

## S2 Text. A probabilistic method for classifying the malaria *var* genes into ups groups (*cUps*)

The *cUps* algorithm can be found at <https://github.com/qianfeng2/cUps>.

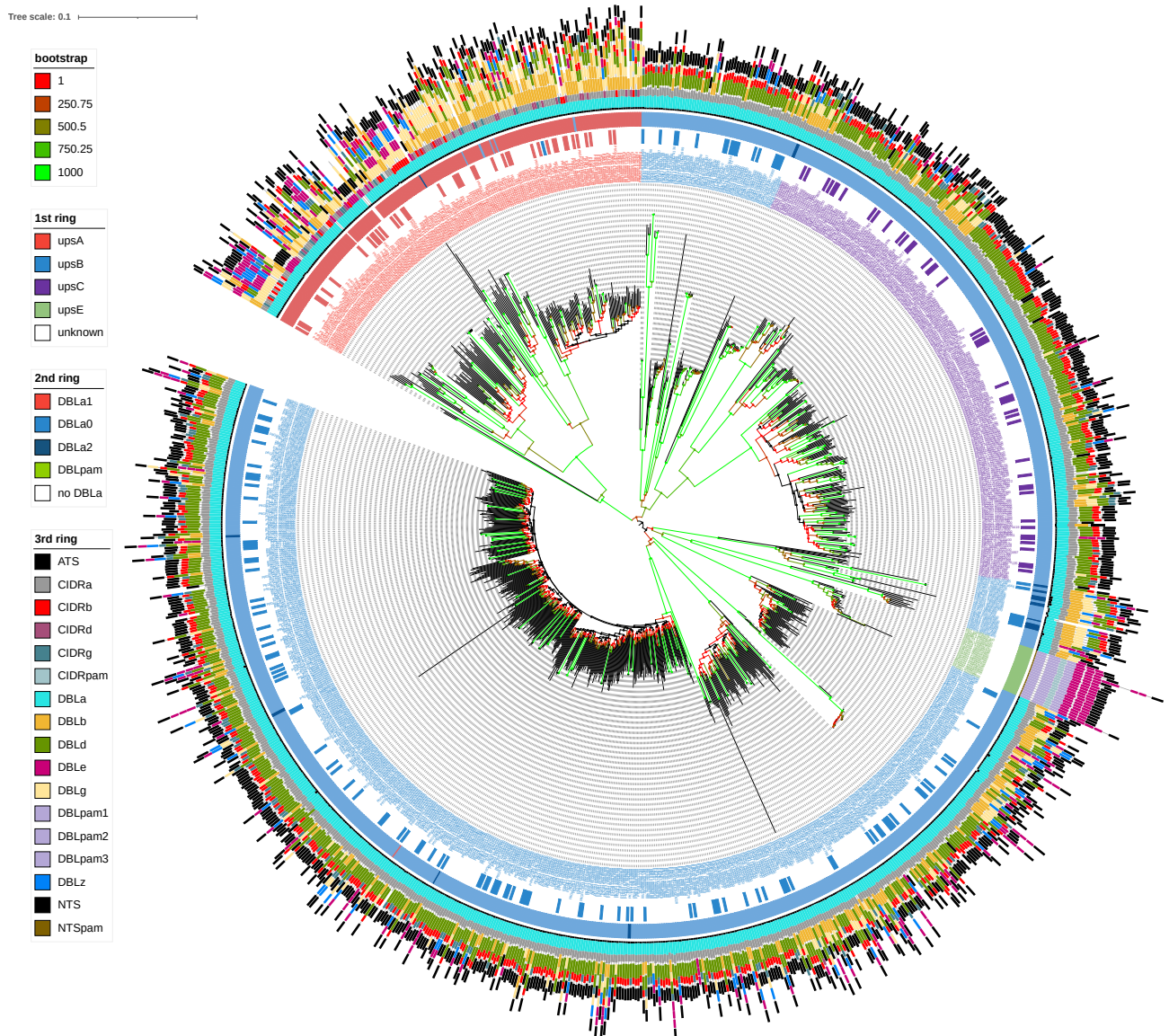
This method is described in much more detail, with verification in a thesis (1).

### S2.1 Creating the reference dataset

Based on methods described by (2), *var* gene sequences were extracted from publicly-available *P. falciparum* genomes and, using their 2kb upstream sequences, assigned to ups groups by Neighbor joining (NJ) and Markov clustering (MCL). Assigned to ups groups, the corresponding DBL $\alpha$  tag sequences were used as reference dataset for the *cUps* algorithm.

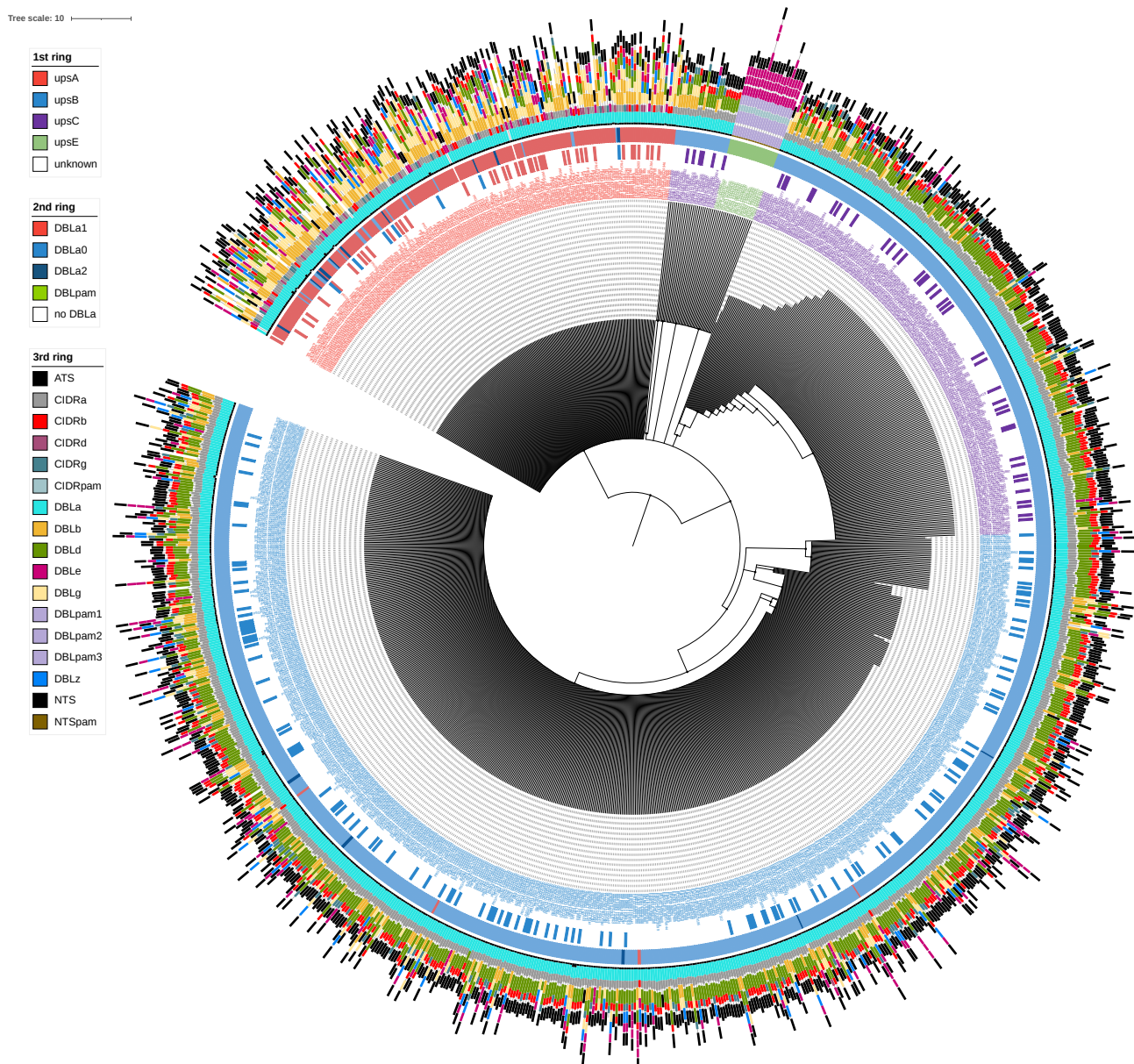
To elaborate, genome assembly and annotation files of a total of 18 *Plasmodium falciparum* strains were downloaded from PlasmoDB release 51 (16 strains: 3D7, 7G8, CD01, Dd2, GA01, GB4, GN01, HB3, IT, KE01, KH01, KH02, ML01, SD01, SN01, and TG01) and NCBI (2 strains: IGH-CR14, RAJ116; accessions: GCA\_000186055.2, GCA\_000186025.2), described in (2,3) to obtain *var* gene sequences and their corresponding 2kb upstream sequences. These strains were selected because they were either long-read assemblies with characterised subtelomeric regions (3) and/or because their *var* genes have been previously described (2). For each genome, GFF-format annotation files were searched for genes assigned with as “PfEMP1”, excluding sequences of pseudogenes, exon 2, or acidic terminal segment (ATS) regions (i.e., those containing keywords such as “pseudogene”, “exon 2”, or “acidic” in their descriptions). Nucleotide and amino acid sequences for the remaining PfEMP1 genes were extracted from coding sequence and protein files that were downloaded from the respective sources. The VarDom 1.0 server (2) was used to predict PfEMP1 domains. To properly identify 2kb upstream sequences, only genes with a NTS domain predicted in the VarDom annotation were included as this domain typically marks the N-terminus of PfEMP1 proteins (2). Bedtools (4) was used to extract genomic sequences 2000 bases upstream from the start of each *var* gene. Using the *hmmscan* in HMMER v3.3.1 (5), DBL $\alpha$  tag sequences were identified based on homology to positions 189 to 430 of the Pfam domain (PF05424). Using *classifyDBLa* v1.0 (6), DBL $\alpha$  tag sequences were classified into different DBL $\alpha$ /pam domain classes (i.e. DBL $\alpha$ 0, DBL $\alpha$ 1, DBL $\alpha$ 2, or DBLpam). Detailed information on the method of extracting DBL $\alpha$  tags and classification into upsA/non-upsA groups is described in (7) and (6), respectively.

Similar to described by (2), multiple sequence alignment of the 2kb upstream sequences was performed with MAFFT v7.471 (8) using the L-INS-i algorithm. ClustalW v2.1 (9) was used to generate a bootstrapped (n=1000) neighbor-joining tree. Upstream sequences were also clustered using the Markov clustering algorithm, following the parameters implemented in (2). This included firstly an all-vs-all pairwise alignment of upstream sequences was performed with *blastn* v2.10.1 (10), followed by clustering with the MCL v14-137 algorithm (11). The same inflation parameters (varied in steps of 0.2 from 1.2 to 5.0) and resource scheme 7 (most accurate) was used, as reported in (2). A consensus of all clustering trees was generated by Majority Rule (with extensions) using Phylip v3.697 (12). Both the NJ and MCL trees were visualised with iTOL v6 (13), alongside assignment of ups groups by (2) where available, DBL $\alpha$ /pam domain classes, and PfEMP1 domain structure (Fig A and B in this document). As ML01 and TG01 were identified as mixed infections in (3), these isolates were excluded from the final reference dataset. Based on results from the NJ and MCL methods, *var* genes with discordant assignments were also excluded from the reference dataset. This resulted in a final consensus ups classification of 846 *var* genes from 16 *P. falciparum* genomes, from which the DBL $\alpha$  tag was included in the reference dataset.



**Fig A. Classification of *var* genes into ups groups by Neighbour-joining, using 2kb upstream sequences.** Tips (leaves) of the tree represent each *var* gene ID, coloured by the ups classification based on clustering shown in this tree. The inner-most (first) ring shows classification of *var* genes into upsA/B/C/E groups by (2) where available. The second ring indicates DBL $\alpha$ /pam domain classes, classified using *classifyDBLa* (6). The outer-most ring shows the PfEMP1 domain composition and structure.





**Fig B. Classification of *var* genes into ups groups by Markov clustering algorithm, using 2kb upstream sequences.** Tips (leaves) of the tree represent each *var* gene ID, coloured by the ups classification based on clustering shown in this tree. The inner-most (first) ring shows classification of *var* genes into upsA/B/C/E groups by (2) where available. The second ring indicates DBLa/pam domain classes, classified using *classifyDBLa* (6). The outer-most ring shows the PfEMP1 domain composition and structure.

## S2.2 Accuracy of ups classification algorithm

### S2.2.1 Performance on reference database

We evaluated the accuracy of our novel ups classification algorithm *via* leave-one-out cross-validation on the reference database, consisting of 846 sequences (148 upsA, 502 upsB, 196 upsC; see Section Data S2.1 above for further details).

In Table A below, we show the confusion matrix for our new method, and in Table B below we assess the accuracy of our method using two standard measures: sensitivity (the proportion of each group that is correctly predicted) and precision (the proportion of predictions for each group that is correct). It is clear that upsA can be classified very accurately; in contrast, although the method retains good predictive power for upsB and upsC, the accuracy is considerably lower. This is consistent with existing methods, which are unable to distinguish B from C at all.

**Table A. Confusion matrix for ups classification algorithm using leave-one-out cross-validation on the reference database.**

		Real			Total
		upsA	upsB	upsC	
Predicted	upsA	145	1	1	147
	upsB	3	350	54	407
	upsC	0	151	141	292
Total		148	502	196	846

**Table B. Accuracy measures for ups classification algorithm using leave-one-out cross-validation on the reference database.**

	upsA	upsB	upsC
Sensitivity	98.0%	69.7%	71.9%
Precision	98.6%	86.0%	48.3%

### S2.2.2 Comparison to existing pipeline

We compared our method to the existing pipeline (2,6). This pipeline first determines the DBL $\alpha$  subclass of the query tag sequence, and classifies DBL $\alpha$ 1 sequences to the upsA group and other sequences to the upsB/C group. This method does not distinguish between the upsB and C groups (we know of no method in the literature that does).

Similarly to our method, the existing pipeline can very accurately classify a sequence to upsA or non-upsA when using leave-one-out cross-validation on the reference database (4 upsA and 1 ups B/C misclassified).

The 3 upsA sequences that are misclassified with our method (identifiers "PfGB4\_000018400", "PfGN01\_100005800", "PfHB3\_120045300") are distinct from the 4 upsA sequences that are misclassified with the existing pipeline ("PfIGH-CR14\_KNG75243", "PfKH02\_030030250",

"PfKH01\_130076800", "PfGA01\_010005600"). In contrast, both methods misclassify the same upsB sequence ("PfGN01\_020028200") to upsA, suggesting that the labelling of this sequence in the reference database may be unreliable. Our method additionally misclassifies an upsC sequence ("PfIGH-CR14\_KNG77255") to upsA.

In addition to the ability to distinguish the upsB and upsC groups, our method also provides meaningful classification (posterior) probabilities that can be exploited for further information. This enables a more detailed comparison of the accuracy of the two methods by considering the output as the classification probabilities (taken as a vector of length 3). For the existing pipeline, we consider a non-upsA classification to have a probability of 0.5 for both upsB and upsC, while an upsA classification has a probability of 1 for upsA as expected.

We compared the methods using three measures: overall accuracy (OA) and kappa coefficient (for definitions see (14)), and root mean square error (RMSE). We show the results in Table C below. Here, we see that our method has significantly better OA and kappa, and similar RMSE to the existing pipeline.

**Table C. Accuracy measures based on classification probabilities for ups classification algorithm using leave-one-out cross-validation on the reference database.**

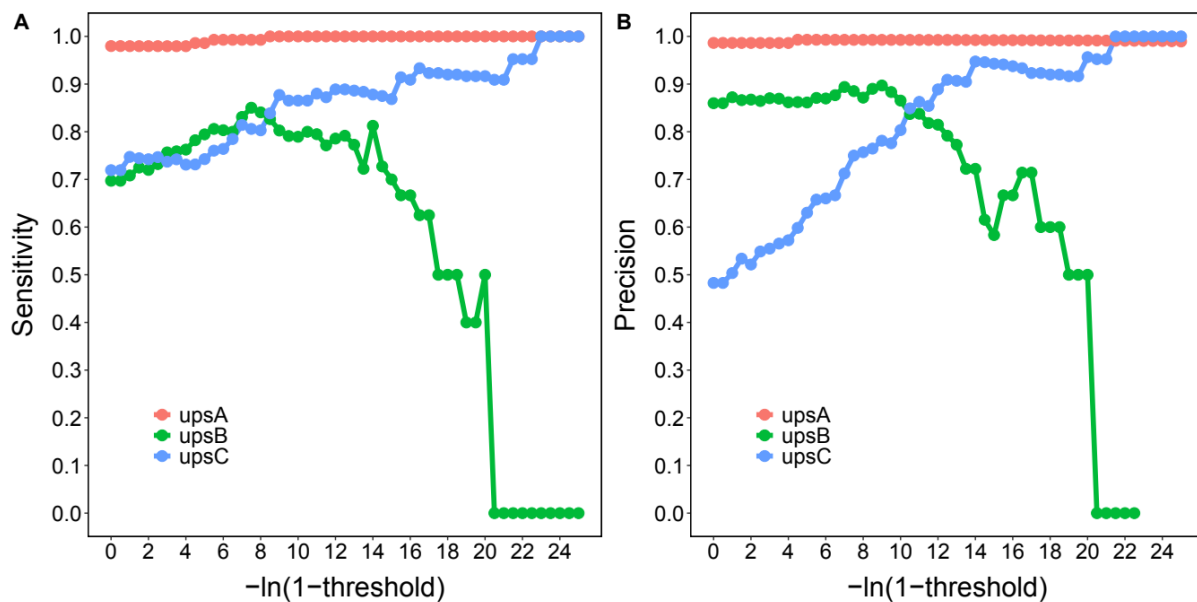
	OA	kappa	RMSE
Existing pipeline	0.582	0.335	0.375
Our method	0.743	0.576	0.377

In summary, we see that our method provides good accuracy, with the ability to distinguish between upsB and upsC groups, and provides further information with classification probabilities.

### ***S2.2.3 Thresholding probabilities to improve accuracy***

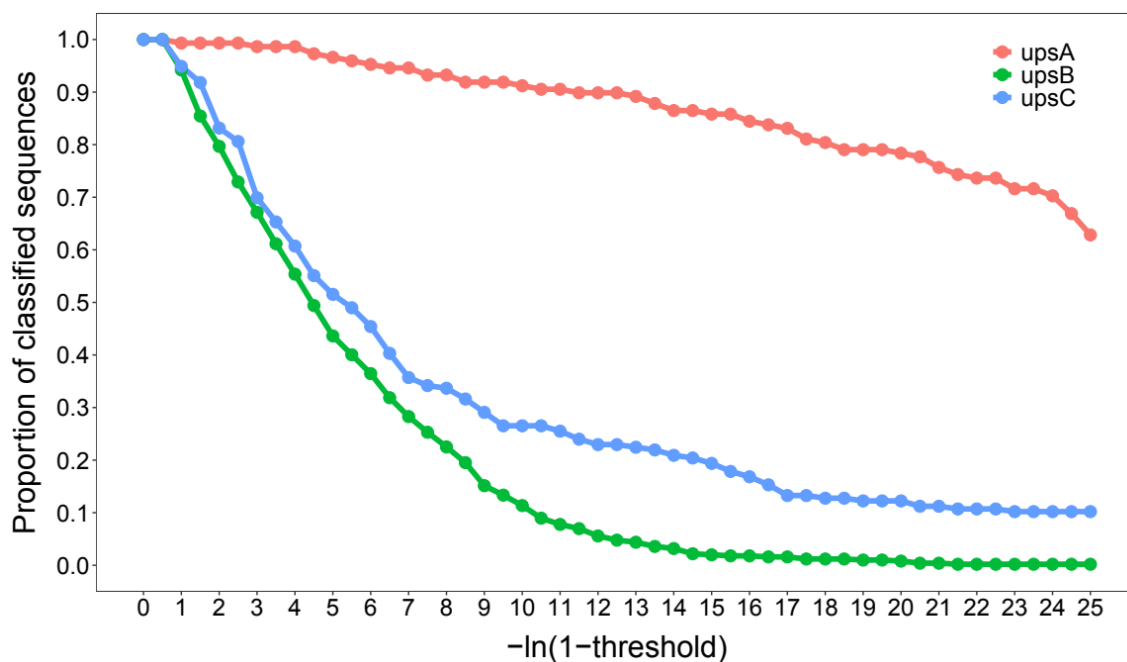
We further explored the possibility of applying a threshold to the classification probabilities from our method, so that sequences with the highest classification probability below the threshold would be considered unclassified.

In Fig C below, we consider the sensitivity and precision for the three groups for a range of thresholds on a logarithmic scale from 1 (the maximum possible threshold). We see that the accuracy of upsA classification is always high, as expected; meanwhile, both upsB and upsC accuracy improves with a higher threshold up to a point, after which upsB accuracy declines. This suggests that we can get better accuracy by applying a threshold, with the optimal threshold around  $1 - \exp(-8)$ . At this threshold, we show a significant improvement on all three accuracy measures used in the previous section (OA 0.899, kappa 0.843, RMSE 0.259).



**Fig C. Sensitivity and precision of the ups classification algorithm for varying thresholds on the maximum classification probability.**

The drawback is that the number of classifications declines steeply with increasing threshold, as shown in Fig D below. At the threshold of  $1-\exp(-8)$ , we are only able to classify 22.5% of upsB and 33.7% of upsC sequences.



**Fig D. Proportion of classified sequences for varying thresholds on the maximum classification probability.**

We conclude that thresholding is likely to be useful in certain cases, such as when studying a small number of particular types for which we want accurate classifications. However, we do not recommend its use when studying broad patterns in the data (as we do in this paper).

## REFERENCES

1. Feng Q. Analysing malaria DBL $\alpha$  sequences with Hidden Markov models. University of Melbourne; 2024.
2. Rask TS, Hansen DA, Theander TG, Gorm Pedersen A, Lavstsen T. *Plasmodium falciparum* Erythrocyte Membrane Protein 1 Diversity in Seven Genomes – Divide and Conquer. PLoS Comput Biol. 2010 Sep 16;6(9):e1000933.
3. Otto TD, Böhme U, Sanders MJ, Reid AJ, Bruske EI, Duffy CW, et al. Long read assemblies of geographically dispersed *Plasmodium falciparum* isolates reveal highly structured subtelomeres [version 1; peer review: 3 approved]. Wellcome Open Res. 2018;3(52).
4. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010 Mar 15;26(6):841–2.
5. Eddy SR. Accelerated Profile HMM Searches. PLoS Comput Biol. 2011;7(10):e1002195.
6. Ruybal-Pesántez S, Tiedje KE, Tonkin-Hill G, Rask TS, Kamya MR, Greenhouse B, et al. Population genomics of virulence genes of *Plasmodium falciparum* in clinical isolates from Uganda. Sci Rep. 2017;7(1):11810.
7. Tan MH, Shim H, Chan Y ban, Day KP. Unravelling *var* complexity: Relationship between DBL $\alpha$  types and *var* genes in *Plasmodium falciparum*. Vol. 1, Frontiers in Parasitology. 2023.
8. Katoh K, Standley DM. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. Molecular Biology and Evolution. 2013 Apr 1;30(4):772–80.
9. Thompson JD, Gibson TobyJ, Higgins DG. Multiple Sequence Alignment Using ClustalW and ClustalX. Current Protocols in Bioinformatics. 2003 Jan 1;00(1):2.3.1-2.3.22.
10. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinformatics. 2009;10(1):421.
11. Van Dongen S. Graph Clustering Via a Discrete Uncoupling Process. SIAM Journal on Matrix Analysis and Applications. 2008 Jan 1;30(1):121–41.
12. Felsenstein J. PHYLIP (phylogeny inference package), version 3.5 c. 1993.
13. Letunic I, Bork P. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic Acids Research. 2021 Jul 2;49(W1):W293–6.
14. Chen J, Zhu X, Imura H, Chen X. Consistency of accuracy assessment indices for soft classification: Simulation analysis. ISPRS Journal of Photogrammetry and Remote Sensing. 2010;65(2):156–64.