Application of Lempel-Ziv complexity to alignment-free sequence comparison of protein families

Sofiène Bacha[1] **and Denis Baurain**[2]

[1] Dept of Electrical Engineering and Computer Science, Montefiore Institute B28, University of Liège, B-4000 Liège.
[2] Dept of Life Sciences, Institute of Botany B22, University of Liège, B-4000 Liège.
bacha_sofiene@yahoo.fr, denis.baurain@ulg.ac.be

MOST sequence comparison methods require an alignment to work on. Although efficient algorithms are readily available, sequence alignment remains difficult and often requires human intervention, which can lead to biased results.

Two main categories of alignment-free methods have been proposed to overcome the limitations of the alignment-based sequence comparisons (reviewed by Vinga and Almeida, 2003). The first category is founded on the statistics of word frequency, whereas the second category includes methods that do not require resolving the sequence with fixed word length segments. Among the latter, methods based on information theory make use of the algorithmic complexity (estimated through sequence compression) as the distance metric. Along the same lines, Otu and Sayood (2003) have proposed to rely on the Lempel-Ziv complexity to compute the distance between two DNA sequences.

Because it is based on exact direct repeats, the LZ complexity works well with the small DNA alphabet. However, when applied to protein sequences, such an approach is expected to miss the subtle and overlapping similarities that characterize the larger and more complex amino-acid alphabet.

In this work, we present several variants of a simple strategy in which protein sequences are encoded to a new alphabet prior to computation of the LZ complexity. The key idea is to capture as much information as possible in order to enhance the phylogenetic performance of the method when applied to proteins.

We then evaluate the usefulness of our proposals by comparing their results against both word statistics methods and alignment-based similarity measures in the context of the recognition of SCOP/ASTRAL relationships as described by Vinga et al. (2004). Furthermore, we examine their ability to infer evolutionary history by applying them to the phylogeny of a family of metal transporters, the HMA subfamily of P-type ATPases.

Methods

LZ complexity (Lempel and Ziv, 1976)

Let us loosely define the *exhaustive history* of a sequence S, $H_E(S)$, as the decomposition of S into a number of components such as each component results (except maybe the last one) from the direct copy of the longest possible preceding substring of S followed by a single letter innovation. It can be shown that the *LZ complexity* of S, denoted $c(S)$, is $c(S) = c_E(S)$, where $c_E(S)$ is the number of components in the exhaustive history of S. For example, for $S = AACGTACCATTG$, the exhaustive history is $H_E(S) = A{\cdot}AC{\cdot}G{\cdot}T{\cdot}ACC{\cdot}AT{\cdot}TG$, while for $Q = ACGGTCACCAA$, the exhaustive history is $H_E(Q) = A \cdot C \cdot G \cdot GT \cdot CA \cdot CC \cdot AA$. Consequently, both $c(S)$ and $c(Q)$ will be equal to 7.

Given two sequences S and Q, consider the sequence SQ and its exhaustive history. Intuitively, one can see that the number of components needed to build Q when appended to S will be less than or equal to the number of components needed to build Q alone because at every step of the *production process* of Q, the search space will be larger due to the existence of S. This is known as the *subadditivity* of the LZ complexity: $c(SQ) \leq c(S) + c(Q)$. How much $c(SQ) - c(S)$ is less than $c(Q)$ will depend on the degree of similarity between S and Q. In the case of S and Q shown above, $H_E(SQ) = A \cdot AC \cdot G \cdot T \cdot ACC \cdot AT \cdot TG \cdot ACGG \cdot TC \cdot ACCAA$ and $c(SQ) = 10$, which is indeed four less than $c(S) + c(Q) = 7 + 7 = 14$.

Now, imagine a third sequence, $R = CTAGGGACTTAT$, for which $H_E(R) = C \cdot T \cdot A \cdot G \cdot GGA \cdot CTT \cdot AT$ and $c(R) = 7$. Then, let us compute the exhaustive history of Q when appended to R: $H_E(RQ) = C \cdot T \cdot A \cdot G \cdot GGA \cdot CTT \cdot AT \cdot ACG \cdot GT \cdot CA \cdot CC \cdot AA$. Note that since $c(RQ) = 12$, it took two more steps to build Q from R than from S. This is because S and Q are 'closer', sharing patterns like ACG and $ACCA$. Based on this idea of closeness, Otu and Sayood (2003) define four distance measures. Here, we will use the second one, which is normalized to eliminate the effect of the length on the distance measure:

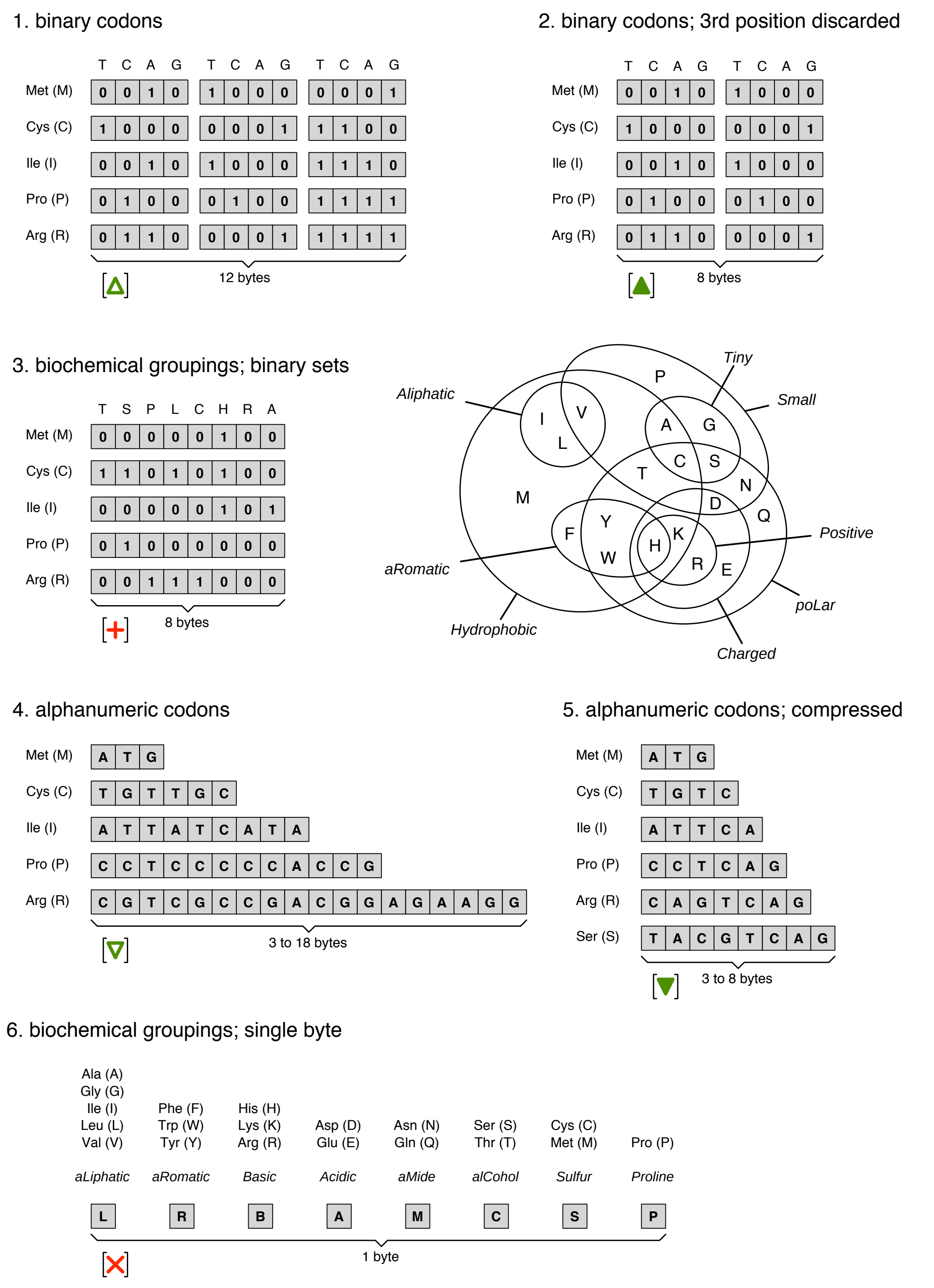
$$d^*(S,Q) = \frac{max\{c(SQ) - c(S), c(QS) - c(Q)\}}{max\{c(S), c(Q)\}}$$

Amino-acid encodings

Since the LZ complexity method does not compare sequence residues on a pairwise basis, the classical transition matrix associated to an evolutionary model cannot be used here. Instead, we propose two different approaches: (i) back-translating sequences to variably degenerate binary (Fig.1.1 and 2) or alphanumeric (Fig.1.4 and 5) codons; and (ii) mapping sequences to strings of binary-coded sets (Fig.1.3) in an attempt to account for the biochemically meaningful grouping of amino-acids. Moreover, we present the results of a destructive encoding where the 20 residues are folded to 8 different symbols (Fig.1.6) to highlight the main chemical group of each side-chain.

Large-scale comparative assessment

The test procedure was an independent re-implementation of the strategy reported by Vinga et al. (2004). Briefly, the distances between 1,683 protein sequences from the SCOP/ASTRAL database (PDB40-v dataset) were computed with 10 different methods: (1) Euclidean distance, (2) W-metric, (3) Smith and Waterman local alignment, and (4 to 10) LZ complexity either on raw sequences or on sequences encoded by one of the six schemes detailed in Fig.1. The ability of each method to cluster evolutionarily- and/or structurally-related sequences was then assessed at each of the four hierarchical levels of the SCOP classification: family, superfamily, fold, and class. Results are shown as Receiver Operating Characteristics (ROC) graphs and corresponding Areas Under the Curve (AUC) plots (Fig.2).



1. binary codons

	T	C	A	G	T	C	A	G	T	C	A	G
Met (M)	0	0	1	0	1	0	0	0	0	0	0	1
Cys (C)	1	0	0	0	0	0	0	1	1	1	0	0
Ile (I)	0	0	1	0	1	0	0	0	1	1	1	0
Pro (P)	0	1	0	0	0	1	0	0	1	1	1	1
Arg (R)	0	1	1	0	0	0	0	1	1	1	1	1

[△] 12 bytes

2. binary codons; 3rd position discarded

	T	C	A	G	T	C	A	G
Met (M)	0	0	1	0	1	0	0	0
Cys (C)	1	0	0	0	0	0	0	1
Ile (I)	0	0	1	0	1	0	0	0
Pro (P)	0	1	0	0	0	1	0	0
Arg (R)	0	1	1	0	0	0	0	1

[▲] 8 bytes

3. biochemical groupings; binary sets

	T	S	P	L	C	H	R	A
Met (M)	0	0	0	0	0	1	0	0
Cys (C)	1	1	0	1	0	1	0	0
Ile (I)	0	0	0	0	0	1	0	1
Pro (P)	0	1	0	0	0	0	0	0
Arg (R)	0	0	1	1	1	0	0	0

[+] 8 bytes

4. alphanumeric codons

Met (M)	A	T	G															
Cys (C)	T	G	T	T	G	C												
Ile (I)	A	T	T	A	T	C	A	T	A									
Pro (P)	C	C	T	C	C	C	C	C	A	C	C	G						
Arg (R)	C	G	T	C	G	C	C	G	A	C	G	G	A	G	A	A	G	G

[▽] 3 to 18 bytes

5. alphanumeric codons; compressed

Met (M)	A	T	G					
Cys (C)	T	G	T	C				
Ile (I)	A	T	T	C	A			
Pro (P)	C	C	T	C	A	G		
Arg (R)	C	A	G	T	C	A	G	
Ser (S)	T	A	C	G	T	C	A	G

[▼] 3 to 8 bytes

6. biochemical groupings; single byte

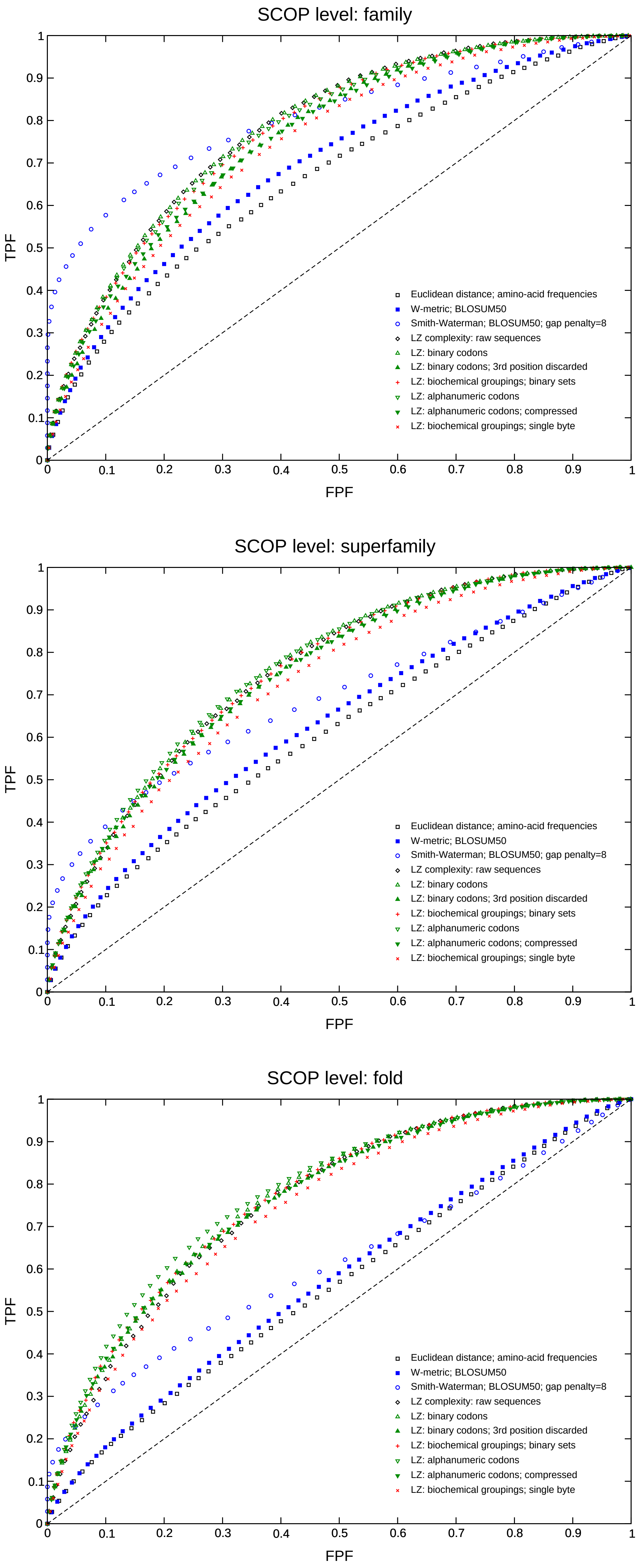
Ala (A) Gly (G) Ile (I) Leu (L) Val (V)	Phe (F) Trp (W) Tyr (Y)	His (H) Lys (K) Arg (R)	Asp (D) Glu (E)	Asn (N) Gln (Q)	Ser (S) Thr (T)	Cys (C) Met (M)	Pro (P)
aLiphatic	*aRomatic*	*Basic*	*Acidic*	*aMide*	*alCohol*	*Sulfur*	*Proline*
L	R	B	A	M	C	S	P

[✕] 1 byte

Figure 1: ***Illustration of the six amino-acid encodings.*** *All schemes are rather self-explanatory except maybe the fifth one, where the different bases found in all codons specifying a given amino-acid are simply enumerated position by position.*

Results

Hierarchical clustering of protein sequences



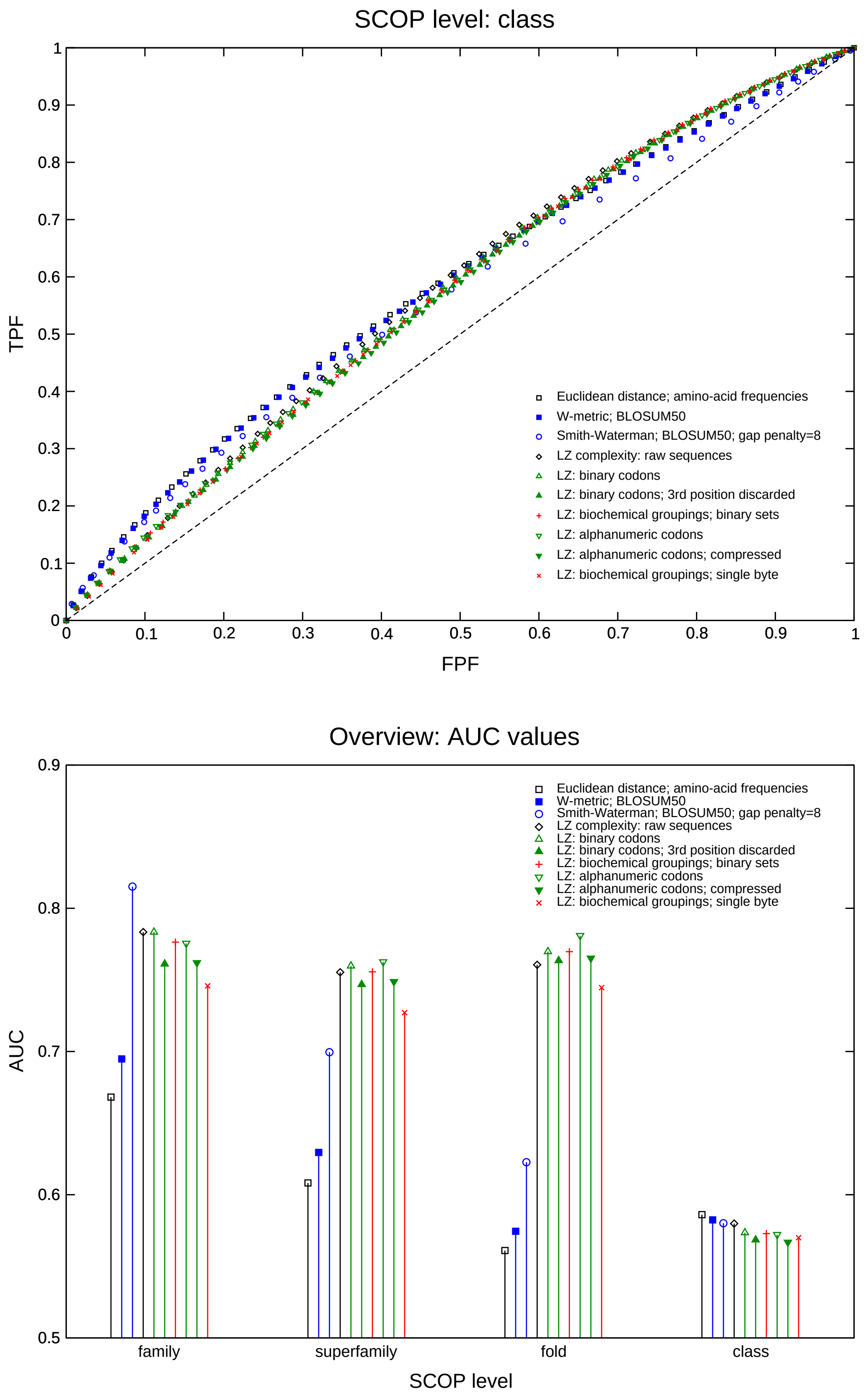


Figure 2: ***ROC curves and AUC values for PDB40-v dataset.*** *True Positive Fraction (TPF) vs False Positive Fraction (FPF). A random classifier would generate equal fractions of TP and FP clustering, which corresponds to the ROC diagonal (dashed line). Accordingly, the better classification schemes have plots with higher values of TPF for equal values of FPF, resulting in higher AUCs.*

Performance considerations

A summary of the CPU-time required by each method is given in the table below. One relative unit corresponds to 7 min 45 on a PowerPC G4 running at 1.25 GHz (Mac OS X) and to 6 min 10 on a Pentium 4 running at 2.4 GHz (SuSE Linux). Our program does a pretty good job in avoiding redundant computations but could be further optimized by packing sequence data in binary format and by making use of the SIMD engines of the CPUs. The perl/C implementation is actually a perl wrapper for the `water` program (written in C) of the EMBOSS package.

	Eu	Wm	SW	LZ	LZ1	LZ2	LZ3	LZ4	LZ5	LZ6
CPU-time	0.2	8	264	1	75	37	37	48	16	1
language	perl		perl/C	C						

Real-world protein phylogeny

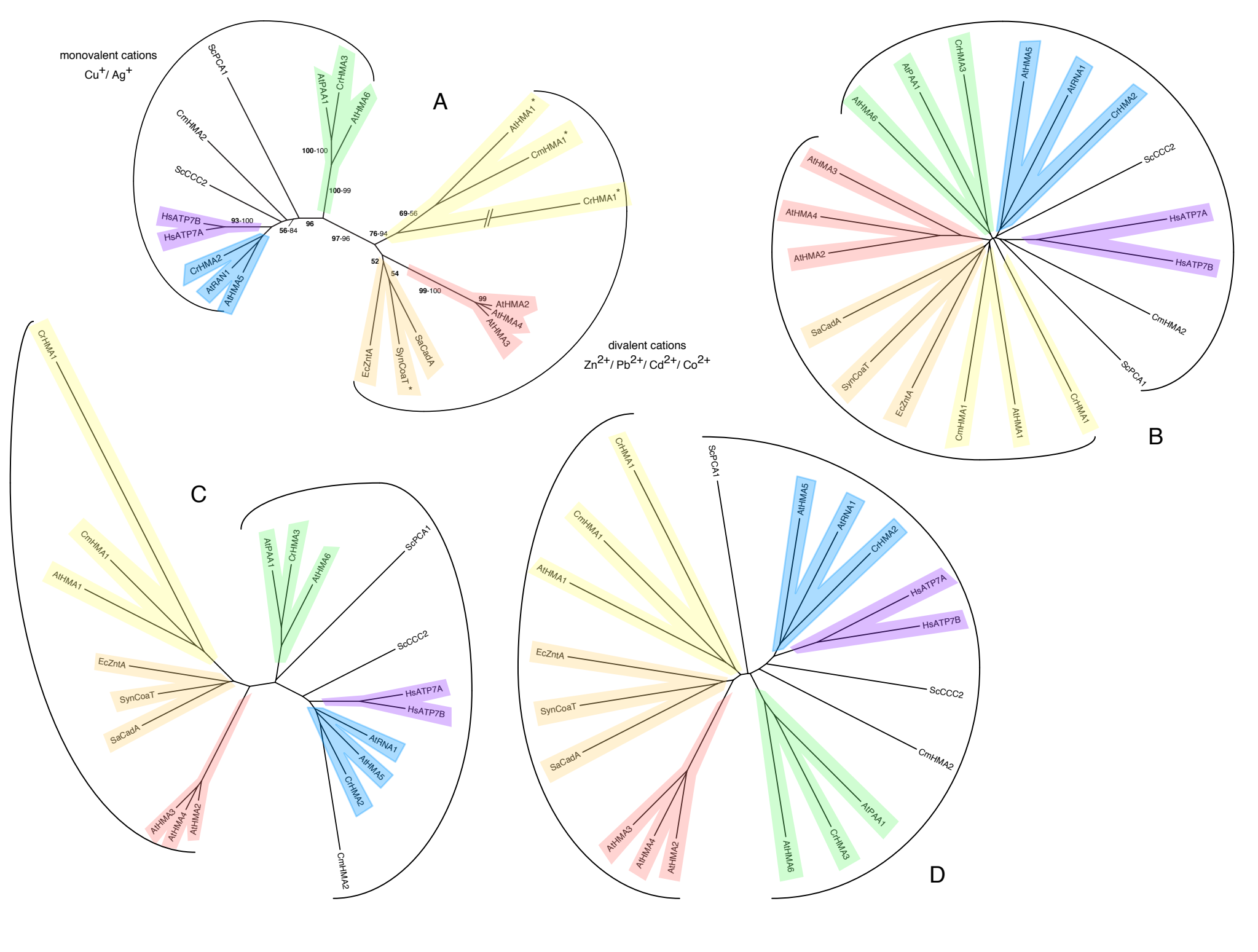


Figure 3: ***Comparison of four unrooted trees of the HMA subfamily of P-type ATPases.*** *Clades from the reference tree are consistently color-coded in all four trees.* ***A****. Reference tree (combination of MP/ML/NJ) from Hanikenne et al. (2005);* ***B****. LZ tree (binary-coded biochemical sets; see Fig.1.3);* ***C****. NJ tree (ClustalX, corrected distances);* ***D****. NJ tree (ClustalX, uncorrected distances).*

Conclusions

1. While computationally affordable, the LZ complexity outperforms all other methods at the three higher levels of the SCOP classification, except the very slow SW alignment at the family level. At superfamily and fold levels, our sequence encodings show slightly better results than the default complexity, but at the expense of considerable computational burden.
2. The LZ complexity is able to retrieve most clades found through alignment-based phylogenetic methods but would need some kind of distance correction to be really useful.

Benelux Bioinformatics Conference 2005, VIB/Ghent University, Ghent, 14–15 April 2005

D. Baurain is a Postdoctoral Researcher of the FNRS.