fM
bal
pec
qQ
fM
GP500 score

Figure B.2

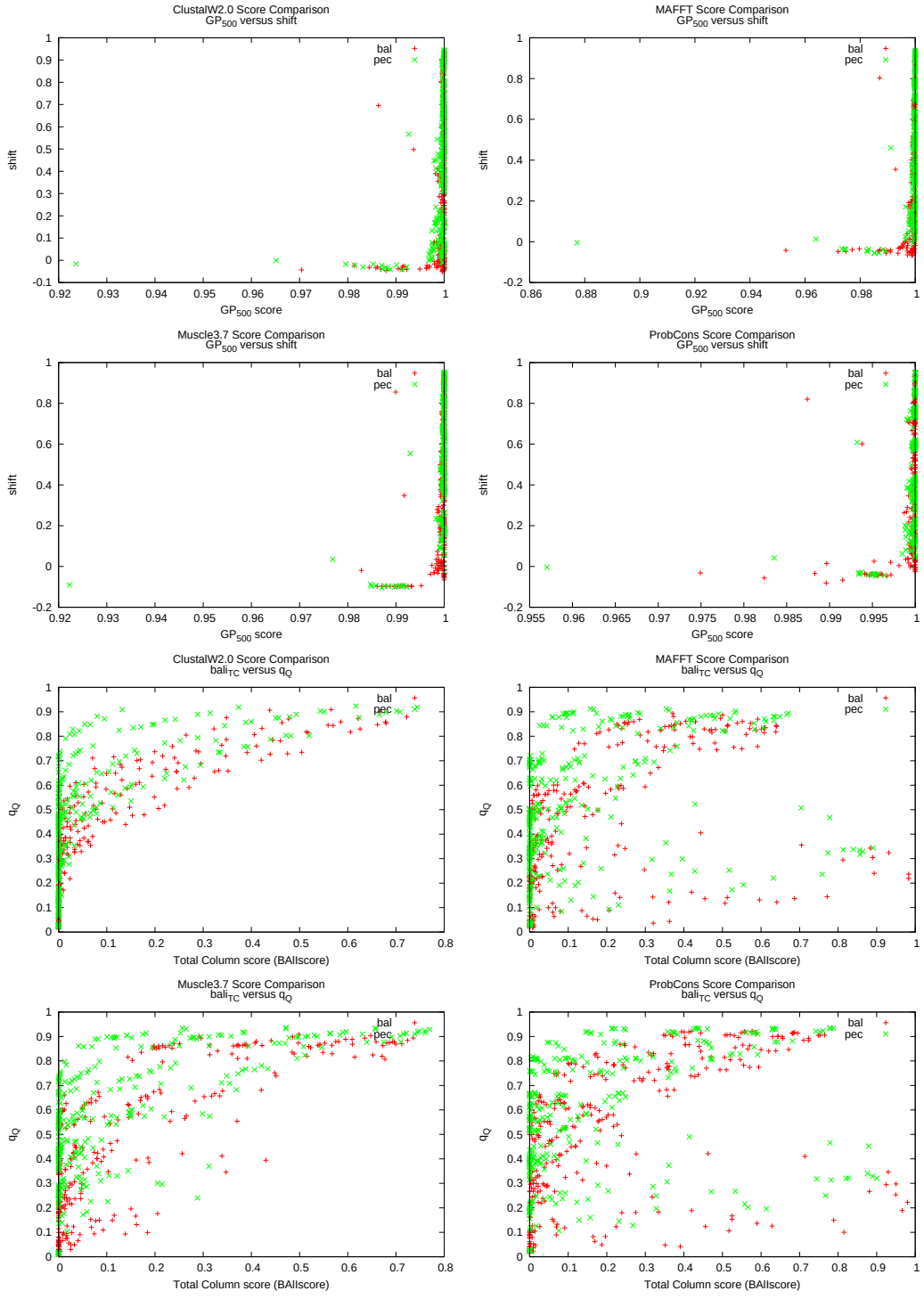

ClustalW2.0 Score Comparison
GP500 versus shift
MAFFT Score Comparison
GP500 versus shift
Muscle3.7 Score Comparison
GP500 versus shift
ProbCons Score Comparison
GP500 versus shift
ClustalW2.0 Score Comparison
balTC versus qQ
MAFFT Score Comparison
balTC versus qQ
Muscle3.7 Score Comparison
balTC versus qQ
ProbCons Score Comparison
balTC versus qQ
bal
pec
shift
GP500 score
qQ
Total Column score (BAIIscore)

Figure B.2

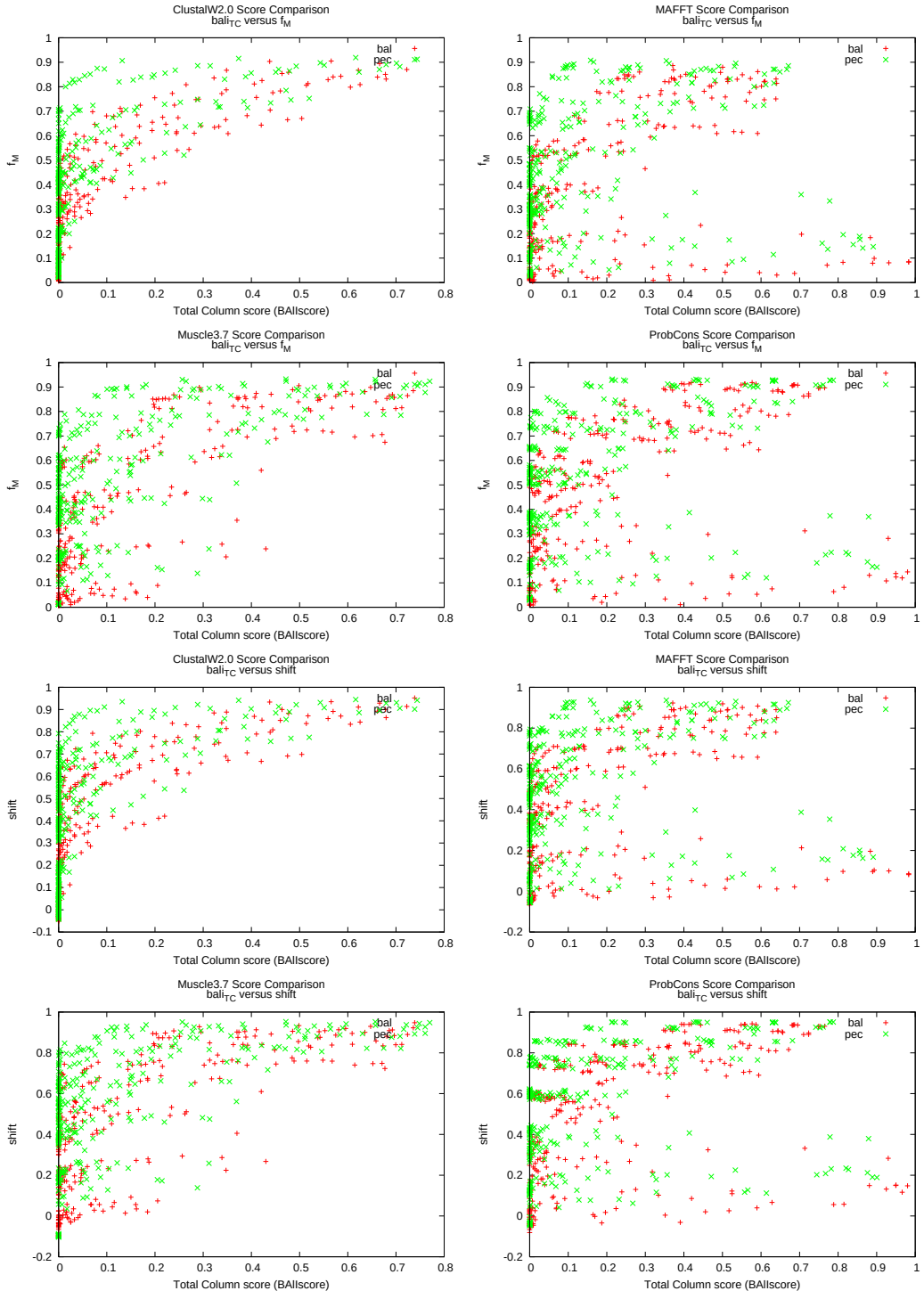

ClustalW2.0 Score Comparison
bali_TC versus f_M
MAFFT Score Comparison
bali_TC versus f_M
Muscle3.7 Score Comparison
bali_TC versus f_M
ProbCons Score Comparison
bali_TC versus f_M
ClustalW2.0 Score Comparison
bali_TC versus shift
MAFFT Score Comparison
bali_TC versus shift
Muscle3.7 Score Comparison
bali_TC versus shift
ProbCons Score Comparison
bali_TC versus shift
bal
pec
f_M
shift
Total Column score (BAIIscore)

Figure B.2

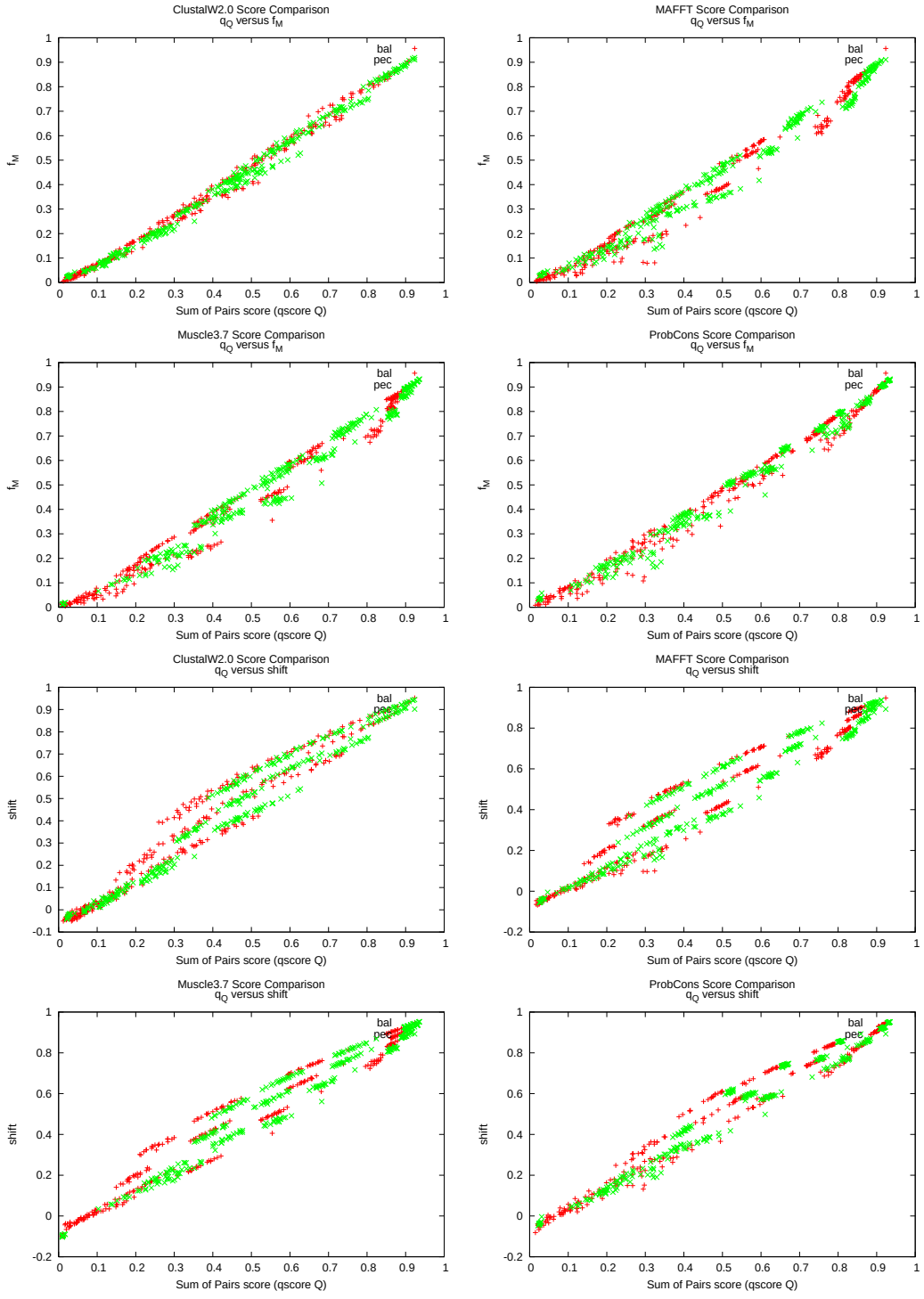

ClustalW2.0 Score Comparison
$q_Q$ versus $f_M$
MAFFT Score Comparison
$q_Q$ versus $f_M$
Muscle3.7 Score Comparison
$q_Q$ versus $f_M$
ProbCons Score Comparison
$q_Q$ versus $f_M$
ClustalW2.0 Score Comparison
$q_Q$ versus shift
MAFFT Score Comparison
$q_Q$ versus shift
Muscle3.7 Score Comparison
$q_Q$ versus shift
ProbCons Score Comparison
$q_Q$ versus shift
bal
pec
$f_M$
shift
Sum of Pairs score (qscore Q)

Figure B.2

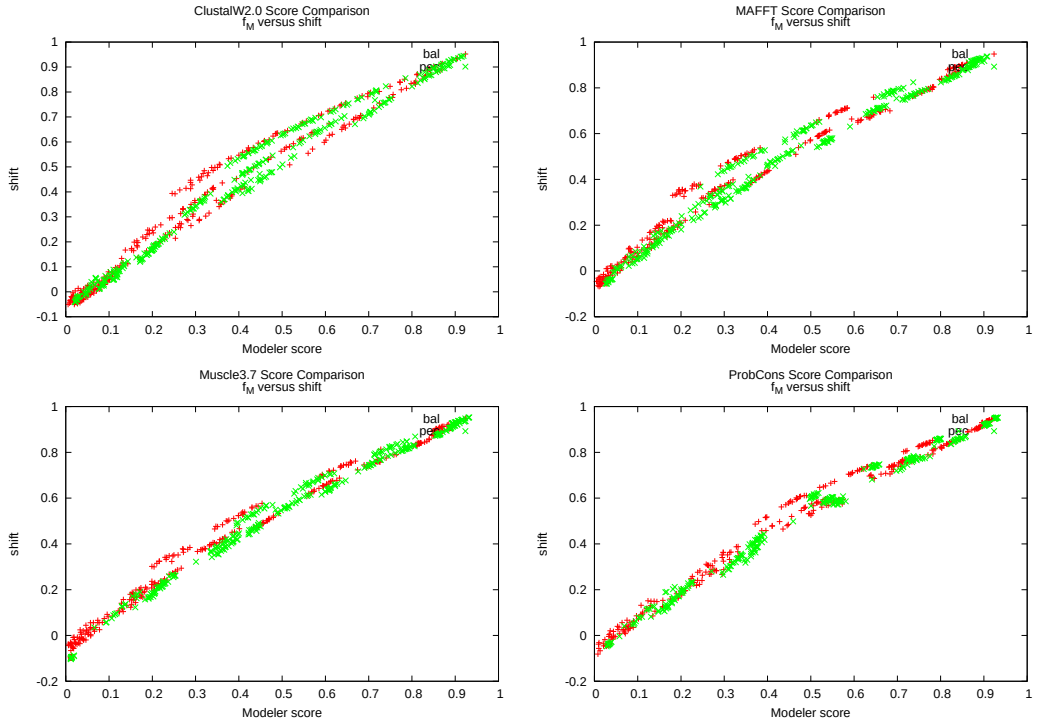



Figure B.2

Figure B.2: Side by side comparison of scoring methods, colored by benchmark statistics.

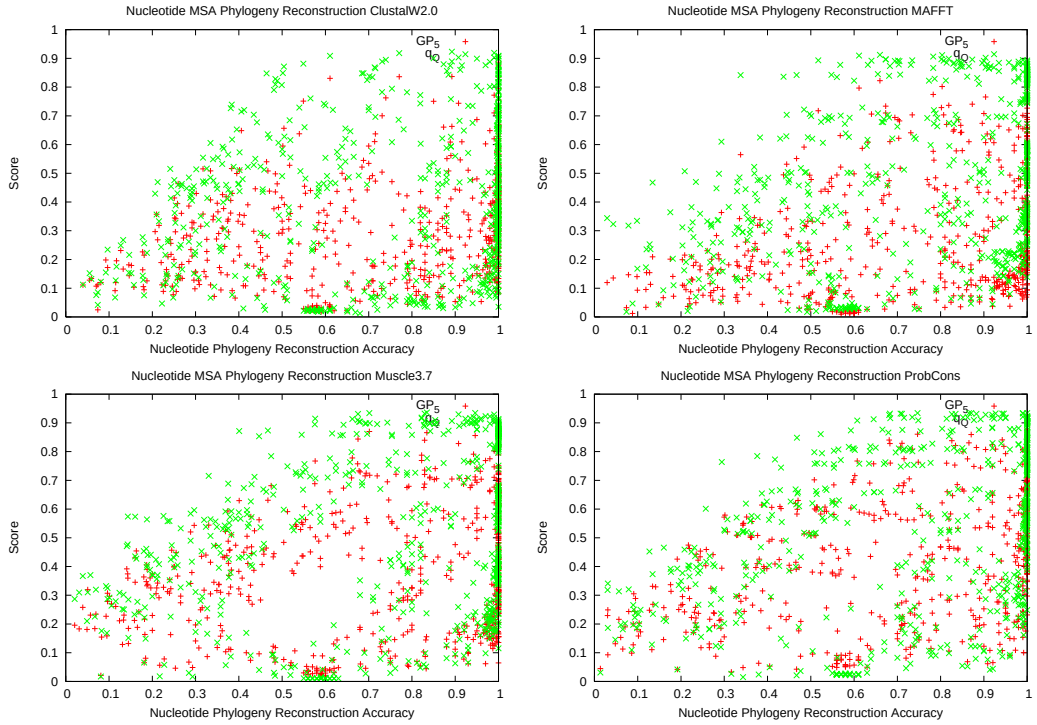



Figure B.3

## B.3 Metric Scores against Nucleotide Phylogeny Reconstruction

This section compares pairwise indel-based metrics against pairwise character-based MSA metric against the accuracy of the maximum likelihood phylogenetic trees reconstructed..

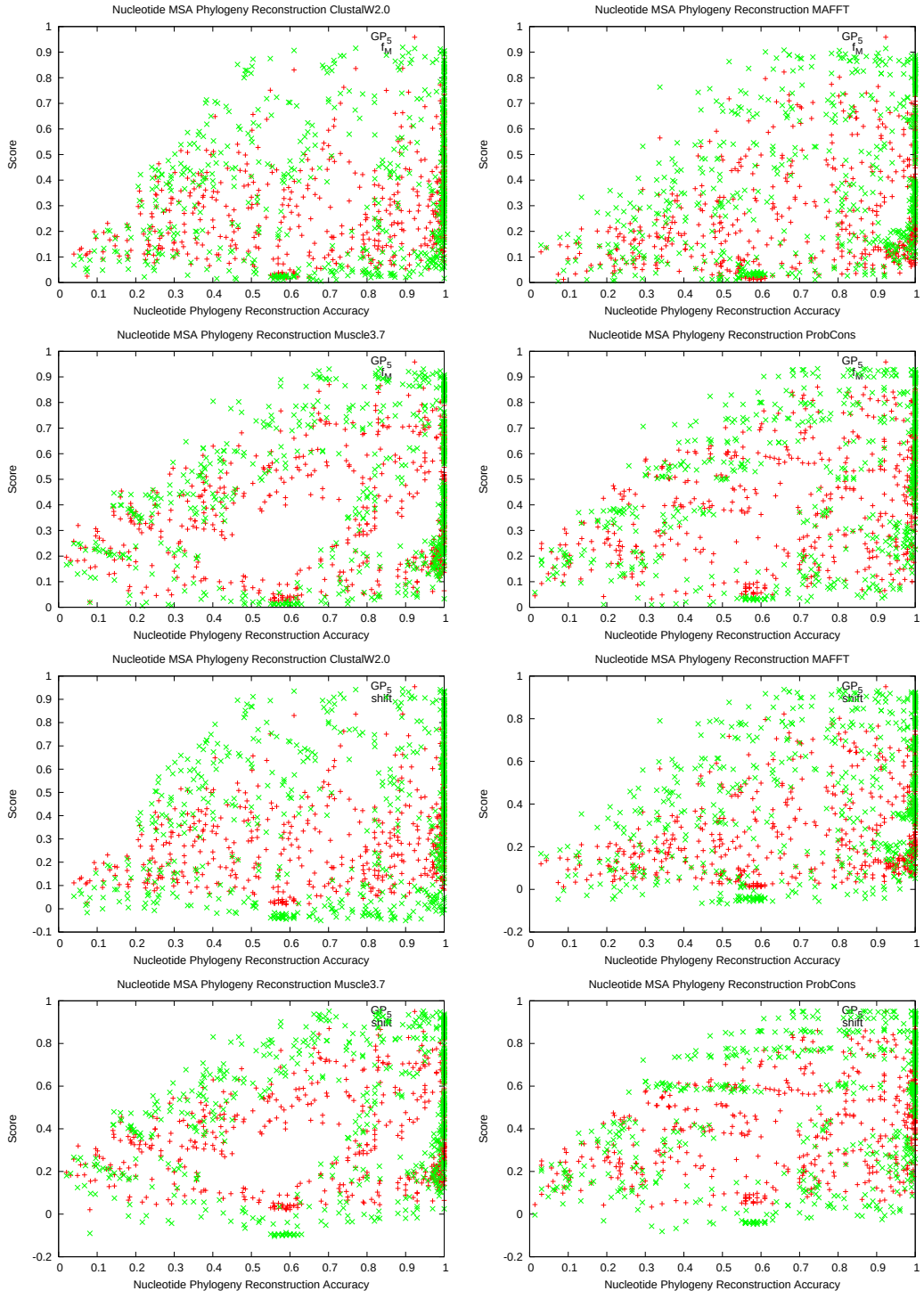

Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
GP5 fM
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction MAFFT
GP5 fM
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
GP5 fM
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction ProbCons
GP5 fM
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
GP5 shift
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction MAFFT
GP5 shift
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
GP5 shift
Score
Nucleotide Phylogeny Reconstruction Accuracy
Nucleotide MSA Phylogeny Reconstruction ProbCons
GP5 shift
Score
Nucleotide Phylogeny Reconstruction Accuracy

Figure B.3

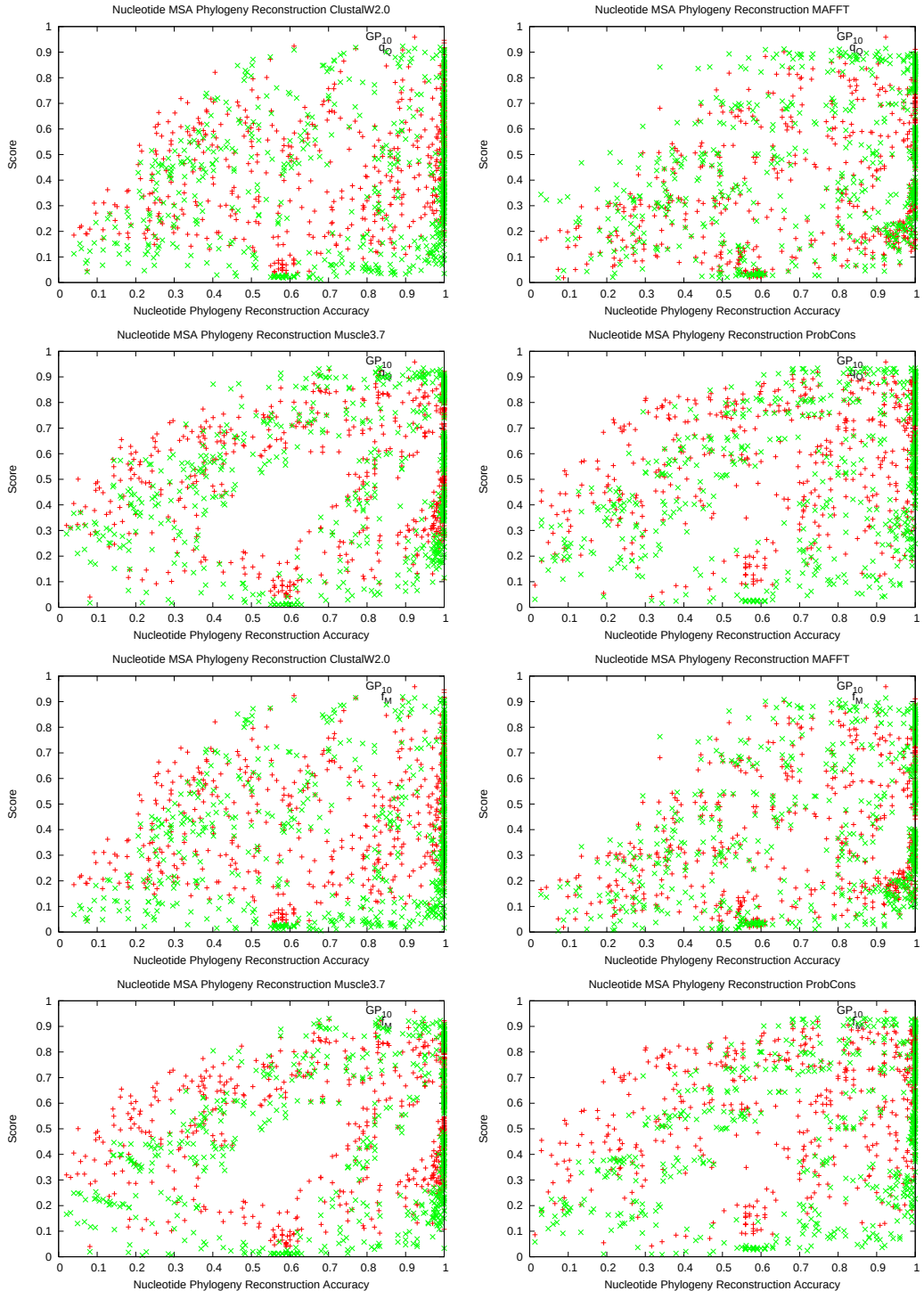

Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
Nucleotide MSA Phylogeny Reconstruction MAFFT
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
Nucleotide MSA Phylogeny Reconstruction ProbCons
Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
Nucleotide MSA Phylogeny Reconstruction MAFFT
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
Nucleotide MSA Phylogeny Reconstruction ProbCons
GP10 qO
GP10 fM
Score
Nucleotide Phylogeny Reconstruction Accuracy

Figure B.3

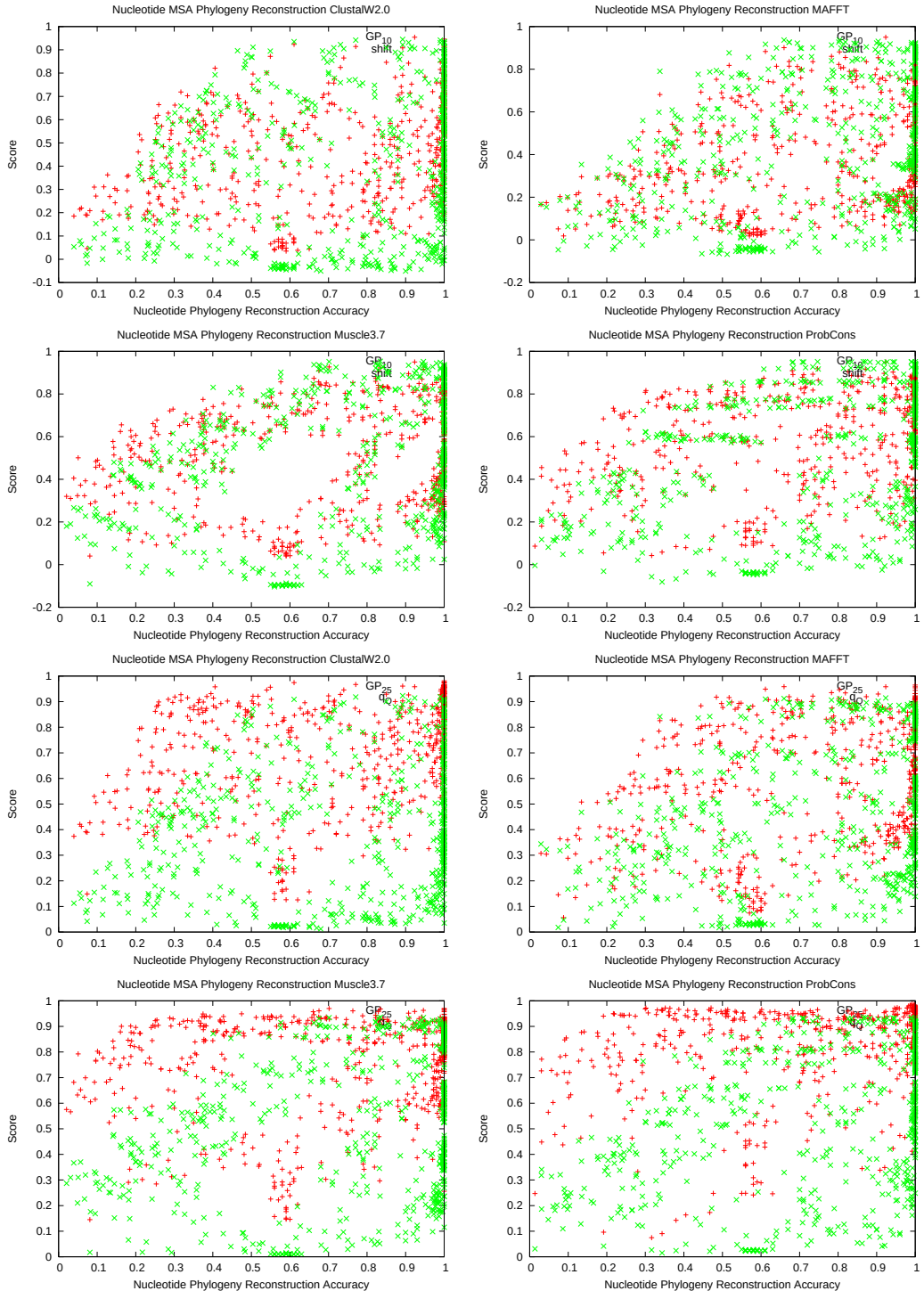

Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
Nucleotide MSA Phylogeny Reconstruction MAFFT
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
Nucleotide MSA Phylogeny Reconstruction ProbCons
Nucleotide MSA Phylogeny Reconstruction ClustalW2.0
Nucleotide MSA Phylogeny Reconstruction MAFFT
Nucleotide MSA Phylogeny Reconstruction Muscle3.7
Nucleotide MSA Phylogeny Reconstruction ProbCons
Score
Nucleotide Phylogeny Reconstruction Accuracy
GP10 shift
GP25 dO

Figure B.3

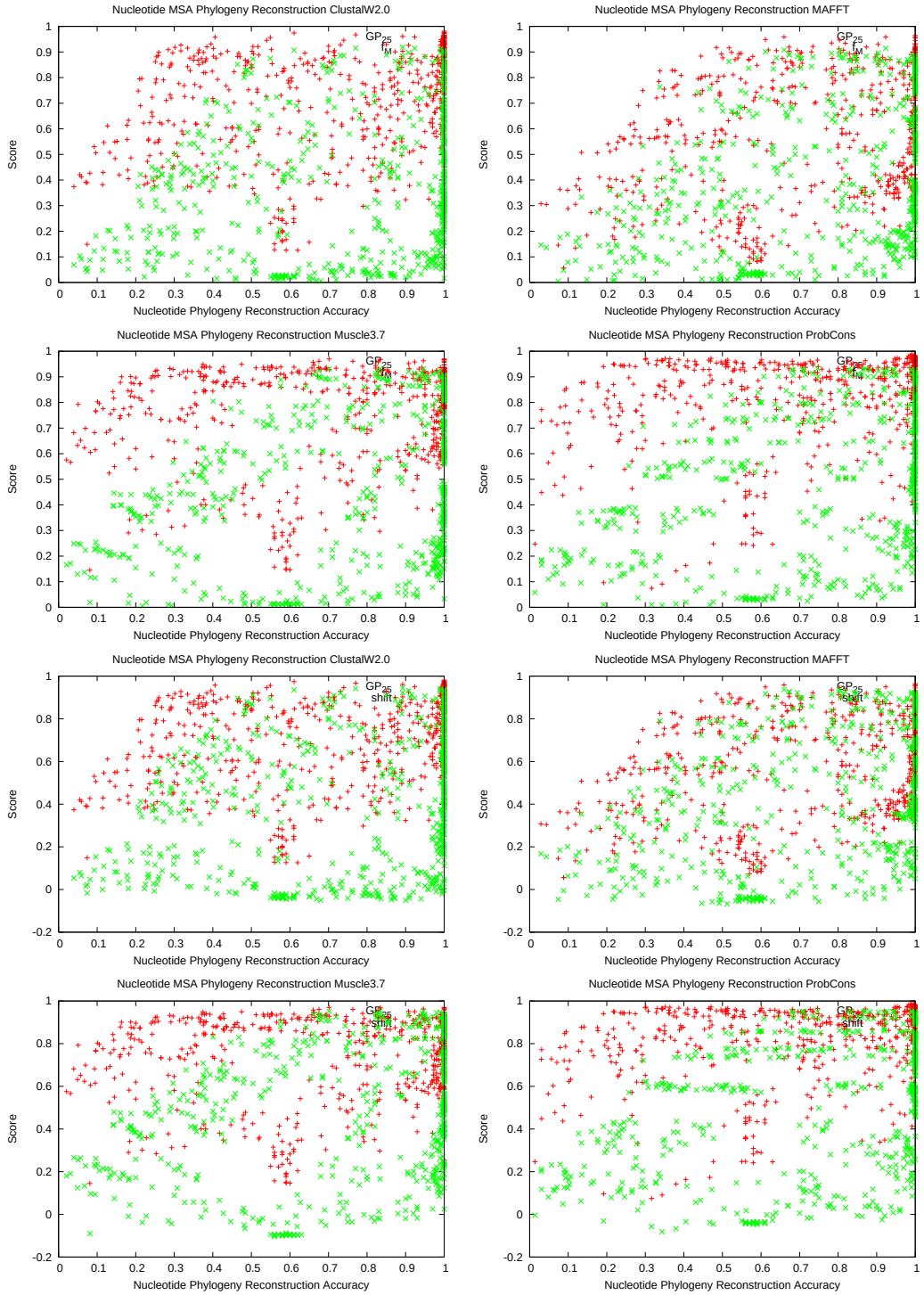



Figure B.3: Comparison of scoring metric values against phylogenetic topological accuracy using Nucleotide reconstructed MSAs.