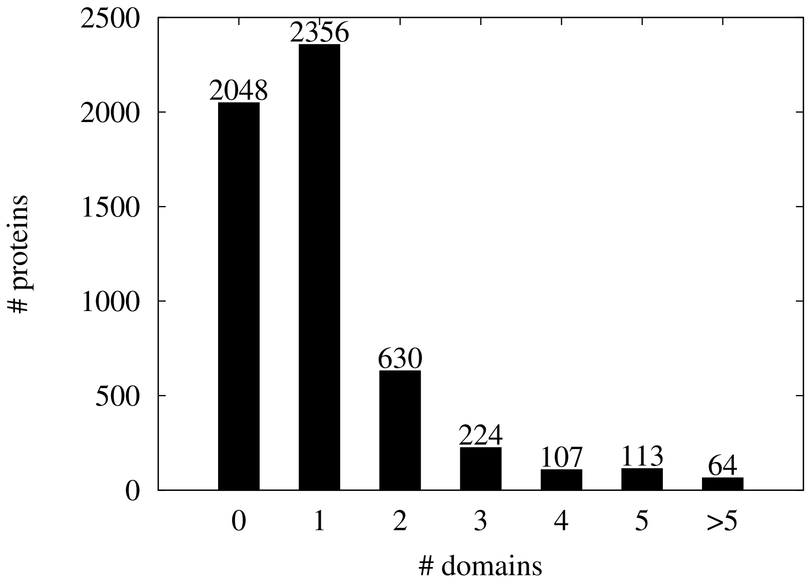
Supplementary File

J.S. Bernardes, F. Vieira, G. Zaverucha and A. Carbone

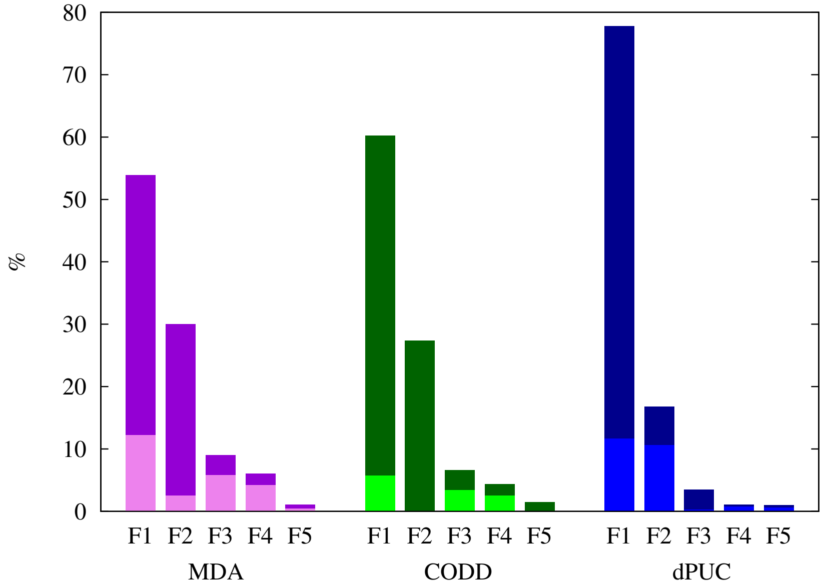
September 13, 2015

Supplementary Figure 1: Distribution of the number of domains present in *P. falciparum* proteins.



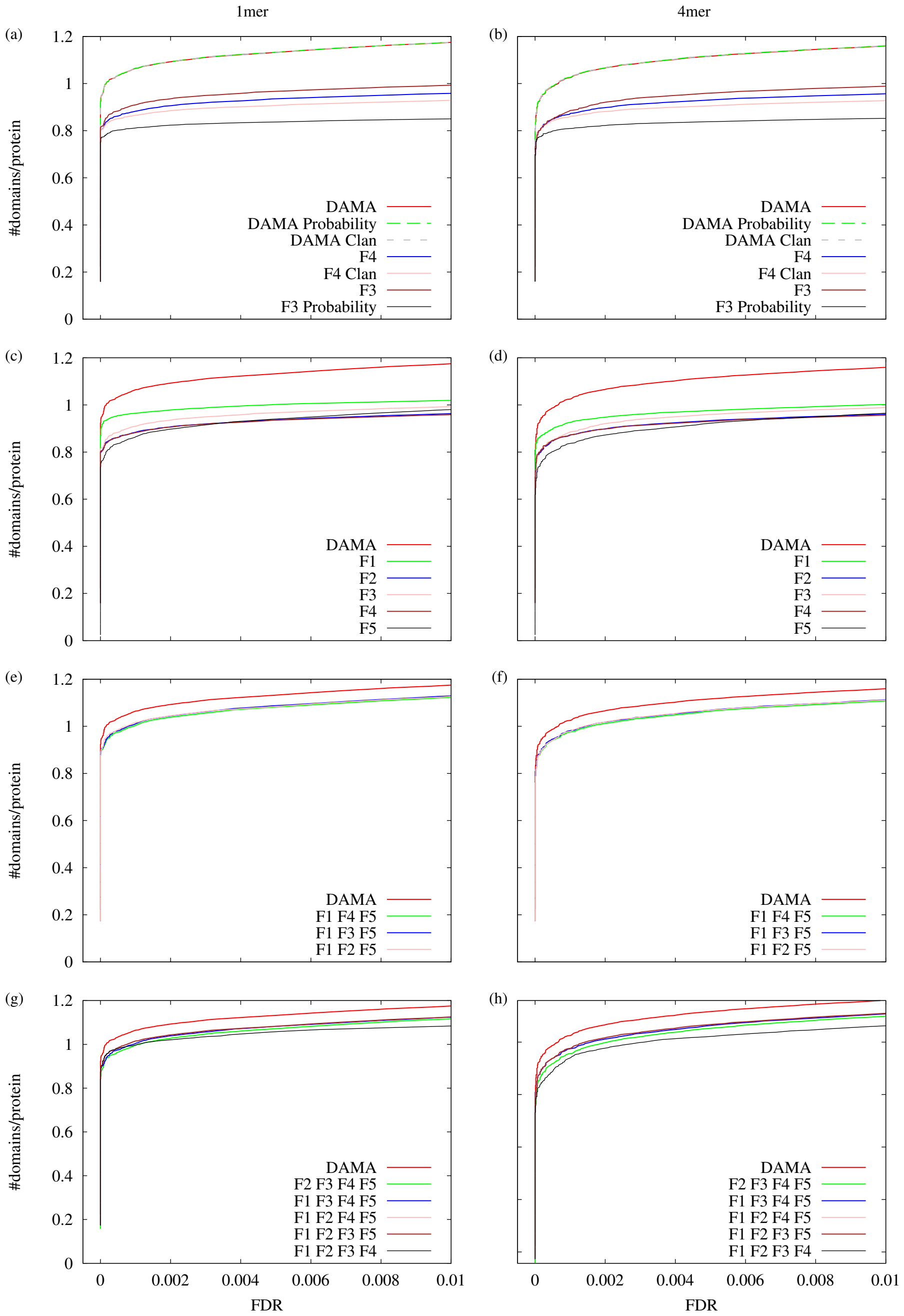
Histogram showing that a large fraction of *P. falciparum* protein sequences is annotated with one domain or none. Only a few proteins contain more than 5 domain occurrences. The annotation was realised with HMMER 3.0 (`hmmscan`), with curated inclusion thresholds (option --cut_ga). The number of sequences with a fixed number of domains is reported on the top of each bin.

Supplementary Figure 2: Role of the five optimisation functions.



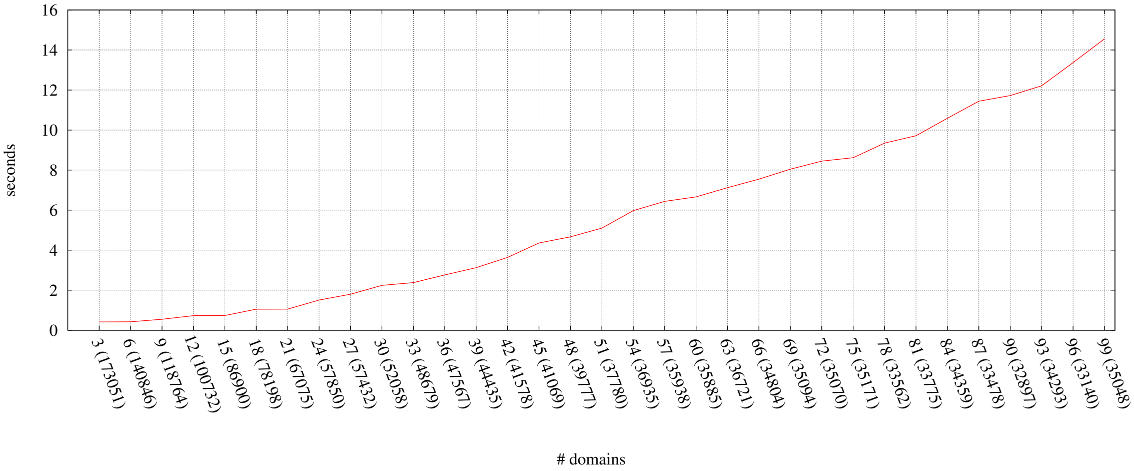
Histogram showing the role of DAMA optimisation functions in filtering out best architectures found by MDA, CODD and dPUC for the *P. falciparum* proteome dataset constituted of 5542 proteins. For each tool, we describe the proportion of its best architectures that was filtered out by a given DAMA optimisation function. Each bin of the histogram is coloured with dark and light tones. Dark colours refer to architectures that are subsets of DAMA best architecture (as in the examples determined in Figure 4), while light colours refer to architectures that contain domains that are not present in DAMA best architecture.

Supplementary Figure 3: Tests on the role of the five optimisation functions.



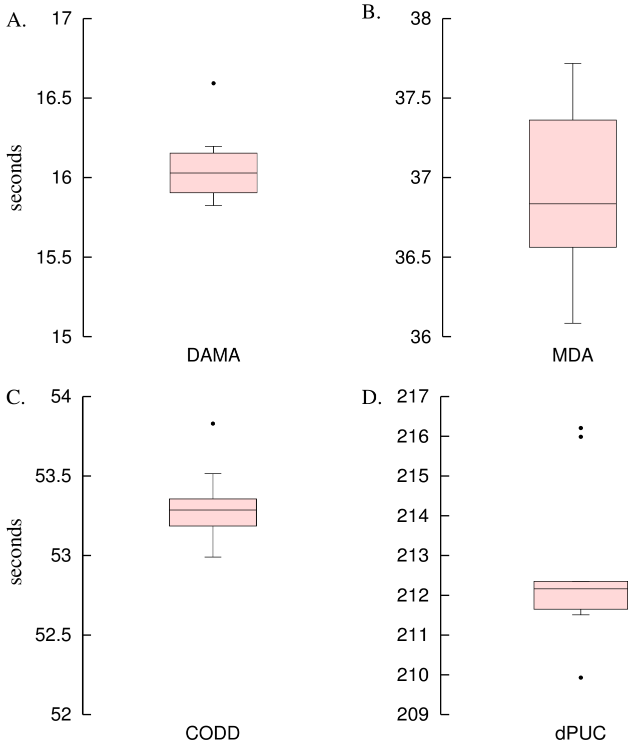
Various tests have been performed on the *P. falciparum* dataset to learn on the role and importance of each optimisation function. Curves report the number of predicted domains per protein obtained at different FDR values. Compare to curves in Figure 3. For all tests, the curve named “DAMA” corresponds to the use of all functions F1-F5 with default parameters; **(a)-(b)**: “DAMA probability”: F3 is defined to be the probability that a distinguished domain pairs in x co-occurs in $\mathcal{L}$, and all other functions remain the same; “DAMA Clan”: F4 is defined with *diff*(x) that returns the number of distinct domains in x that co-occur in $\mathcal{L}$ and do not belong to the same clan. All other functions remain the same; “F4”: domain filtering is realised with F4 only; “F4 Clan”: domain filtering is realised with F4 redefined with *diff*(x) that returns the number of distinct domains in x that co-occur in $\mathcal{L}$ and do not belong to the same clan; “F3”: domain filtering is realised with F3 only; “F3 Probability”: domain filtering is realised with F3 only, where F3 is defined to be the probability that a distinguished domain pairs in x co-occurs in $\mathcal{L}$; **(c)-(d)**: the plots display curves of performance for optimisation functions run alone. **(e)-(f)**: the plots show the influence of F2, F3, F4 taken alone and combined with F1 and F5. **(g)-(h)**: the plots show the influence of F2, F3, F4 taken in pairs and combined with F1 and F5.

Supplementary Figure 4: DAMA time performance on the dataset of generated sequences.



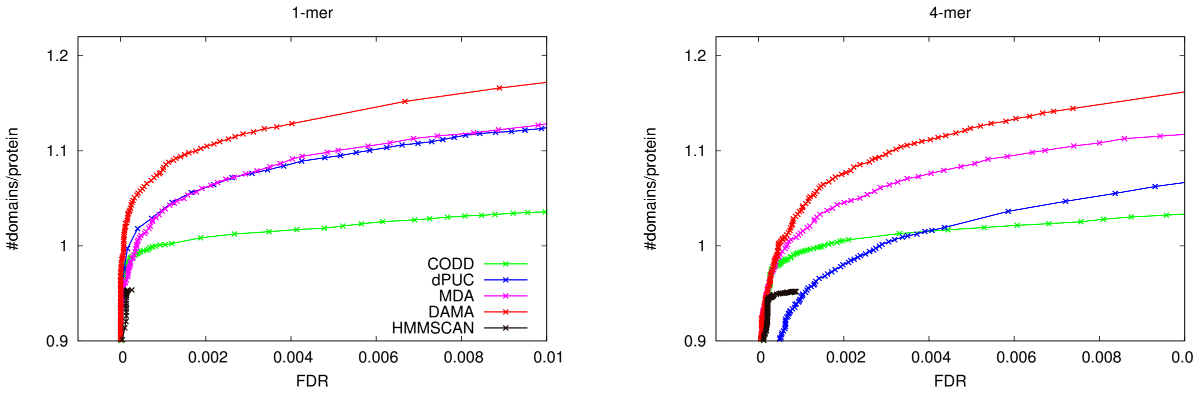
Progressively increasing time complexity is shown on generated protein sequences containing n domains, for n going from 3 to 99. Each point corresponds to the run time obtained on 100 generated proteins. The number of domain hits for each generated set of sequences with n domains decreases with the number of domains, and it is indicated in parenthesis.

Supplementary Figure 5: DAMA time performance on the *P. falciparum* dataset.



Boxplots of the run time of the four methods on the set of *P. falciparum* protein sequences: DAMA (A), MDA (B), CODD (C) and dPUC (D). All experiments were performed on a one core single-user linux (kernel 2.6.32-431.11.2.el6.x86_64 - Red Hat 4.4.7-4) Intel(R) Xeon(R) CPU E5-2650 v2 2.60GHz, with 64GB of RAM.

Supplementary Figure 6: Performance of DAMA and other tools on the *P. falciparum* protein annotation when a common pool of domain hits is selected.



In contrast to Ochoa evaluation method (Ochoa et al., *BMC Bioinformatics*, 2011) used to plot Fig. 3A, here we plot the FDR curves for DAMA, dPUC, MDA and CODD methods by generating all domain hits with HMMER 3.0 at E-value $< 1e\text{-}1$ and by using them as input for the four tools. Each tool reconstructs all architectures at once from the set of these domain hits. For each tool, points in the corresponding FDR curve are obtained by varying the E-value threshold M and filtering out from the architectures all domain hits with E-values $> M$. We varied E-values from $1e\text{-}30$ to $< 1e\text{-}1$ by small steps. From the resulting set of architectures (made of domain hits of E-value $< M$ only), we compute the FDR and the number of domains per protein, and repeat the procedure until the whole curve is drawn. Compare the 1-mer and 4-mer experiments with the ones reported in Figure 3A.

When predictions of protein domain architectures are realised over large sets of proteins, one might be interested to have only one run that accepts domains with a high E-value and then decide how to select architectures out of this run, depending on the characteristics of the output he finds. Also, he might be interested to explore the landscape of architecture predictions before deciding what E-values to use as a threshold. In doing this, one would like to know whether the tool remains robust while competing with a larger number of domain hits/false positives. The evaluation method used here describes the behaviour of the four tools under this condition and it brings information on the tools performances. Several observations can be drawn.

Observe that in Figure 3A, tools are asked to run on several sets of domain hits and that these sets are smaller than the one used here (defined by hits with E-value $< 1e\text{-}1$). This means that in Figure 3A, by considering as inputs smaller sets of domain hits, at each run we construct architectures by handling smaller quantities of negatives (namely, only negatives with E-value $< M$) compared to the evaluation method used here. Recall that domain hits enter in competition against each other during the construction of the architectures and their selection is realised with strategies that are specific to each tools. This difference between the two evaluation methods is fundamental and plays a crucial role in highlighting tools performances.

For example, we observe that dPUC performance is much better evaluated in Figure 3A. In fact, when the number of hits given as input to dPUC is larger, dPUC constructs larger architectures and with a higher number of false positives. This behavior is even more emphasized with the 4-mer experiments where the number of negatives augments and dPUC difficulty in selecting good architectures is highlighted by the divergence of DAMA and dPUC curves.

MDA, CODD and DAMA are less affected in their behaviour because they all handle E-values associated to domain hits explicitly. As a consequence, they deal better than dPUC with negatives. In fact, MDA constructs architectures by optimising first on E-values, and placing first all domains with a good score before adding up to the architecture other domains. CODD follows a strategy similar to MDA since it starts its analysis from architectures suggested by `hmmscan`, having good scores by definition. On its turn, notice that in DAMA architecture reconstruction, the F1 criteria (describing the role of E-values) appears as the most important criteria, and that it precedes the role of co-occurrence. Hence, it is not surprising that dPUC, based on a score formula mixing the context score (evaluating pairwise co-occurrence) with the domain hit score, performs less well than DAMA.

To conclude, in applications where the user can control domain hit E-values and he/she is interested on architectures made of domains with low E-values, then the usage of dPUC is encouraged. On the other hand, MDA, CODD and DAMA display FDR curves that remain robust to the two evaluation procedures since less sensitive to the set of negatives.