Papers Summary

Qian Feng

March 18, 2018

Contents

1 Paper 1: A survey of combinatorial methods for phylogenetic networks

A survey of combinatorial methods for phylogenetic networks, Huson and Scornavacca, Genome Biol. Evol., 2011. The full pdf version please see here.

Phylogenetic trees are not suitable to describe evolutionary history when datasets involve significantly plenty of reticulate events, including horizontal gene transfer, hybridization, recombination, reassortment etc. Phylogenetic network provides an alternative. It is any graph used to represent evolutionary relationships between a set of taxa that label some of its nodes.[2]

This paper is a literature review, introducing briefly fundamental concepts about phylogenetic network and summarizing separate algorithms correspond to each type of network. Figure 1 vividly show all different types of phylogenetic network introduced in this essay.

In theory, Phylogenetic network consists of two types: unrooted phylogenetic network and rooted phylogenetic network. The former one is much more widely used than the latter one in practice, because there are many problems needed to solve in rooted phylogenetic network. First, many algorithms are not designed as a tool in real studies though they have proof-of-concept implementations; second, algorithms have impractical running times. Therefore, developing suitable methods for rooted phylogenetic network is still a unforeseeable challenge.

From another point of view, phylogenetic networks are used in two ways, the first one is as a tool for visualizing incompatible clusters/taxa, we call it “abstract”, “implicit” or “data-displayed” networks; another one represents the evolutionary history including reticulate events, called “explicit” or “evolutionary ” networks. In some sense, most unrooted phylogenetic networks are “abstact”, however, rooted phylogenetic networks could be either abstract or explicit.

★ Pay attention to the difference among these words: Hybridization, Recombination, Mutation, Crossover, Duplication. I could not distinguish them actually. Answers given by Yao-ban:Hybridization is about the species level, Recombination is about gene level, Crossover is double recombination. Mutation is change of specific gene.Duplication could repeat, like could make the gene longer, likewise, gene loss.

2 Paper 2: Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data

Li, Na, and Matthew Stephens. "Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data." Genetics 165.4 (2003): 2213-2233. The full pdf version please see here.

The patterns of Linkage Disequilibrium(LD) are the result of genetic factors and demographic history in a population[3]. Particularly, recombination plays a key role relating with the patterns of LD. For example, when a recombination occurs between two loci, it will reduce the dependence between two alleles, then reduce LD. In this article, authors propose a new statistical method to estimate the underlying recombination rate to get a better review of pattern of LD. It is also meaningful in understanding and interpreting patterns of LD and LD mapping.

Our model based on

$$P(h_1, ...h_n|\rho) = P(h_1|\rho)P(h_2|h_1;\rho), ...P(h_n|h_1, ...h_{n-1};\rho) \qquad (1)$$

where $h_1, ...h_n$ denote n sampled haplotypes, ρ is the recombination parameter. In this new proposed method, we substitute an **approximation** (noted as $\widehat{\pi}$) for those conditional distributions in right term of (1), namely

$$P(h_1, ...h_n|\rho) \approx \widehat{\pi}(h_1|\rho)\widehat{\pi}(h_2|h_1;\rho), ...\widehat{\pi}(h_n|h_1, ...h_{n-1};\rho) \qquad (2)$$

The right term above is what we called likelihood L_{PAC}, through maximizing this likelihood, we could get the recombination parameter ρ.

How to compute $\widehat{\pi}$ is the key point in this article. In fact, according to the appendix A, the core is the utilize of forward algorithm in HMM.

Let X_j denote which haplotype / h_{k+1} copies at site j. X_j is a Markov model on $\{1, 2, ..., k\}$ with emission probability $P(X_1 = x) = \frac{1}{k}(x \in \{1, 2, ...k\})$, the transition probability is as follows

$$P(X_{j+1} = x\prime|X_j = x) = \begin{cases} p_j + \frac{1}{k}(1 - p_j) & x\prime = x \\ \frac{1}{k}(1 - p_j) & \text{otherwise} \end{cases} \qquad (3)$$

where $p_j = exp(-\frac{\rho_j d_j}{k})$, $rho_j = 4Nc_j$, N is the effective population size ,and c_j is the recombination rate per physical distance, d_j is the distance between marker j and marker $j + 1$.

Computing $\widehat{\pi}(h_{k+1}|h_1,...h_k;\rho)$ requires a sum over of all possible values of X_j, this is why we exploit forward algorithm to compute it recursively.

$$\pi(h_{k+1}|h_1,h_2,...h_k)=\sum_{x=1}^{k}\alpha_s(x) \tag{4}$$

$$\alpha_{j+1}(x)=e_{j+1}(x)\sum_{x\prime=1}^{k}\alpha_j(x\prime)P(X_{j+1}=x|X_j=x\prime) \tag{5}$$

A noticeable point in this article is that c_j has different set in different models. When the recombination is constant, ie.$c_j=\bar{c}$, we only have one parameter $\bar{c}$, in this case, we use the *golden bisection search* when maximizing the likelihood. When the recombination parameter is variable, *ad hoc* two-stage strategy is used to estimate c_j and λ_j, but we need to know the *ad hoc* two-stage approach is not guaranteed to get reliably global maximum.

★ What on earth is the *ad hoc* two-stage approach? Since authors do not pursue here, why they do not use MCMC mentioned before?

Answers given by Heejung: It is about Baysian theory. Through maximizing product of likelihood and prior distribution to get the estimation.

$$P(\mu|X_1,X_2...X_n)=\frac{P(X_1,X_2...X_n|\mu)P(\mu)}{P(X_1,X_2...X_n)} \tag{6}$$

In summary, this algorithm has several advantages and a small disadvantage. It is computationally fast, able to avoid the assumption that LD has the "block-like" structure, also able to consider all loci simultaneous rather than pairwise. However, an unwelcoming feature of this method is it does not consider the order of haplotypes which other currently available algorithms take account. Fortunately, by averaging the L_{PAC} over random orders of the haplotypes, authors find related performance is not significantly sensitive to the orders used.

3 Paper 3: Phylogenetic networks: a review of methods to display evolutionary history

Phylogenetic networks: a review of methods to display evolutionary history, David A.Morrison, Annual Research and Review in Biology, 2014

The full pdf version please see here.

Evolution involves a series of unobservable historical events, we could neither make direct observation nor perform experiment to investigate them, making phylogenetic an interesting and challenging discipline.

Sources of evolutionary novelty include vertical evolutionary processes and horizontal evolutionary processes. Phylogenetic trees are intended to solely for vertical processes, however, phylogenetic networks is more general with accommodating horizontal events. These horizontal evolutionary processes are represented by reticulation in networks. In this review paper, the author focus on the rooted phylogenetic network, even though there are a few automated methods available for constructing them. Most empirical networks are constructed either manually or by modifying the output of computer program.

Pay attention to these words:*diploid, polyploid, homoploid*

Horizontal evolutionary processes contain hybridization, introgression, HGT, recombination, viral reassortment and genome fusion. The last one is considered to be rather rare since it means the addition of whole genome from one specie to another specie.

Hybridization: Hybridization is very common in plants and a small group of animals like fish and reptiles. The new hybrid species consist same amount of genomic materials from each of the two parental species, ie 50:50 composition. Fig 2 shows the difference between homoploid and polyploid. In particular, polyploid is the only form of reticulate evolution we can construct the history by trees. From multi-labelled trees, K.T.Huber has constructed available and implementable method to construct phylogenetic network that is guaranteed to have a minimal number of interaction nodes. The core of this algorithm is merge and prune maximal inextendible subtrees and its equivalent subtrees, this process is repeated until a network is obtained that contains no repeated labeled leaves. Java package PADRE is available to visualize the network.

Introgression is not 50:50 composition because hybrid individuals backcross preferentially to one of the parental species.

Introgression and HGT are both the transfer of genetic materials from one specie to another, but the former one occur via sexual reproduction and latter one does not. Fig 3 also vividly show the introgression and HGT. HGT is detected by incompatibility between two or more trees for the same site of species. HGT is common among bacterias.

Reassortment means when two strains co-infect a host cell, then create a new strain by re-combing these two genetic materials.

Recombination concludes intra-genic Recombination and inter-genic Recombination. The former one represents the break-points occur within a single gene, however, the latter one can occur in different genes or non-coding space between genes. A noticeable point is crossover means double recombination. In general , genes with low level of recombination will have low levels of polymorphism, hence, recombination has important influence on genome and genetic structure in population.

Homoplasy is the development of organs or other bodily structure within different species, resemble each other and have same functions, but do not have the same ancestral origins. For instance, the wings of insects, birds and bats, are homoplastic (meaning: similar in form and structure, but not in origin).

Apparently, there is a growing need for researchers to detect and display the evolutionary networks.

Lastly, author introduced briefly current usage of different reticulate patterns. Available programs are as following: Dendroscope and SplitsTree for hybridization, SPRIT for HGT, Kwarg and SHRUB for recombination.

Feedbacks:

1. Look at paper in reference 72 since we should know more about recombination.
2. Look at more details of SHRUB software.

4 Paper 4: Reconstructing the evolutionary history of polyploid from multilabeled trees

Reconstructing the evolutionary history of polyploid from multilabeled trees. Huber, Katharina T, et al. Molecular Biology and Evolution 23.9 (2006): 1784-1791.

Polyploid species played a major role in the evolution of plants. In this paper, we focus on present all possible phylogenetic networks from a multilabeled tree that are guaranteed to have minimal number of interaction nodes.

Based on Fig.2 in this paper, we can see (b)(c)(d) are all phylogenetic networks that exhibit (a), however, we will use efficient algorithm to draw a phylogenetic network like (d) rather than (b) and (c).

From Fig.3 in this paper, we can see subtrees $T_{(u)}$,$T_{(v)}$ and $T_{(w)}$ are maximal inextendible. Fours useful concepts are subtree, equivalent, inextendible and maximal inextendible. In fact, we will focus on how to find maximal inextendible subtrees from a given MUL tree in actual algorithm. Note that the definition of *inextendible* is not clear for me, so I borrowed another one in the paper of Huber KT and Moulton to get a better understanding this terminology.

inextendible: suppose T is MUL tree, for every vertex $v \in V(T)$ that is not the root of T we denote the parent of v by $\bar{v}$, suppose T' is a sub MUL tree with vertex v, we say T' is *inextendible* if there existed another sub MUL tree T'' with root vertex w so that T'' is isomorphic to T', and $T(\bar{v})$ is not isomorphic to $T(\bar{w})$.

So there would be a **contradiction** in the statement of inextendible definition. Take a look at Fig.3 again, whether on earth each subtree having leaves labeled with "b" and "c" is inextendible or not ?

The core of this algorithm showed in Fig.4 is to merge and prune maximal inextendible subtrees and its equivalent subtrees, this process is repeated until a network is obtained that contains no repeated labeled leaves.Unfortunately, I could understand how do they find the maximal inextendible subtrees by height list H and code $c(v)$ in the initial step.

A noticeable limitation is when MUL tree contains polytomies showed in Fig.6,using this presented constructing methods would lead to several different phylogenetic networks.

5 Paper 5: Joint Bayesian Estimation of Alignment and Phylogeny

Benjamin D. Redelings, Marc A. Suchard; Joint Bayesian Estimation of Alignment and Phylogeny, Systematic Biology, Volume 54, Issue 3, 1 June 2005, Pages 401-418.
The full pdf version please see here.

In this paper, authors propose a novel model to estimate multiple sequence alignment , phylogenetic tree reconstruction and their support simultaneously. They operate in Bayesian framework and use MCMC methods.

In general, researchers first need to finish multiple sequence alignment, and then use it to reconstruct the phylogeny. However, if the alignment contains ambiguous regions particular like in distantly related sequences, this would lead to inaccurate result. A normal technique is to remove these regions, hence, leading to a loss of a large fraction of informative sites. Due to this, researchers come up with a simple method: split ambiguous columns into groups of residues in which homology is unambiguous, then, place them in separate columns, yet, it is still worth noting that identifying these ambiguous regions is too subjective.

There are a number of techniques are developed to get a better use of ambiguous alignment. One technique, known as elision, concatenates a set of near-optimal alignments into a larger alignment and use them for reconstructing phylogeny. However, elision treat all the near optimal alignment equally instead of weighted, as a consequence, it may be not so good. Another technique, known as optimization alignment, involves estimation of alignment and phylogenies simultaneously with parsimony framework. One problem of optimization alignment is the measure of uncertainty is hard to obtain since standard bootstrap couldn't be applied due to the dependent alignment columns. Method in this paper not only weighted the alignment naturally, but also assess the confidence using posterior probability.

Advantages of joint estimation are as follows: one thing is joint estimation doesn't require extra guide tree, this contrast with alignment with progressive alignment are biased and need a guide tree. Another thing is joint estimation could provide more accurate substitution and indel (inseration/deletion) models, in scoring alignment by extended substitution model, in using shared indels to group taxa on a tree.

In developing indel model, there is a simplified assumption: indel event occurs independent on each branch. The reason of this is it allows us to use a simple pair-HMM to model the alignment. In addition, insertions and deletions are equally likely and sequence lengths do not grow or shrink over time. In substitution model, evolution is independent across columns of f and each branch.

MCMC needs to sample from posterior distribution of the alignment, phylogeny and model parameters given only unaligned sequences. In this paper, researchers introduce a new transition kernel to resample the topology and alignment that improves mixing efficiency, allowing chains to converge even when start with an arbitrary alignment.

Multiple sequence alignment A is actually a matrix f, it specifies which letters from sequences are homologous by arranging homologous letters into the same column in this matrix.

Methods
This is an annotation list:

Item	Name	Meaning
1	Y	a set of n homologous molecular sequences
2	A	multiple sequence alignment
3	τ	unrooted tree topology
4	T	branch length
5	Θ	substitution process parameters
6	Λ	indel process parameters
7	N	total number of nodes in τ:$N = 2n - 2$
8	B	total number of branches in τ:$B = 2n - 3$
9	γ	distribution of ancestral letters at the root node
10	b	every branch
11	$\rho(b)$	parent branch for b
12	$n(b)$	node in τ shared by the branch b and $\rho(b)$

Therefore, the whole state space Ω is composed of points: $\omega = (Y, A, \tau, T, \Theta, \Lambda)$.

5.1 Substitution model

Likelihood $P(Y|A, \tau, T, \Theta, \Lambda)$ is given in substitution model, before getting this result, we need to accomplish two tasks, first one is to specify how A arranges Y into matrix f, next one is to specify the probabilistic model on the columns of f. Figure 1 in the original paper solves above first task, the tuples at the leaf node within a column in matrix f are from a multinomial distribution which addresses second task mentioned before. An worth noting thing is there is *Felsenstein wildcards* in matrix f which represents internal nodes that are present but unobserved denoted by $*$. The full likelihood is by multiplying likelihood of each column using peeling algorithm, meanwhile Felsenstein wildcards and gaps are treated as missing data.

A Markov model is reversible equivalent to hold the detail balance equation:

$$\pi_i p_{ij} = \pi_j p_{ji} \tag{7}$$

5.2 Gap model

We describe prior of indel process parameters Λ in gap model. In gap model, A could be represented as a tuple pairwise alignments: $(A^{(1)}, A^{(2)}, ..., A^{(B)})$, we replace the alignment prior (equation(11)) with standard prior (equation(10)), then after calculating this modified posterior, Gibbs sample from it using DP programs, detailed DP algorithm is provided in Appendix.

Three parameters are used in pair-HMM, including δ,$\epsilon$and ζ,parameter δ refers to the probability of an indel in either sequence, we assume double-exponential distribution on the approximate log odds of δ; ϵ refers to the probability of extending an existing gap, and exponential distribution is assumed, for ζ, it means the transition probability from any state to the end state, in our example, $\zeta = 1/1000$.

5.3 MCMC sampling process

In MCMC sampling process, researchers employ a random scan Metropolis-within-Gibbs approach. In every iteration, they attempt to sample from every model parameters at least once. Indel parameters are resampled more frequently than substitution parameters. NNI is used to update topology τ, it will move across every internal node at least once per iteration.

In whole, topology τ is updated by a number of MH steps, each alters only part of the topology. Once a topology is chosen, the internal nodes are resampled from the DP matrix, after this step, alignment A and topology τ is resampled again. The MH acceptance probability is given in equation (12). This is an 1D DP problem.

Regarding to alignment sampling, traditional two MCMC transition kernels are not efficient enough in some circumstances shown in Figure 4, researchers decide to resample the alignment along a branch and the sequence at one end of this branch in the same step, this is a 2D DP problem.

Alignment uncertainty plot (AU) is introduced to depict the alignment variability, it is drawn from the posterior alignment, and provides a valuable tool to assess alignment ambiguity.

We show this modified MCMC algorithm converges to equilibrium distribution more quickly than previously available MCMC transition kernel. This allows us to choose randomly one alignment rather than start from an estimated alignment via other software like ClustalW, which is the first advantage, another benefit is that it would decrease burn-in time and less autocorrelation. In order to assess the convergence of continuous parameters, Gelman-Rubin R statistics is employed, results suggesting each chain converges to the same distribution, 95% Baysian credible intervals about posterior probability (PP) are also given in table 2.

5.4 BAli-Phy show

BAli-Phy is a C++ software providing samples from posterior of alignment and phylogeny model. I will show it step by step to interested readers. This software please click here.

Firstly install the BAli-Phy using homebrew based on its User's Guide in this software website. We will run commands in terminal. Type this command to check its version: bali-phy –version

Next step is to install programs used for viewing the results
1. Tracer : MCMC parameter, diagnostic viewer.
2. FigTree : Phylogeny Viewer
3. SeaView : Alignment viewer.

Then a quick start to run BAli-Phy is to type in two following commands:

```
bali-phy  /examples/sequences/5S-rRNA/5d.fasta –iter=150
bp-analyze 5d-1/
```

The output results are in a separate file named "5d-1" which is in root directory (/fengqian), including

Item	Name	Meaning
1	C1.log	Numeric parameters: indel and substitution rates, etc. Opened by Tracer
2	C1.trees	Tree samples: one sample per line, in Newick format. Opened by FigTree
3	C1.Pp.fastas	Sampled alignments for partition p including ancestral sequences
4	C1.out	Iteration numbers, probabilities, success probabilities for transition kernels

and other little files.

Last, to summarize the output, type this to find majority consensus tree, note the dir has changed to 5d-1.

```
trees-consensus C1.trees > c50.PP.tree
```

To compute the maximum a posteriori tree, input :

```
trees-consensus –skip=10% C1.trees –map-tree=MAP.tree
```

Checking topology convergence use:

```
trees-bootstrap C1.trees
```

it will show us the PP and LOD values regarding to each topology as well.

Anyway, even though this framework enjoys great advantages in estimating alignment and phylogeny at the same time, there exist some limitations as well. First is the parameters prior are not sufficient for biological meaning. Another shortcoming is indel process parameters are the same along each branch, which contrasts with ClustalW. In ClustalW, there are a function of branch lengths.

It is a challenge for me to understand equation (4) and (6) in this paper fully.

PS: I am learning MCMC from Chib Siddhartha and Edward Greenberg (1995)'s paper, a powerful algorithm, it seems pretty interesting. ★

6 Paper 6: Plasmodium falciparum antigenic variation. Mapping mosaic var gene sequences onto a network of shared, highly polymorphic sequence blocks

Bull, Peter C., et al. "Plasmodium falciparum antigenic variation. Mapping mosaic var gene sequences onto a network of shared, highly polymorphic sequence blocks." Molecular microbiology 68.6 (2008): 1519-1534. The full pdf version please see here.

var gene have a modular organization, consisting of various numbers and combinations of duffy binding like (DBL)domains of different types$(\alpha, \beta, \delta, \epsilon, \gamma and x)$ and cysteine rich interdomain regions(CIDR) of different classes(α, β, γ).The overall architecture of var genes is highly variable in terms of total number of domains and their order. Recombination between non homologous chromosomes is tat least part of reason of it. PFEMP1 plays a central role in malaria transmittion and are immune target(antigen), encoded by an extremely diverse gene family called var. Each genome is made up of 50-60 var genes.

7 Paper 7: Ape parasite origins of human malaria virulence genes

Larremore, D. B. et al.Nat. Commun. 6:8368 doi: 10.1038/ncomms9368 (2015) The full pdf version please see here.

An general biological introduction about *var* genes are as follows:

There are four parasites in total, one of them is *plasmodium*. Regarding plasmodium, they could exist in Chimpanzees, gorillas, bird, reptiles, and even humans. There are four species of plasmodium could infect humans in nature, in addition, one is zoonotic malaria plasmodium. They are:

1. **P.falciparum**: Cause of severe malaria, found in tropical and sub-tropical areas, infected parasites even could clog blood vessels.
2. **P.vivax**: Mostly found in Asia where has high population density.
3. **P.oval**: Mostly found in Africa where most people are negative for the Duffy blood group.
4. **P.malariae**: Three-day cycle, former three kinds are two-day cycle.
5. **P.knowlesi**: Zoonotic malaria in Southeast Asia, particular in Malaysia. 24 hour cycle leading to rapidly severe infection.

Each *plasmodium* genome encodes approximinatly 60 different PfEMP1 proteins, which are expressed from *var* genes, one at a time. Each *var* gene consists of various numbers and combinations of DBLα and CIDR domains. Another way of say from QixinHe, Each *var* gene is composed of l epitopes that connected linearly, and each epitope can be viewed as a multi-allele locus with n alleles.

Meiotic recombination and mitotic recombination is major mechanism of *var* genes.

1. **Meiotic recombination**: Found within mosquitos. Germ cells is produced, sexual stage of parasite.
2. **Mitotic recombination**: Found in asexual blood where the parasites spend most of their life cycle. Somatic cells is produced.

This paper shows us five conclusions as follows:

1. **P.r and P.f shared same modular HVR architecture**: PrCDC is one previous sample from species P.r, previous study has showed the evidence for the presence of HB(Homology Blocks)in P.r and P.f. In contrast, researchers here focus on HVR(Highly variable regions) in

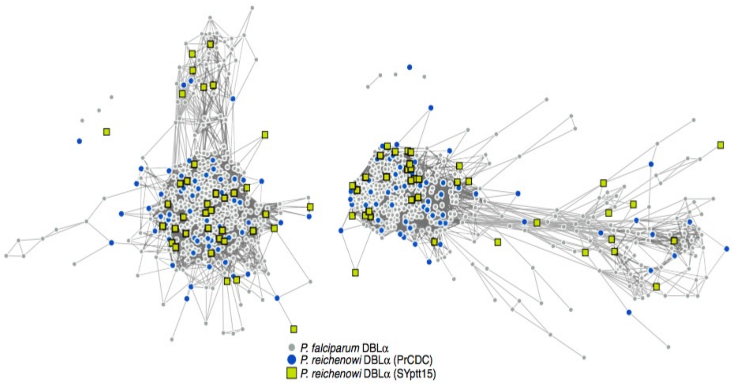

● *P. falciparum* DBLα
● *P. reichenowi* DBLα (PrCDC)
🟩 *P. reichenowi* DBLα (SYptt15)

Figure 2 | Networks of DBLα sequences from *P. reichenowi* and *P. falciparum*. Each node represents a DBLα HVR sequence and each link represents a shared amino-acid substring of significant length[21]. *Laverania* species and strain origin is indicated by node colour and shape. Left and right networks correspond to left and right HVRs, respectively. *P. falciparum* and *P. reichenowi* sequences do not cluster by species or sample in either HVR. Link lengths and node placements are determined by a force-directed layout to better reveal structure, if it exists (see the Methods section). Additional analyses of these networks are shown in Supplementary Fig. 1.

Figure 1: This graphic is adapted from original paper

DBLα domain. After the network construction by "webweb" tool, as shown in Figure 2 and Supplementary Figure 1, it exhibits both this two species share the same modular HVR architecture. It is further established P.r and P.f are sister taxa.

2. **Var DBLα structure are similar among the whole Laverania species**: When we extend and analysis parasite sequences to the whole Laverania subjenus. From Figure 3, we can clearly see the single-infection species (C1(P.r), G1(P.p), C3(P.b), C2(P.g))and P.f indicate the presence of shared mosaic elements after comparing the non-var DBL domains, moreover , every Laverania var tag contains three conserved motifs separating two HVRs, indicating all species in Laverania subjenus share similar DBLα tag structure.

   ★ Add my thought when I read through this part: one is when we look at this Figure 3 carefully, we could find part of P.g species (red square) are far from the other P.f, P.r and P.b, remaining sequences in P.g exist in major HVR lest and right region. Apparently authors do not provide sensible explanation. The other one is if we look at the right HVR network, it is clear to notice the yellow circles are assembled in the major region particularly. According the legend, yellow circles represent non-var DBL circle, it seems not very sensible in some sense, authors even do not mention it in the paper.

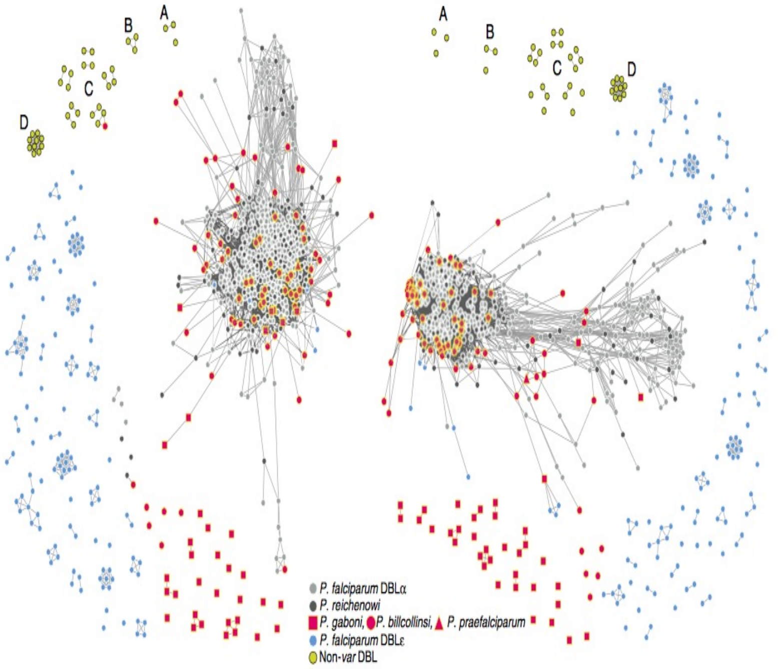


Figure 3 | Networks of DBL sequences from *Laverania* single-species infections in the context of known DBLα and non-DBLα sequences. Each node represents a DBL HVR sequence from a single-species infection and each link represents a shared amino-acid substring of significant length. Note that for each sample, only unique *var* DBL haplotypes were included in the network analysis. Nodes with zero links indicate sequences that share no significant amino-acid substrings with other sequences. Networks were built separately for each HVR, where mosaic diversity is highest (see the Methods section). Colours correspond to *Laverania* species as indicated; annotated yellow nodes correspond to (A) *dblsmsp1* and (B) *dblmsp2* from Pf3D7, PfIT and PrCDC; (C) both DBL domains from *ebl1*, *eba140*, *eba165*, *eba175* and *eba181* of Pf3D7 and PfIT; (D) *P. vivax* Duffy-binding proteins; see Supplementary Table 3 for a comprehensive list of non-DBLα sequences.

Figure 2: This graphic is adapted from original paper

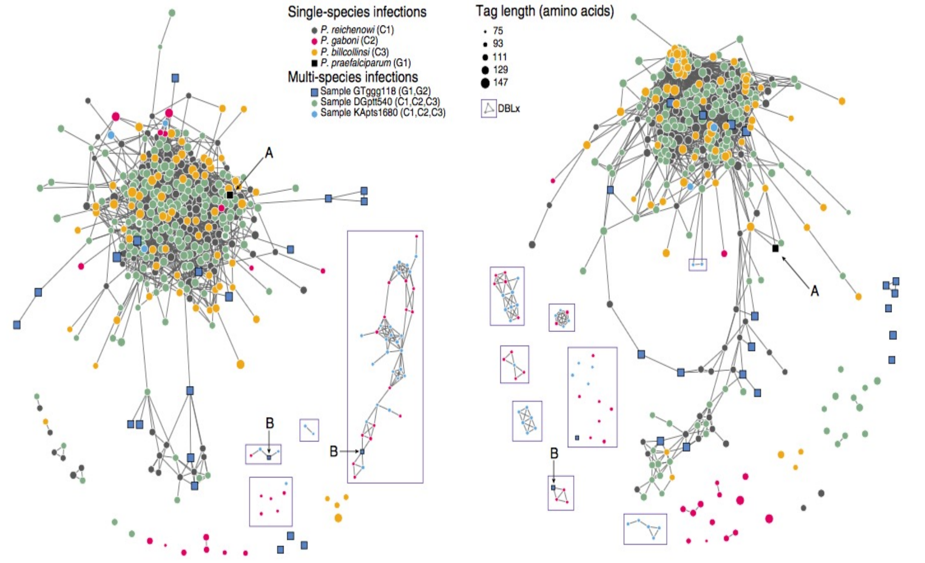


Figure 4 | Networks of DBL sequences from single- and multi-species *Laverania* infections. Each node represents a DBL HVR sequence and each link represents a shared amino-acid substring of significant length. Note that for each sample only unique *var* DBL haplotypes were included in the network analysis. Nodes with zero links indicate sequences that share no significant amino-acid substrings with other sequences. Networks were built separately for each HVR, where mosaic diversity is highest (see the Methods section). Circular nodes represent chimpanzee parasites and square nodes represent gorilla parasites. Node colour corresponds to species and node size corresponds to tag length as indicated. DBLx sequences are enclosed in boxes. Annotations call attention to (A) *P. praefalciparum* single-species infection sequence; (B) DBLx sequences from gorilla samples, hypothesized to be *P. adleri*, that share mosaic elements with DBLx chimpanzee parasites.

Figure 3: This graphic is adapted from original paper

3. **Discover DBLx domain in C2/G2 branch**: Here researchers investigate all the Laverania species by constructing HVR networks but exclude P.f, in this part, C1,C3 and (C1,C2,C3) are assembled together clearly, which is consistent with previous classification. Here authors offer one illustration about the placement of P.g, the P.g appear to fall into two subgroups, longer sequence group are partially overlapping with P.r and P.b, while the shorter sequence group are far away the main part, they are given two terms respectively, DBLα-like and DBLx-like.

   Then look at the samples of single-infection P.g and multi-species infection GTggg118(P.p and P.a), they discover both of them contain DBLx sequences from networks shown in Figure 4, they thus hypothesize DBLx found in GTggg118 comes from P.a, which is a sister taxa of P.g. Hence, it is highly likely DBLx sequences is a new subdomain in C2/G2 branch of Laverania radiation.

4. **Multi-domain structure shared among the whole Laverania species** : There are two parts in this section. First part tells us the

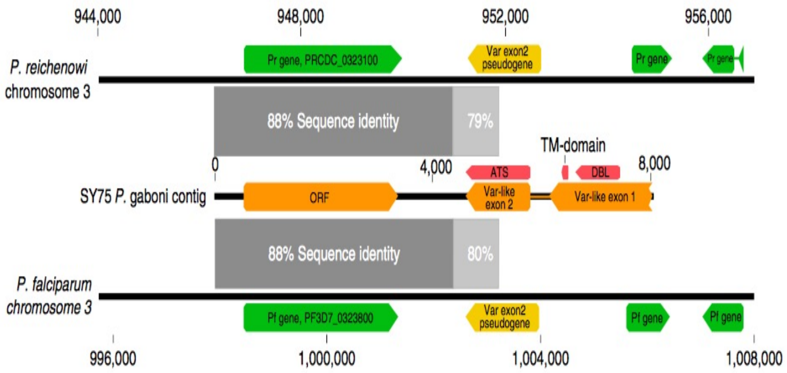


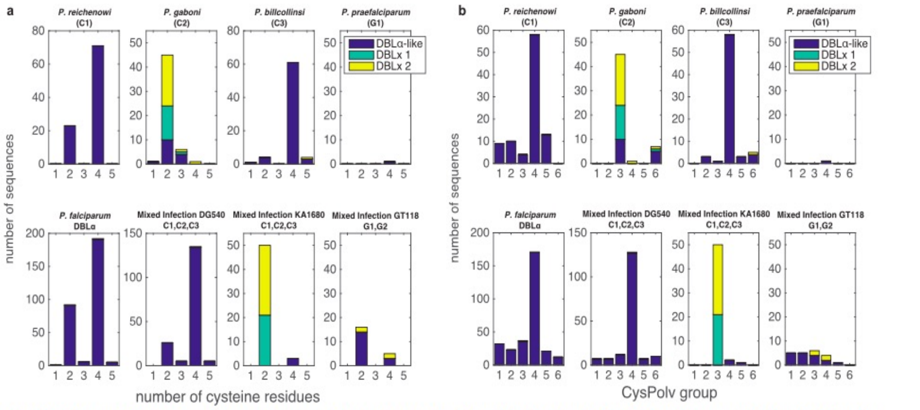
Figure 6 | Shared synteny of *var*-like genes in *P. falciparum*, *P. reichenowi* and *P. gaboni*. An open-reading frame (ORF) located downstream of a predicted *var*-like gene in *P. gaboni* showed 88% sequence identity (dark grey bars) with a single-copy gene present in both *P. falciparum* 3D7 (PF3D7_0323800) and *P. reichenowi* CDC1 (PRCDC_0323100). The *P. gaboni var*-like gene is syntenic with a *var* exon 2 pseudogene in both *P. falciparum* and *P. reichenowi*, suggesting that a *var* gene was present at this location in the ancestor of all three *Laverania* species.

Figure 4: This graphic is adapted from original paper

presence of var TAS domains in P.g by identifying 7 of 10known major HBs in P.f and P.r via VarDom server. The details are in Figure 5. In another story, based on Figure 5, ORF(Open reading Frame) share 88% nucleotide sequence identity with P.f and P.r in same chromosome 3. Apart from that, exon 2 is a single-copy of var exon2 pseudogene on chromosome 3 of both P.f and P.r. All these clues imply the existence of ancestral ORF and two-exon var structure.

5. **CP group classification in P.f and P.r could extend to P.b**: It has been previously published var genes could divided into two main group based on the number of cysteine residues, then according to the presence or absence of key amino acid residues, these two main group could be further subdivided into a total of six CP groups. Different groups are associated with different clinical phenotypes. Based on Supplementary Figure 6, we could find the P.b also exhibit same organization, both with cysteine residues count and CP groups. P.g are sot sufficiently to infer since it is lack of enough DBLα-like motifs.

Another two interesting points I need add is the identification of HVR and Bayesian K-mer analysis. Researchers identified HVRs using a sequence entropy approach, aligning HB3 firstly, then compute Shannon entropy and choose sequences which entropy was more than 2 bits, then repeat same step in HB5, last is HB2. Bayesian K-mer analysis is employed to estimate the overlap for global populations of P.f and P.r instead of between species for our currently available datasets. The overlap parameter p is views as beta distribution with parameters α and β, then calculate these two parameters



Supplementary Figure 6: Repertoire structure of *Laverania var* tag sequences by cysteine counts and CysPolv (CP) groups. Histograms show **(a)** the number of cysteine residues in each tag sequence, and **(b)** CP groups[2] for tag sequences, sorted by sample. Repertoires appear stably structured for *P. falciparum, P. reichenowi*, and *P. billcollinsi*. Data for other species and samples are too few to draw conclusions but are shown. DBLα-like, DBLx1, and DBLx2 tags are colored as indicated in legend.

Figure 5: This graphic is adapted from original paper

by maximizing the log-likelihood.

Lastly, one note: the whole structure of this paper is kind of different with normal publications. Authors place Result in front of Methods, and Methods are printed in smaller font size.

8 Paper 8: Hypervariable antigen genes in malaria have ancient roots

Zilversmit, Martine M., et al. BMC evolutionary biology 13.1 (2013): 110. The full pdf version please see here.

In this paper, using a novel HMM-based approach, researchers compare sequences of var gene DBLα domains from two divergent isolates in P.falciparum: 3D7 and HB3, and its sister taxa P.reichenowi. Results demonstrate this two species share similar gene size, and gene structure. To more specifically, they both have more than 51 var genes in a genome, and they both have a series of "homology blocks". In addition, through simulation, they found recombination occurs almost every residue in DBLα domain which appears to be unusual, and no hotspot structure for this type of intradomain recombination even though interdomain hotspot structure could be considered in previous Rask's study.

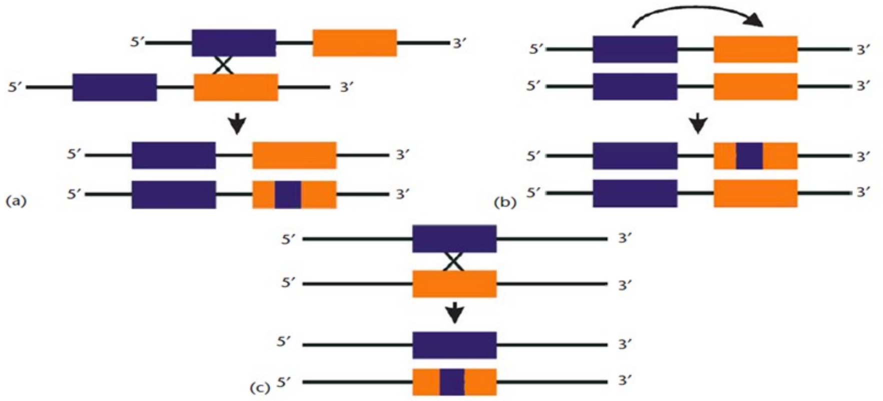
High level of sequence diversity in the PfEMP1 proteins, encoded by 50 to 60 var genes, express on the surface of infected red cells, provoking an immune response, and are known virulence factors. Previous studies show that recombination, combined with point mutation, is the mechanism of var gene revolution. Another noteworthy point is that since host's immune system is more effective against antigens, as a result, parasites expressing less-common proteins avoid detection more effectively.

DBLα domain, averaging 1.8kb in length, is the only functional domain in P.falciparum and P.reichenowi, and it has stable location as well.

using the tBLASTx algorithm and the prototype DBLα domain, researchers are able to extract reads, then using Clean Data assembly algorithm in Sequencer(GeneCodes), 51 unique DBLα regions are recovered, which is within the range of P.falciparum, thus, demonstrating that the family is equally large in both species. In further phylogenetic analysis of the var gene DBLα domains, P.reichenowi are not clustered together, indicating var genes likely arose as an entire family before the P.falciparum-P.reichenowi speciation event 2.5-6 million years ago.

Before introduction of method part, there are two pictures showing us the difference of gene conversion and crossover.

Obviously, gene conversion in second figure belongs to the third type (Interallelic gene conversion)shown in first figure.



Types of Gene Conversion. (a) Nonallelic or interlocus gene conversion events in trans [between nonallelic gene copies (shown as blue and orange boxes) residing on sister chromatids or homologous chromosomes]. (b) Interlocus gene conversion events in cis (between nonallelic gene copies residing on the same chromatid). Gene conversion events, depicted in (a) and (b), are virtually indistinguishable from each other. (c) Interallelic gene conversion events between alleles residing on homologous chromosomes. Adapted from Chen et al. (2007).

Figure 6: This graphic is adapted from Chen et al.(2007)

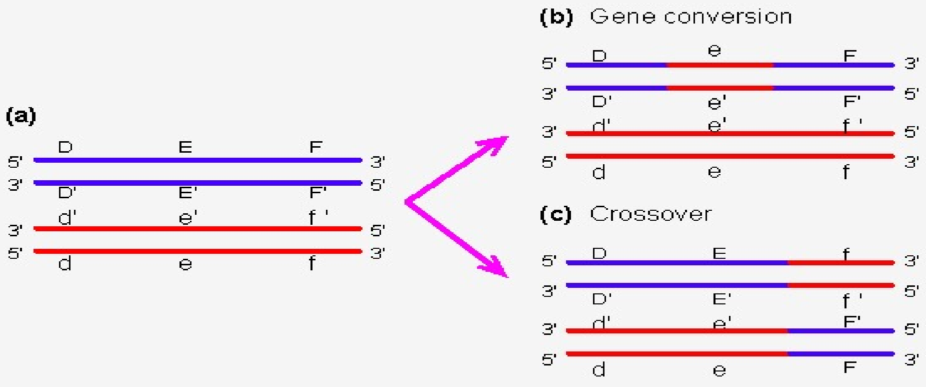


Figure 8-D-4. Comparison between gene conversion and DNA crossover. **(a)** Two DNA molecules. **(b)** Gene conversion - the red DNA **donates** part of its genetic information (e-e' region) to the blue DNA. **(c)** DNA crossover - the two DNAs **exchange** part of their genetic information (f-f' and F-F').

Figure 7: This graphic is adapted from webbook

New method is necessary to identify the relationship between species. In this paper, *Tesserae* program written in C language is used and implemented. HMM is employed to find homology, global alignment algorithm (Needleman-Wunsch) is to detect mosaic recombination, product of approximately conditionals (PAC) likelihood is utilized to estimate the recombination parameter. This paper aim to reconstruct each sequence in dataset as a mosaic of one or more donor sequences, allowing substitution, indel and recombination. The precise steps are as follows:

Initially, recombination parameter set to zero, transition and emission probabilities , indels and mutations are estimated using Baum-Welch algorithm, then fix these parameters, a likelihood surface is constructed for the recombination parameter. Once the MLE for recombination parameter is found, Viterbi path is computed for each sequence, this path provide the mosaic alignments.

For simulated sequences, they construct 10 gene families, each family has 60 genes, each gene is composed of 150 amino acid residues in length, and a table of input parameters are needed(Table S2 in paper). Each gene family is used for 8 different sets of parameters, namely, different levels of recombination and conversion. Each gene family's simulation could viewed as two groups, one is indels without recombination, another one is coalescent with recombination, since there is currently no available program for joint estimation about recombination and indel.

Here we have to note that the coalescent is likely to be an inaccurate description of the true var gene evolution. However, basic coalescent processes of coancestry and allelic recombination may represent var gene duplication and non-allelic recombination, thus, making aspects of coalescent represent several features of var gene family evolution. Here authors employ calibration method, allowing them to make comparisons between the ρ and recombination parameter in coalescent models.

Firstly, comparisons of the exact values show a high level of accuracy in the estimated recombination rates, through computing the difference of likelihood with and without recombination for each sequence, the statistical significance of the improvement as p less than $1 * 10^{-32}$.

In order to test false positives further, they uses a non-recombining data from P.falciparum, the program is able to recover all known recombination event, while finding no recombination history, as expected.

In further examination of the P.falciparum and P.reichenowi DBLα domain homology regions, researchers find multiple regions pf particular high

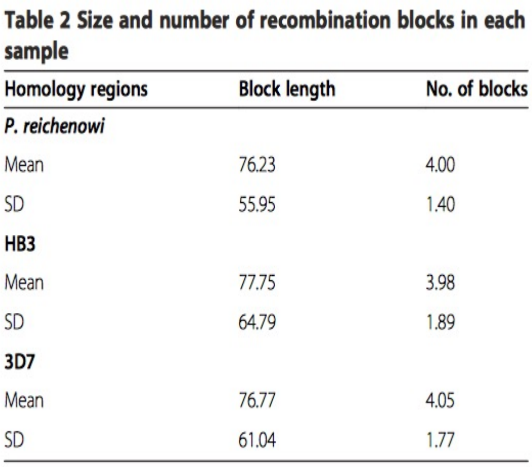


Table 2 Size and number of recombination blocks in each sample

Homology regions	Block length	No. of blocks
P. reichenowi		
Mean	76.23	4.00
SD	55.95	1.40
HB3		
Mean	77.75	3.98
SD	64.79	1.89
3D7		
Mean	76.77	4.05
SD	61.04	1.77

Figure 8: This graphic is adapted from original paper

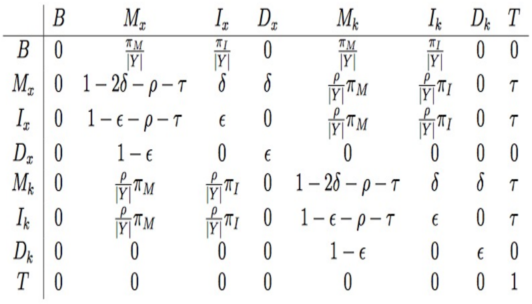
homology, classified into two groups: core motifs and conserved peptides. The former one is between 18 and 28 residues, corresponding to HB1-5(termed "homology blocks" by Rask et al.), among them, HB2 motif are he most frequently ones, then is HB3 and HB5. The latter one is between 24 and 140 residues, and between 80% and 100% similarity.

Lastly, the most important analysis result is recombination is uniform throughout the DBLα domain and does not show a hot- or coldspot structure, that means recombination breaks at almost every residue. From the following table, high variance of block length indicate the lack of hot - or coldspots of recombination.

One basic introduction I would add is the programs generating sequences , they are Seq-Gen, ms and Rose, which are all shown in the first column of Table S2.

Seq-Gen simulates sequences given a substitution model along a phylogeny. Ms, is an all-time classic engine for coalescent sequence simulations under a Wright-Fisher neutral model. and Rose, simulates sequences given a substitution model along a phylogeny incorporating indels. This introduction is originally from a web link.

Here is a summary of talk between Heejung and me on Monday and Tuesday afternoon in the first week of March 2018. Our talk focus on the explanation of the following transition matrix allowing insertion and deletion events.



	B	M_x	I_x	D_x	M_k	I_k	D_k	T
B	0	$\frac{\pi_M}{\lvert Y\rvert}$	$\frac{\pi_I}{\lvert Y\rvert}$	0	$\frac{\pi_M}{\lvert Y\rvert}$	$\frac{\pi_I}{\lvert Y\rvert}$	0	0
M_x	0	$1-2\delta-\rho-\tau$	δ	δ	$\frac{\rho}{\lvert Y\rvert}\pi_M$	$\frac{\rho}{\lvert Y\rvert}\pi_I$	0	τ
I_x	0	$1-\epsilon-\rho-\tau$	ϵ	0	$\frac{\rho}{\lvert Y\rvert}\pi_M$	$\frac{\rho}{\lvert Y\rvert}\pi_I$	0	τ
D_x	0	$1-\epsilon$	0	ϵ	0	0	0	0
M_k	0	$\frac{\rho}{\lvert Y\rvert}\pi_M$	$\frac{\rho}{\lvert Y\rvert}\pi_I$	0	$1-2\delta-\rho-\tau$	δ	δ	τ
I_k	0	$\frac{\rho}{\lvert Y\rvert}\pi_M$	$\frac{\rho}{\lvert Y\rvert}\pi_I$	0	$1-\epsilon-\rho-\tau$	ϵ	0	τ
D_k	0	0	0	0	$1-\epsilon$	0	ϵ	0
T	0	0	0	0	0	0	0	1

Figure 9: This graphic is adapted from original paper

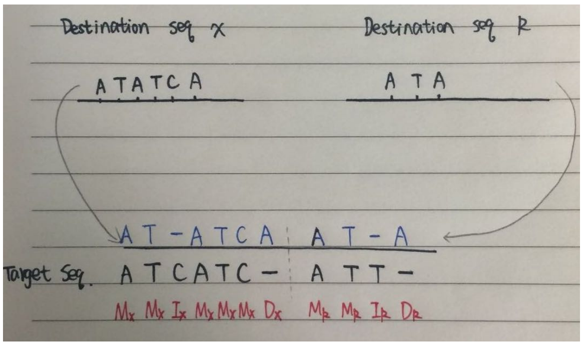


Figure 10: This graphic is drawn by myself

Firstly we have n destination sequences and one target sequence, assuming this target sequence as a mosaic of segments from those destination sequences. In this matrix, subscript "x" represents the target sequence and "k" represents any other destination sequence in the dataset.

Firstly, this matrix is not a symmetric matrix. For example, elements in the first column are not the same with elements in the first row.

The meaning of $M_x, I_x, D_x, M_k, I_k, D_k$ are shown in the following picture:

If we look at the two destination sequences in this figure, there are $6 + 3 = 9$ positions, at first, the start position uniformly come from this nine positions, therefore, every position has the same probability $1/9$, representing $1/|Y|$ in the transition matrix. Next, let's look at this matrix row by row, I will use T_{ij} represent the element in row i and column j.

In the first row, T_{11} and $T_{18}$are both 0, which are sensible, the target sequence's start point come from and only come from Match or insert, not

Delete. So $T_{14}$and T_{17} are 0, moreover, the start point may come from one of sequence x or sequence k , so my understanding is the first four elements are independent of the last four elements. Lastly, if start position comes from sequence x, it is match or insertion, so I suppose:

$\pi_M + \pi_I = 1$

In the second row, gap open probability is both δm if we jump to another sequence k, then will add recombination probability. Here we should notice that T_{27} is 0, which means the start point in target sequence should not be a gap when we jump to another sequence.

In the third row, $T_{34} = T_{37} = 0$, here we can suppose there is no allowance to transition from insertion to deletion, gap extension is allowed, of course.

In the fourth row, There is no allowance to transition from deletion to insertion. gap extension is allowed, of course.

The last four rows also show the same rule with the first four rows. Therefore, we should notice that the last two positions shown in picture should not happen in practice!

Last point I would add is gap **should** also exist in target sequence. In the simulation, these gaps is sampled from the statonary distribution of the emission matrix.

9 Paper 9: Plasmodium falciparum erythrocyte membrane protein 1 diversity in seven genomes–divide and conquer

Rask, Thomas S., et al. PLoS computational biology 6.9 (2010): e1000933. The full pdf version please see here.

Researchers analysis 399 var sequences (number varies from 39 to 63)from seven P.falciparum genomes (3D7,HB3,DD2,IT,IGH,RAJ116 and PFCLIN), including four Asian, two African and one Central American isolate. Authors redefine and reassess the identification and classification of var genes. In addition, a novel iterative homology detection method is proposed and is potentially applicable any other compositional analysis for protein or gene families.

9.1 Figure 1

Figure1.A is a schematic representation of structure of var gene locus. It begins 5$'$UTR and ends $3'UTR$. It has two exons, containing the combination of DBL and CIDR domains and ATS respectively. It also has NTS and TM region. Distance tree analysis confirmed the grouping of DBL into six major classes($\alpha, \beta, \gamma, \delta, \epsilon and \zeta$) and cysteine rich interdomain regions(CIDR) of five different classes($\alpha, \beta, \gamma, \delta and pam$). NTS sequences are divided into three classes, NTSA, NTSB and NTSpam while ATS sequences are divided into ATSA, ATSB, ATSPAM, ATSvar1, and ATSvar3. Moreover, UPS could be identified with these subgroups: UPSA1-2,UPSB1-4,UPSC1-2,UPSE, and additional UPSA3, UPSB5-7. var 1, var 3 and var2csa are three most conserved var genes, and UPSE only is found in var2csa, shown in Figure1.C.

From Figure1.B, we could see there are four components in these var genes, among them, component 1($DBL\alpha$ and CIDR α) occurs 95% of var genes, follows with component 3.

based on Figure1.D, DBD domain is divided into three structural subdomains, and CIDR is divided into two subdomains(previously studies show three subdomains). The numbered blocks represents the core homology blocks in all DBL (HB2,3,4,5)and CIDR domains(HB1,8) or both domain(HB1).

9.2 Figure 2

Figure 2 tells us the basic composition and other information about each subdomain in DBL and CIDR. Regarding average sequence length, DBL

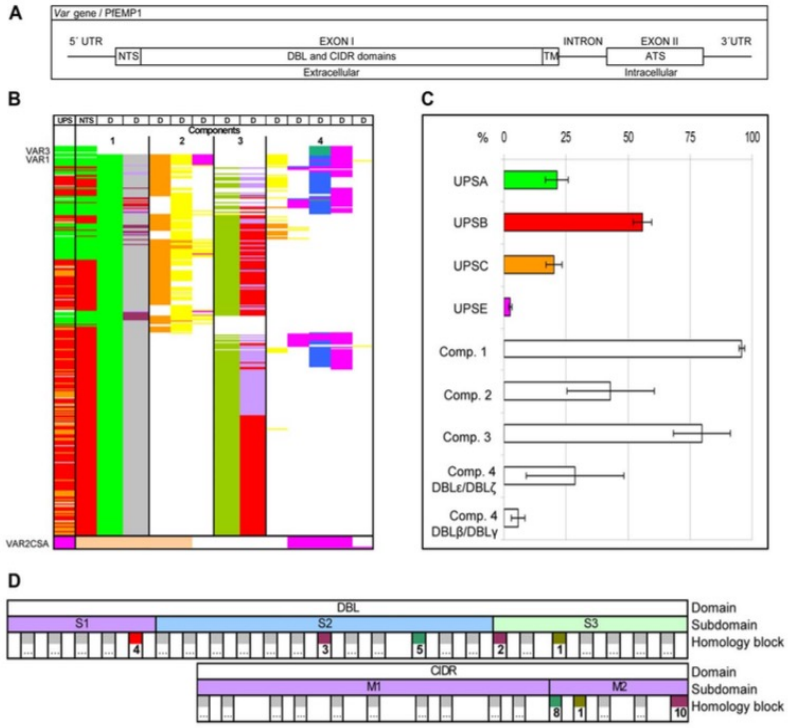
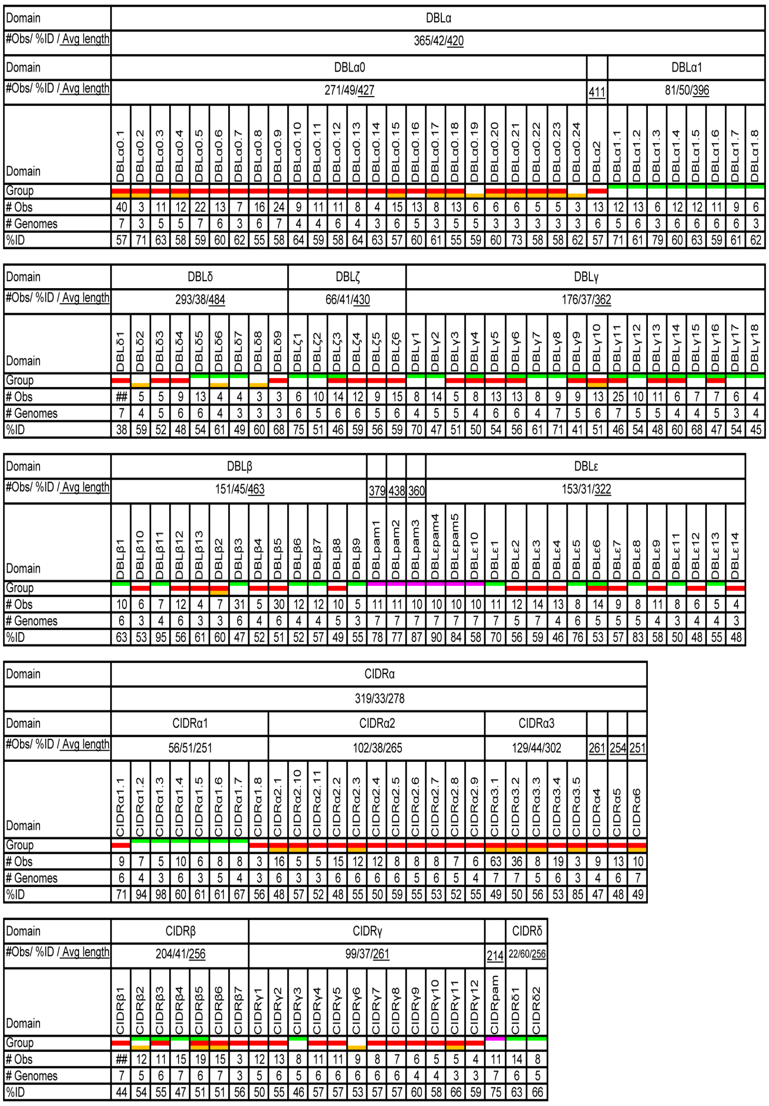


Figure 1. PfEMP1 annotation overview. (A) Schematic of the *var* gene locus. (B) 399 *var* exon1 annotated with UPS class and encoded major NTS, DBL and CIDR domain classes and their arrangement in four components. Color code for UPS column: Green: UPSA; Red: UPSB; Orange: UPSC; Pink: UPSE. Color code for NTS column: Green NTSA, Red: NTSB, Cream: NTSpam. Color code for DBL and CIDR domains (D columns): Bright Green: DBLα; Orange: DBLβ; Yellow: DBLγ; Olive green: DBLδ; Pink: DBLε; Blue: DBLζ; Blue stripes: DBLα of VAR3. Grey: CIDRα; Red: CIDRβ; Light purple: CIDRγ; Dark purple: CIDRδ. (C) Average distribution (% +/− 95% confidence intervals) of UPSA–E flanked and component 1–4 containing genes in the seven sequenced genomes 3D7, HB3, DD2, IT4, PFCLIN, RAJ116 and IGH. (D) Schematic presentation of DBL and CIDR subdomains and homology blocks. The numbered blocks represent the core homology blocks found in all DBL domains (HB2, 3, 4 and 5), all CIDR domains (HB8 and 10) or both domain types (HB1), further described in Figure 5.
doi:10.1371/journal.pcbi.1000933.g001

Figure 11: This graphic is adapted from original paper



Domain	DBLα																																
#Obs/ %ID / Avg length	365/42/420																																
Domain	DBLα0																									DBLα1							
#Obs/ %ID / Avg length	271/49/427																								411	81/50/396							
Domain	DBLα0.1	DBLα0.2	DBLα0.3	DBLα0.4	DBLα0.5	DBLα0.6	DBLα0.7	DBLα0.8	DBLα0.9	DBLα0.10	DBLα0.11	DBLα0.12	DBLα0.13	DBLα0.14	DBLα0.15	DBLα0.16	DBLα0.17	DBLα0.18	DBLα0.19	DBLα0.20	DBLα0.21	DBLα0.22	DBLα0.23	DBLα0.24	DBLα2	DBLα1.1	DBLα1.2	DBLα1.3	DBLα1.4	DBLα1.5	DBLα1.6	DBLα1.7	DBLα1.8
Group	🟥🟧	🟥🟧	🟥	🟥🟧	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥🟧	🟥	🟥🟧	🟥🟧	🟧	🟥🟧	🟥🟧	🟥🟧	🟥🟧	🟧	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
# Obs	40	3	11	12	22	13	7	16	24	9	11	11	8	4	15	13	8	13	6	6	6	5	5	3	13	12	13	6	12	12	11	9	6
# Genomes	7	3	5	5	7	6	3	6	7	4	4	6	4	3	6	5	3	5	5	3	3	3	3	3	6	5	6	3	6	6	6	6	3
%ID	57	71	63	58	59	60	62	55	58	64	59	58	64	63	57	60	61	55	59	60	73	58	58	62	57	71	61	79	60	63	59	61	62

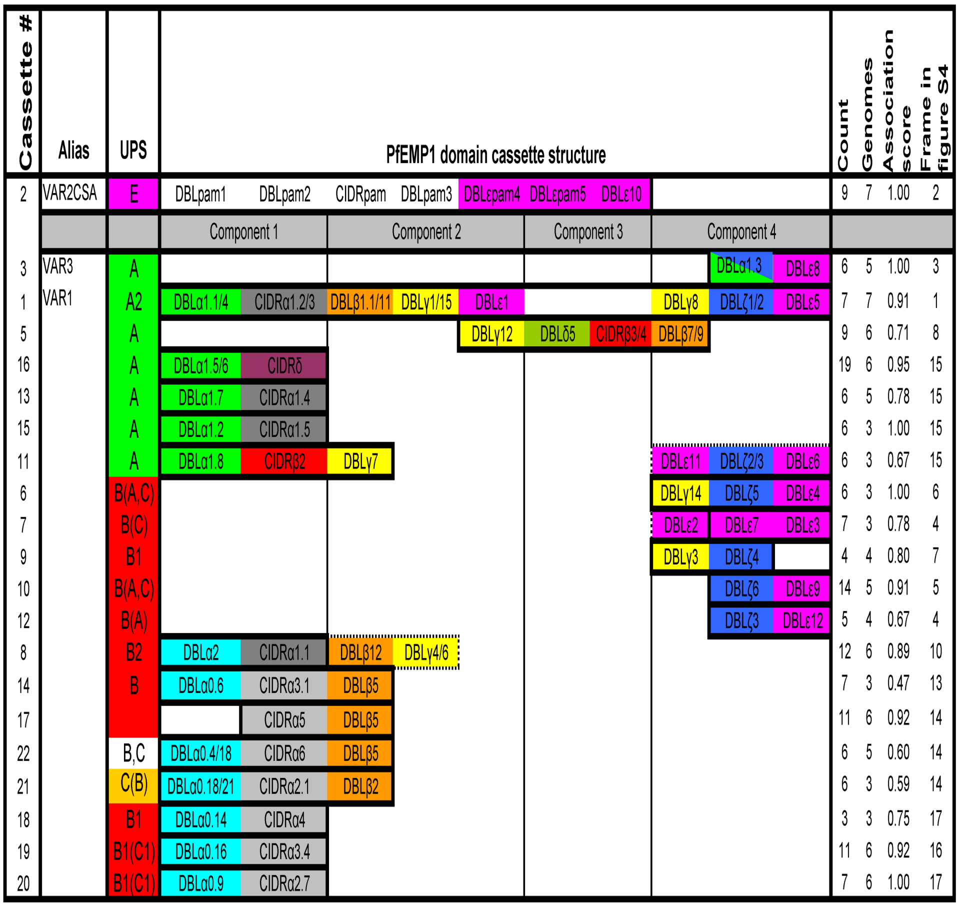
Domain	DBLδ									DBLζ						DBLγ																	
#Obs/ %ID / Avg length	293/38/484									66/41/430						176/37/362																	
Domain	DBLδ1	DBLδ2	DBLδ3	DBLδ4	DBLδ5	DBLδ6	DBLδ7	DBLδ8	DBLδ9	DBLζ1	DBLζ2	DBLζ3	DBLζ4	DBLζ5	DBLζ6	DBLγ1	DBLγ2	DBLγ3	DBLγ4	DBLγ5	DBLγ6	DBLγ7	DBLγ8	DBLγ9	DBLγ10	DBLγ11	DBLγ12	DBLγ13	DBLγ14	DBLγ15	DBLγ16	DBLγ17	DBLγ18
Group	🟥	🟧	🟥	🟥	🟩	🟩🟧	🟩	🟧	🟥	🟩	🟩	🟩🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟩🟥	🟥	🟩🟥	🟩	🟩	🟩🟥	🟥🟧	🟩🟥	🟩	🟩🟥	🟩🟥	🟩	🟩🟥	🟩	🟩
# Obs	##	5	5	9	13	4	4	3	3	6	10	14	12	9	15	8	14	5	8	13	13	8	9	9	13	25	10	11	6	7	7	6	4
# Genomes	7	4	5	6	6	4	3	3	3	6	5	6	6	5	6	4	5	5	4	6	6	4	7	5	6	7	5	5	4	4	5	3	4
%ID	38	59	52	48	54	61	49	60	68	75	51	46	59	56	59	70	47	51	50	54	56	61	71	41	51	46	54	48	60	68	47	54	45

Domain	DBLβ																DBLε															
#Obs/ %ID / Avg length	151/45/463													379	438	360	153/31/322															
Domain	DBLβ1	DBLβ10	DBLβ11	DBLβ12	DBLβ13	DBLβ2	DBLβ3	DBLβ4	DBLβ5	DBLβ6	DBLβ7	DBLβ8	DBLβ9	DBLpam1	DBLpam2	DBLpam3	DBLεpam4	DBLεpam5	DBLε10	DBLε1	DBLε2	DBLε3	DBLε4	DBLε5	DBLε6	DBLε7	DBLε8	DBLε9	DBLε11	DBLε12	DBLε13	DBLε14
Group	🟩	🟥	🟩	🟥	🟥	🟥🟧	🟩	🟥	🟥	🟩	🟩	🟥	🟩	🟪	🟪	🟪	🟪	🟪	🟪	🟩	🟥	🟥	🟥	🟩	🟩🟥	🟥	🟩	🟥	🟩	🟥	🟩	🟥
# Obs	10	6	7	12	4	7	31	5	30	12	12	10	5	11	11	10	10	10	10	11	12	14	13	8	14	9	8	11	8	6	5	4
# Genomes	6	3	4	6	3	3	6	4	6	4	4	5	3	7	7	7	7	7	7	7	5	7	4	6	5	5	5	4	3	4	4	3
%ID	63	53	95	56	61	60	47	52	51	52	57	49	55	78	77	87	90	84	58	70	56	59	46	76	53	57	83	58	50	48	55	48

Domain	CIDRα																										
	319/33/278																										
Domain	CIDRα1								CIDRα2											CIDRα3							
#Obs/ %ID / Avg length	56/51/251								102/38/265											129/44/302					261	254	251
Domain	CIDRα1.1	CIDRα1.2	CIDRα1.3	CIDRα1.4	CIDRα1.5	CIDRα1.6	CIDRα1.7	CIDRα1.8	CIDRα2.1	CIDRα2.10	CIDRα2.11	CIDRα2.2	CIDRα2.3	CIDRα2.4	CIDRα2.5	CIDRα2.6	CIDRα2.7	CIDRα2.8	CIDRα2.9	CIDRα3.1	CIDRα3.2	CIDRα3.3	CIDRα3.4	CIDRα3.5	CIDRα4	CIDRα5	CIDRα6
Group	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥🟧	🟥🟧	🟥	🟥	🟥🟧	🟥	🟥	🟥	🟥	🟥	🟥	🟥🟧	🟥🟧	🟥🟧	🟥	🟥🟧	🟥	🟥	🟥🟧
# Obs	9	7	5	10	6	8	8	3	16	5	5	15	12	12	8	8	8	7	6	63	36	8	19	3	9	13	10
# Genomes	6	4	3	6	3	5	4	3	6	3	3	6	6	6	6	5	6	5	4	7	7	5	6	3	4	6	7
%ID	71	94	98	60	61	61	67	56	48	57	52	48	55	50	59	55	53	52	55	49	50	56	53	85	47	48	49

Domain	CIDRβ							CIDRγ													CIDRδ	
#Obs/ %ID / Avg length	204/41/256							99/37/261												214	22/60/256	
Domain	CIDRβ1	CIDRβ2	CIDRβ3	CIDRβ4	CIDRβ5	CIDRβ6	CIDRβ7	CIDRγ1	CIDRγ2	CIDRγ3	CIDRγ4	CIDRγ5	CIDRγ6	CIDRγ7	CIDRγ8	CIDRγ9	CIDRγ10	CIDRγ11	CIDRγ12	CIDRpam	CIDRδ1	CIDRδ2
Group	🟥	🟩🟧	🟩🟥	🟩	🟩🟥🟧	🟥🟧	🟥	🟥	🟥	🟩	🟥	🟥	🟧	🟥	🟥	🟥	🟥	🟥🟧	🟥	🟪	🟩	🟩
# Obs	##	12	11	15	19	15	3	12	13	8	11	11	9	8	7	6	5	5	4	11	14	8
# Genomes	7	5	6	7	6	7	3	5	6	5	6	6	6	6	6	4	4	3	3	7	6	5
%ID	44	54	55	47	51	51	56	50	55	46	57	57	53	57	57	60	58	66	59	75	63	66

Figure 12: This graphic is adapted from original paper



Cassette #	Alias	UPS	PfEMP1 domain cassette structure				Count	Genomes	Association score	Frame in figure S4
2	VAR2CSA	E	DBLpam1 DBLpam2	CIDRpam DBLpam3 DBLεpam4	DBLεpam5 DBLε10		9	7	1.00	2
			Component 1	Component 2	Component 3	Component 4				
3	VAR3	A				DBLα1.3 DBLε8	6	5	1.00	3
1	VAR1	A2	DBLα1.1/4 CIDRα1.2/3	DBLβ1.1/11 DBLγ1/15 DBLε1		DBLγ8 DBLζ1/2 DBLε5	7	7	0.91	1
5		A		DBLγ12	DBLδ5 CIDRβ3/4	DBLβ7/9	9	6	0.71	8
16		A	DBLα1.5/6 CIDRδ				19	6	0.95	15
13		A	DBLα1.7 CIDRα1.4				6	5	0.78	15
15		A	DBLα1.2 CIDRα1.5				6	3	1.00	15
11		A	DBLα1.8 CIDRβ2	DBLγ7		DBLε11 DBLζ2/3 DBLε6	6	3	0.67	15
6		B(A,C)				DBLγ14 DBLζ5 DBLε4	6	3	1.00	6
7		B(C)				DBLε2 DBLε7 DBLε3	7	3	0.78	4
9		B1				DBLγ3 DBLζ4	4	4	0.80	7
10		B(A,C)				DBLζ6 DBLε9	14	5	0.91	5
12		B(A)				DBLζ3 DBLε12	5	4	0.67	4
8		B2	DBLα2 CIDRα1.1	DBLβ12 DBLγ4/6			12	6	0.89	10
14		B	DBLα0.6 CIDRα3.1	DBLβ5			7	3	0.47	13
17			CIDRα5	DBLβ5			11	6	0.92	14
22		B,C	DBLα0.4/18 CIDRα6	DBLβ5			6	5	0.60	14
21		C(B)	DBLα0.18/21 CIDRα2.1	DBLβ2			6	3	0.59	14
18		B1	DBLα0.14 CIDRα4				3	3	0.75	17
19		B1(C1)	DBLα0.16 CIDRα3.4				11	6	0.92	16
20		B1(C1)	DBLα0.9 CIDRα2.7				7	6	1.00	17

Figure 13: This graphic is adapted from original paper

has much longer sequences than CIDR significantly, regarding observation values, whole observation values between DBL and CIDR are similar. One noteworthy point is most of the $DBL\delta$ sequences could not be subclassified, same as $CIDR\beta$. In addition, most classes could be linked to one specific UPS class.

In conclusion, this classification is based on domain similarities averaged over the whole domains. The validity of the classification must be experimentally tested further.

9.3 Figure 3

A PfEMP1 domain cassette is defined as a var gene sequence encoding two or more DBL or CIDR domains with subclasses that could be predicted from each other. The three conserved var genes var 1, var 3 and var2csa , all encoding unique DBL domains, are present in all seven genomes, except var 3 which is not in HB3 and IGH. The domain composition variation within these three genes highlight the importance of ectopic recombination for the generation of PfEMP1 diversity.

Result also shows there is no basic difference between PfEMP1 repertoires around the world. Var2CASA and its relevance in pregnancy malaria is well established, apart from that , some studies emphasize the importance of group A PfEMP1 in severe malaria, and often the particularly group A domain cassette 5.

9.4 Novel iterative homology detection method

This method is potentially applicable to any other protein dataset, and would be suitable for compositional analysis of other frequently recombining gene families. Homology blocks(HB) cover on average 83% of a PfEMP1 sequence. The HB analysis also revealed a recombination hotspot between subdomain S2 and S3 in DBL domains (around HB2). The homology blocks were numbered according to the frequency in the seven genome dataset, with the most frequent being HB number one.

VarDom server provides a chance to classify related homology blocks and domains after submitting a new sequence into it.

Actually writer also mention the composition of HBs in DBL, CIDR, NTS and ATS, I am not that interested, especially when they introduce the crystal structure of this genes.

Let's focus on the novel iterative homology detection method(Figure 14):

A serial iterative approach was employed, where per iteration could generate only one homology block, the most conserved sequence in the database. Subsequently the members of the selected homology block were removed from the database to avoid overlap in the following iteration. Three steps are implemented to uncover the most conserved homology block:

(1) Up to 100 different seed sequences were roughly selected using BLAST, each to potentially form a homology block. Ungapped BLAST was initially used to select seed sequences. These seed sequences should only cover one homology block each. Then normal gapped BLAST was used to detect homology which had escaped from the ungapped BLAST.

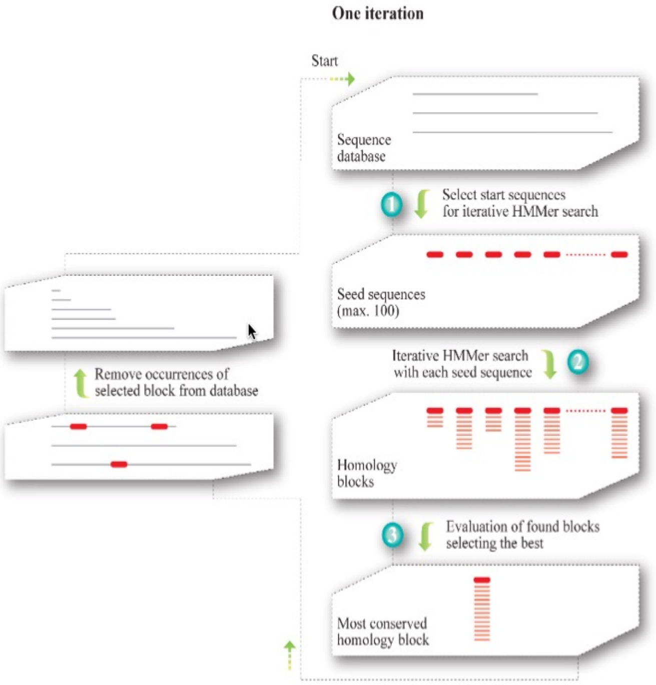


Figure A: One iteration defining one homology block. Flow diagram describing one iteration in the process of automatically defining a set of homology blocks covering the sequence database. Different stages in the iterative process are marked by boxes, processes are marked by arrows. The steps 1-3 are elaborated in Figure B, C, and D.

One iteration
Start
Sequence database
1
Select start sequences for iterative HMMer search
Seed sequences (max. 100)
Iterative HMMer search with each seed sequence
2
Homology blocks
3
Evaluation of found blocks selecting the best
Most conserved homology block
Remove occurrences of selected block from database

Figure 14: This graphic is adapted from text S2 in original paper's supporting materials

(2) Starting from a single query sequence selected in step 1. HMMs were built by iterative HMMer (iHMMer) algorithm using the HMMer package. The results from iHMMer consists of a multiple sequence alignment defining an HMM, where the HMM can refind the exact definition sequences.

(3) One optimized homology block was finally selected, by taking into account both the number of hits as a measure of conservation, but also how many times the same block occurred, as a measure of how well parameter space had been sampled for that specific homology block, and thus how likely it was that the block was optimal.

10 Book 1: Phylogenetic networks: concepts, algorithms and applications.

Huson, Daniel H., Regula Rupp, and Celine Scornavacca. Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press, 2010. The full pdf version please see here.

Reassortment is the mixing of the genetic material of a species into new combinations in different individuals. It is particularly used when two similar viruses that are infecting the same cell exchange genetic material. [1]

There are several interesting concepts in graph theory. Please look at the Figure 1.1, 1.2 and 1.3.

1. The *degree* of node u is the sum of its indegree and outdegree.
2. A *biconnected components* is the maximal subgraph that is induced by set of edges and doesn't contain a cut node. A good example is in Figure 1.3.
3. A graph $G = (V, E)$ is called *bipartite* if and only if its set of nodes can be partitioned into subsets V_1, V_2,with $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$, such that for every edge $e \in E$,one of the endpoints lies in V_1, another endpoint lies in V_2.
   Explain Exercise 1.2.3.
4. Two nodes v and w are *imcomparable*, if neither node is lower than other; similarily, two edges e and f are *imcomparable* if neither is lower than other;
5. Different *traversals* give rise to different orders in which nodes are examined. Pay attention to preorder, postorder and breadth-first traversal in Figure 1.7. In particular, breadth-first traversal, please reference here.

PS: Nodes are also called vertices, edges are also called branches or arcs.
Bifurcating Tree = Resolve Tree = Binary Tree

Let $\chi = \{x_1, x_2, ...x_n\}$ be a set of taxa, a *cluster* is any subset of χ,excluding the empty set $\emptyset$ and full set χ. The ultimate goal of phylogenetic analysis is to compute a set of clusters on χ such that each cluster is monophyletic(also called clade. Monophyletic group contains all descendants of the common ancestor and the ancestor itself).

A *split* is any bipartitioning of χ into two non-empty subsets A and B of χ, such that $\chi = A \cup B$ and $A \cap B = \emptyset$.

In phylogenetic analysis, a set of taxa $\chi = \{x_1, x_2, ...x_n\}$ is often represented by a set of molecular sequences $A = \{a_1, a_2, ...a_n\}$ where a_i comes from taxon x_i and correspond to some specific genes or locus. We also need to ensure that the sequences are **homologous**, that is, have evolved from a common ancestor sequence.

In *Pairwise sequence alignment* , with the help of substitution matrix, for example the BLOSSUM matrix, which assigns empirically score, we could calculate the score of each pair of residues and then sum over scores among all pairs would be the score of whole alignment.

Sequence are often aligned by inserting gaps into each sequence shown in Figure 2.6 such that all sequences have same length m, forming a *multiple sequence alignment* of length m. Our goal is to find a multiple sequence alignment that achieves the optimal score according to an appropriate score scheme. **Progressive method** as a heuristic approach, is used to align multiple sequences, its outline is shown in Figure 2.7. The core is to align a pair of similar sequences into *profiles*, then align profiles into final multiple sequence alignment.

Let M be a multiple sequence alignment on χ, each column of M is called a character, each symbol that occurs in this column is called a character state.

Now we are introducing some basic concepts and main methods for inferring phylogenetic trees.

Phylogenetic trees are usually computed from molecular sequences. They not only could uncover the relationship between different species or taxa, but also have many other applications. For instance, they are used to determine the age and the rate of diversification. In sequence-analysis method, they are allowed *phylogenetic footprinting.*

In practice, there are two types of analysis after the initial multiple sequence alignment: distance-based analysis and sequence-based one. Its outline is shown in Figure 3.1 at page 24.

Definition Phylogenetic Tree
Given a set of taxa χ, this is a phylogenetic tree $T = (V, E)$, its all nodes have degree $\neq 2$, together with a taxon labeling $\lambda : \chi \to V$that assigns actually one taxon to every leave and none to internal nodes.

From a theoretical and algorithmic point of view, unrooted phylogenetic trees are much more easier than rooted ones, however, in biology, rooted phylogenetic trees are usually more of interest. A phylogenetic tree is called an edge-weighted tree if we are given a map ω that assigns a non-negative weight or length $\omega(e)$ to every edge e of the tree. In drawings, we usually use length of the edge to indicate the scale rather than write the lengths explicitly next to edges.

Jukes-Cantor model tells us the probability formula of change during time t or along the edge, given the mutation rate. This model of DNA evolution assumes the fours bases (A,C,G and T) occur with equal frequencies(0.25) and change from one base to another occurs at the same rate. If we relax the conditions, for example, let the bases occur at different and arbitrary rates (although they have to sum to 1), change rates in transitions and transversions, then we could get more generate model, anyway, they are both special cases of general time reversible model.

Classical phylogenetic trees construction approaches consist of two following types:

* **Sequence-based method** usually searches for best phylogenetic tree which can optimally explain the given multiple sequence alignment M. We discuss the three main approaches about it: maximum parsimony, ML and Bayesian inference.

* **Distance-based method** usually constructs phylogenetic tree from a given a distance matrix D.

10.1 Maximum parsimony

Maximum parsimony method is to look for a phylogenetic tree that explains the given set of aligned sequences using a minimum number of evolutionary events. The premise of parsimony method needs to input a multiple sequence alignment.

The *parsimony score* of T(tree) and M(given multiple sequence alignment) is defined as:

$$PS(T, M) = \min_{\alpha} \sum_{\{x,y\}} diff(x, y) \tag{8}$$

where diff() function is known as *hamming distance* between sequence $x = (x_1, x_2, ...x_m)$and sequence $y = (y_1, y_2, ...y_m)$ that describes the difference of $x$and$y$

$$diff(x, y) = |\{i|x_i \neq y_i\}| \tag{9}$$

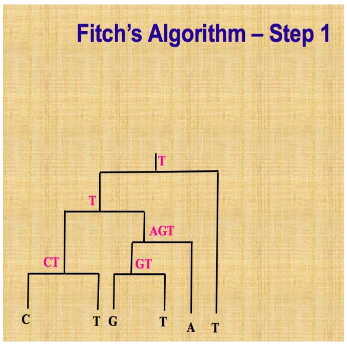
The minimum is taken over all possible assignments α that make the sequences of length m to be the internal nodes, summation is taken over all possible pairs of x and y that are assigned at opposite end of edge of T .

This task of computing parsimony score is the known *small parsimony* problem. In small parsimony problem, input is aligned sequences and a tree with sequences at leaves, output is an sequences assignment of all internal nodes in this tree with minimum number of changes across all edges. For bifurcating tree, *Fitch algorithm* is used to calculate efficiently, in more general settings, *Sankoff's algorithm* can be applied.

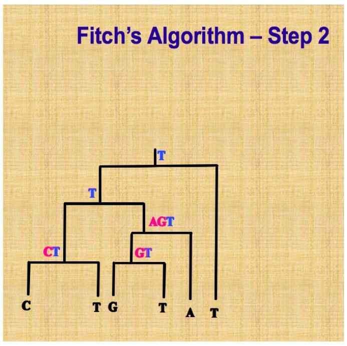
Let's describe Fitch algorithm which has a linear time when solving above small parsimony problem. Assume we are given a multiple sequence alignment M and a bifurcating tree T on χ, we need to score each character (that is, column of the alignment) separately, and then obtain the parsimony score $PS(T, M)$ by summing over all characters.

This algorithm proceeds in two parts, as shown in following pictures. The first part is called *bottom up* phrase, from leaves to root, finding sets of possible ancestral states (labels) for each internal node, next part is *top down* process, from root to leaves, determining ancestral states (labels) for internal nodes. Different site is independent, so we can solve one site at a time.

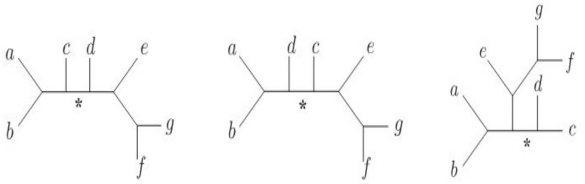
On the contrary, there is *big parsimony*(called large parsimony also). It's related with a search through the space of trees. In big parsimony problem, input is only aligned sequences, output is a labeled tree with minimum number of changes across all edges (over all trees). It is a NP hard problem.



Fitch's Algorithm – Step 1



Fitch's Algorithm – Step 2



(a) Phylogenetic tree (b) One NNI tree (c) Other NNI tree

Figure 3.14 Nearest neighbor interchange (NNI). (a) An unrooted phylogenetic tree T. The four subtrees attached to the two ends of the edge marked "$*$" can be interchanged in four different ways, leading to two distinct phylogenetic trees, shown in (b) and (c).

10.2 Branch-swapping method

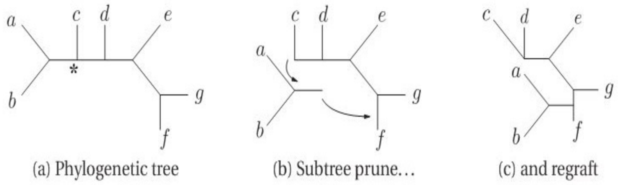
When researchers explore the bifurcating trees space, they actually move from one bifurcating tree T to another bifurcating tree T' by applying *Branch-swapping method* to rearrange a part of tree T. There are three methods : NNI(Nearest neighbor interchange),SPR(Subtree prune and regraft) and TBR(Tree bisection and reconnection). In order to show this three methods vividly, please look at the figures 3.14-3.16 which are all from this book. Let NNI(T), SPR(T) and TBR(T) represent the set of all possible trees that can be obtained by applying NNI,SPR and TBR respectively. It is not difficult to note that:

$$NNI(T) \subseteq SPR(T) \subseteq TBR(T) \tag{10}$$

Each of these three methods could be used to define distance of two phylogenetic trees. For example, the SPR distance of two trees is minimum number of SPR operations necessary to transform from one tree into another.

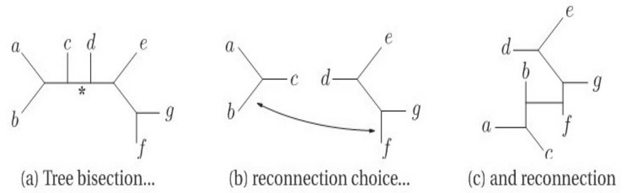
10.3 Bayesian methods

Bayesian inference employs MCMC to sample from the posterior probability distribution. Posterior probability distribution in phylogeny inference is the conditional probability of T given input dataset. Each step, Metropolis-Hastings algorithm is used here, each step we propose a modified new tree



(a) Phylogenetic tree (b) Subtree prune... (c) and regraft

Figure 3.15 Subtree prune and regraft (SPR). (a) An unrooted phylogenetic tree T. The subtree consisting of three edges and the two leaves labeled a and b is pruned at the location marked "$*$" to produce the two subtrees shown in (b), and then reattached into the edge leading to the leaf labeled f (c).



(a) Tree bisection... (b) reconnection choice... (c) and reconnection

Figure 3.16 Tree bisection and reconnection (TBR). The phylogenetic tree T shown in (a) is bisected into two subtrees by deleting the edge marked "$*$". As illustrated in (b), one possible choice is to reattach the two subtrees by a new edge joining the middles of the two leaf edges associated with taxa b and f. The resulting new phylogenetic tree is shown in (c).

topology, then accept it with the probability of $min(1, \alpha)$, α is the ratio of new prior times transition kernel divide the old one, otherwise, we reject and still use the current one. Therefore, a suitable modification of this proposed new tree should be taken into account with care. After discarding the burn in, sometimes we would also do sparsely sampling from output a series of trees, retaining only 1000th tree, say, in order to avoid the problem of autocorrelation, since the samples the not independent in this case. Let's talk about the modification in detail.

The first one is called *local algorithm*, which is a modification of NNI(Nearest neighbor interchange). This algorithm is very magic and powerful. Through a path in a tree, then update length of different nodes and even topology by introducing random number. Regarding parameters in evolutionary model, like mutation rate μ in Jukes Cantor model, are modified by adding a random number uniformly chosen from an appropriate interval centered at 0. For a set of parameters that is constrained to sum to some specific value, such as 1 in the case of probabilities, the values are randomly modified according to a Dirichlet distribution.

Modifying the tree and proposal for new parameters are generated independently and simultaneously, and then accept or reject by a single Metropolis-Hastings ratio.

The second one is called *Metropolis-coupled MCMC* or (MC^3) , this is a variant of MCMC which are suitable if there are a large amount of parameters. The first chain Z_1 is called cold chain, other chains $Z_2, ...Z_K$ are called heated chains, all chains are run parallel. After all chains have moved one step, Let T_i denote the current state of all chains, for all i =1 ,2 ... K, then after randomly choosing two chains Z_i and Z_j, swap the result with specific probability. Swapping of chains can help the cold chains to move a new part of of parameter space that it may have difficulty in reaching. At the end of run, discard all heated chains and cold chain is processed as before.

A challenge here is we may not know whether this chain has converged. Two approaches are introduced here: one is to monitor whether there there is no further increase in likelihood, another one is run multiple chains in parallel, to keep them running until appear to sample from the same distribution. In addition, areas like choosing prior distribution and check convergence is ongoing research.

10.4 Bootstrap analysis

In order to evaluate the robustness of estimated phylogenetic tree, bootstrapping is employed. Given a multiple sequence alignment M of length m and phylogenetic tree construction method, first step is to get bootstrap replicate M' by randomly sampling from M that consists of m columns, with replacement, normally a set of 100 or 1000, noted $\{M^1, M^2, ...M^{1000}\}$(say 1000), the method is applied in each replicate, producing a collection of phylogenetic trees $B = \{T^1, T^2, ...T^{1000}\}$, then use these trees to determine bootstrap support of each split. In practice, a bootstrap support of at least 70%, is required for a split to be considered trustworthy.

Above are the sequence-based methods for constructing a phylogenetic tree, however, when a set of sequences are available, distance-based method is considered a first, fast approximation, then more elaborate sequence-based methods are used to obtain a more trusted phylogeny. Usually a distance matrix D is obtained from given sequences in distance-based method, after that, a phylogenetic tree is constructed from that matrix D.

10.5 UPGMA and NJ

Neighbour-joining is the most popular method for computing the phylogenetic tree, viewed as a modification of UPGMA. In biology, molecular clock hypothesis states the mutation rate is constant over all sites of sequence and over all edges of the model tree. It implies that all leaves of tree all

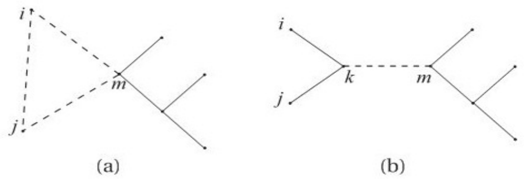


Figure 3.21 The dashed lines in (a) represent the distances between the three clusters C_i, C_j and C_m before pairing the nodes i and j. The dashed line in (b) represents the distance between the new cluster $C_k = C_i \cup C_j$ and C_m after pairing the nodes i and j. The neighbor-joining algorithm sets this distance to $d(C_k, C_m) = \frac{1}{2}(d(C_i, C_m) + d(C_j, C_m) - d(C_i, C_j))$.

have same distance from the root. Actually, any tree produced by UPGMA algorithm has the property that all leaves have the same distance to the root.

The steps of UPGMA are pretty easy to understand: given a set of sequences, first calculate a table showing the hamming distance. Then merge the smallest distance in pairs of sequences to one cluster, creating a new node. After that, calculate the new distance table again.UPDMA is assembled bottom-up.

Neighbour-joining doesn't need to assume a molecular clock like UPGMA. Its output is an unrooted tree. One difference between UPGMA and NJ is how to update the distance matrix after merging two clusters. The formula when computing the distance between a new cluster and one merged cluster is different(Look at figure 3.21). In UPGMA, the last term "$-d(C_i, C_j)$" is not needed.

Another significant difference between UPGMA and NJ is how the length of edges are set, formula in terms os computing the distance between a leave and merged node is in following figure 3.22. In summary, based on these formula, we could recalculate the length of each branch in model tree, the direct difference when looking at UPGMA and NJ trees is the giant different in each branch length.

However, we need to note that one disadvantage using NJ algorithm is that it may produce negative branch length, which happens quite frequently.

10.6 Balanced minimum evolution

A new and faster algorithm called FastME is proposed recently based on balanced minimum evolution(BME) framework, which could provide more accurate trees than NJ and never produce negative edge lengths.

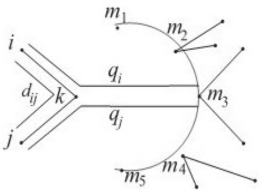
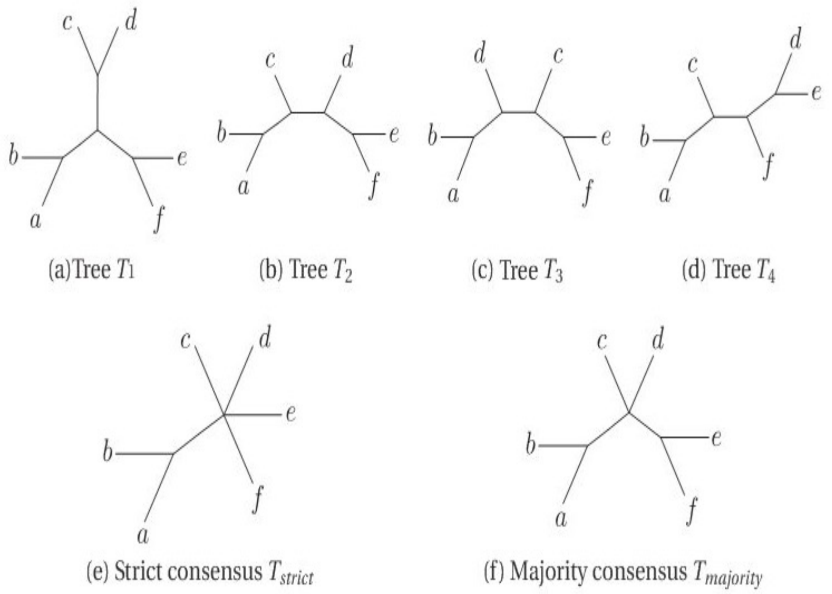


Figure 3.22 The neighbor-joining algorithm sets the length of the edge from i to k equal to $\frac{1}{2}$ of the average distance q_i from C_i to all current clusters C_m with $m \neq i, j$ plus the distance from C_i to C_j, minus the average distance q_j from C_j to all such clusters C_m.

Given a distance matrix and an unrooted bifurcating tree topology, we could calculate the leave edge and internal edge length based on clear formulas and total tree length is given as well, in other words, calculating BME tree is a statistically consistent tree construction method, however, finding the optimal BME tree is NP hard, so we turn to heuristics, more precise, FastME heuristic. It has two phases. First step is an initial tree is constructed , next step is to iteratively update the tree topology using NNI until no further improvement in the score of total tree length. Since it doesn't need to explicitly point out each branch length, that's why it is significantly faster than NJ in practice.

BME could be viewed as an improvement of classic minimum evolution problem. Moreover, it also consider the pairwise distance variance. In OLS method when constructing a tree, all distance have the same variance. But in general, it is not true, the variance of larger distance tend to be larger. Actually, based on the formula of total length tree length, longer distance implies a smaller weight, therefore, BME can also be interpreted as a weighted least square method. FastME runs much faster than all previous weighted least square approaches.

When comparing the distance between two phylogenetic trees, generally the trees are unrooted ones. Robinson-Foulds distance and quartet distance is used when measuring the similarity for a pair of trees. Even though there are so-called NNI, SPR and TBR distances which determine the minimal number of related operations from one topology to another, computation is NP hard and is rarely used in practice.



(a)Tree T_1 (b) Tree T_2 (c) Tree T_3 (d) Tree T_4

(e) Strict consensus T_{strict} (f) Majority consensus $T_{majority}$

Figure 3.26 For the collection of four phylogenetic trees $\mathcal{T} = (T_1, \ldots, T_4)$ shown in (a–d) we display the strict consensus tree T_{strict} in (e) and the majority consensus tree $T_{majority}$ in (f).

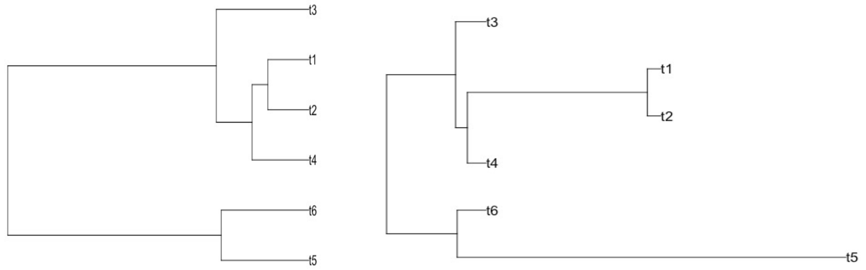
10.7 Consensus Tree

In general, when facing with a collection of different phylogenetic trees, producing from different methods given the same multiple sequence alignment like MLE , MP and so forth, or from Bayesian inference, *consensus method* is necessary used to obtain *consensus tree* that represents evolutionary history which different phylogenetic trees agree, in some sense.

Here we introduce two kinds of consensus trees: strict consensus and majority consensus. Let $S_{strict}(T)$ denotes the set of all splits that occur in every tree in tree set. $S_{majority}(T)$ denotes the set of splits that occur in more than half of all trees in tree set. Here is an example figure to explain the strict and majority tree. In practice, majority consensus tree is more informative than strict consensus tree since the latter tends to be a star tree easily.

Newick format is used to describe a rooted phylogenetic tree, of course it is also used to describe unrooted tree by erasing the root. example is here: text="((t4:0.104381,(t2:0.075411,t1:0.075411):0.5):0.065840,t3:0.170221);"

If possible, you could try this R code to plot a tree and check whether it is consistent with mine as shown beside the code:



```
library(ape)
tr1 <- read.tree(text= ”((t5:0.161175,t6:0.161175)
    :0.392293,((t4:0.104381,(t2:0.075411,t1:0.075411)
    :0.028969):0.065840,t3:0.170221):0.383247);”)
tr2 <- read.tree(text= ”((t5:2.161175,t6:0.161175)
    :0.392293,((t4:0.104381,(t2:0.075411,t1:0.075411)
    :1):0.065840,t3:0.170221):0.383247);”)
plot(tr1,font=1)
plot(tr2,font=1)
```

10.8 Phylogenetic network

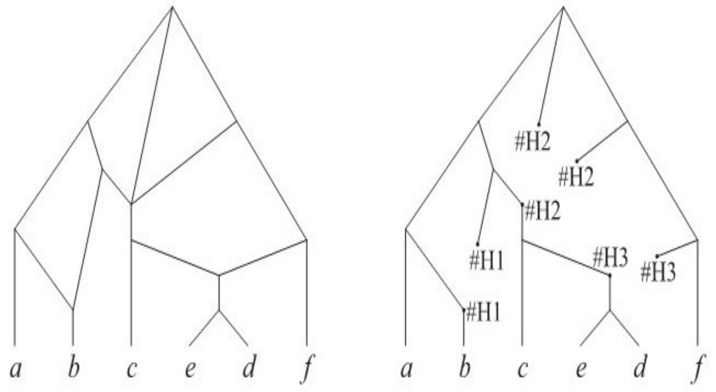
An overview of phylogenetic network is here

Phylogenetic networks can be computed from wide range of datasets, including multiple sequence alignments, splits, distance matrices, set of trees, clusters, rooted triplets or unrooted quartets.

Newick string could be extended to describe a rooted phylogenetic network. The method is to to assign formal labels Hi to reticulate nodes, details could be in the following Figure 4.7. Conversely, if we are given a this kind of Newick format to construct a network, first step is to construct a rooted phylogenetic tree, then merge the same formal taxon H.

Since my PhD project is related with recombination network. Let's introduce this type.

Recombination network is a rooted network that describe the evolution of a set of sequences(usually come from different individuals) in terms of mutation(along the edges of branch), speciation events (at tree nodes) and recombination events (at reticulate nodes).



(a) Rooted network N (b) Rooted tree T

```
(((a,(b)#H1),(#H1,((c,((d,e))#H3))#H2)),#H2,(#H2,(#H3,f)));
```

(c) Newick description

Figure 4.7 (a) A rooted phylogenetic network N on $\mathcal{X} = \{a, \ldots, f\}$. (b) A multi-labeled phylogenetic tree T on $\mathcal{X}' = \{a, \ldots, f, \#\mathrm{H1}, \#\mathrm{H2}, \#\mathrm{H3}\}$ that can be used to determine the Newick description of N. (c) The resulting Newick string.

Here is a formal definition of recombination network:

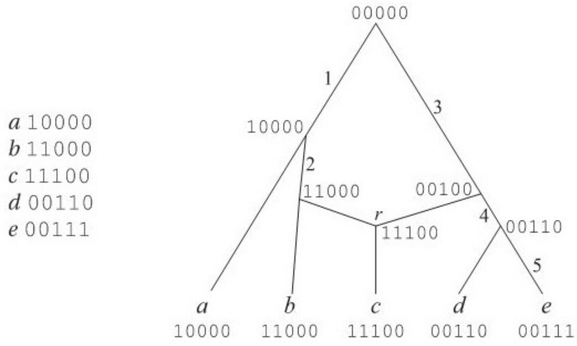
Let M be a multiple alignment of binary sequences of length L, on χ. A recombination network N representing M is given by a combining of rooted phylogenetic network on χ, together with two additional labelings:

(i)each node v of N is labeled by a binary sequence $\delta(v)$ of length L.

(ii)each tree edge e is labeled by a set of positions $\delta(e) \subseteq \{1, 2, ...L\}$.

These two labelings must fulfill the following compatibility conditions:

1. These sequence $\delta(v)$ assigned to any leaf v must equal the sequence in M that is given for the taxon associated with v.
2. If r is a reticulate node(often called recombination node) with parents v and w, then the sequence $\delta(r)$ must be *obtainable* from $\delta(v)$ and $\delta(w)$ by a crossover.
3. If $e = (v, w)$ is a tree edge, then the set of positions at which two sequences $\delta(v)$ and $\delta(w)$ differ must equal $\delta(e)$.



(a) Alignment M (b) Recombination network N

Figure 9.15 (a) A multiple alignment of binary sequences of length five on $\mathcal{X} = \{a, \ldots, e\}$. (b) A recombination network N for M. For each node v we show the sequence $\sigma(v)$ and for each edge e we show the set of positions $\delta(e)$. In this example, that set is either empty (unlabeled edges) or contains only one position (from 1 to 5).

For computational reasons, the following condition is usually also required:

4. Any given position may mutate at most once in the network. In other words, for any given position i, there exist at most one edge e with $i \in \delta(e)$

Here we can verify this simple recombination network example, can the node r be obtained by a single-crossover recombination from parental sequences?

Evolution in the presence of recombination is usually studied in population genetics, rather than in phylogeny, focusing a statistical method , under the "coalescent with recombination" model, a description of history from n sampled sequences give rise to a graph called ancestral recombination graph(ARG).

In fact, evolutionary history of any sufficiently short segment of sequence is a rooted phylogenetic tree. Therefore, one natural and direct way is to construct a suitable rooted phylogenetic tree T_i for each position i in the alignment M, we call any such tree T_i as **local tree**, then combine all trees into a suitable phylogenetic network N.

Currently, there are two ways to construct phylogenetic network by these local trees:

1. **The local-tree parsimony approach**
2. **A heuristic for the local tree approach**

First in the local-tree parsimony approach, in the local-tree graph G any path P is defined as :

$$P = (v(1,T_1), v(2,T_2), ...v(L,T_L)) \tag{11}$$

The total weight of such path is the summation of node weight and edge weight. node weight means the minimum number of substitutions required for the i-th character on the rooted phylogenetic tree, edge weight is the minimum number of reticulate nodes which links tree T and T', reflecting the recombination distance. Any two nodes $v(i,T)$ and $v(i+1,T')$ at adjacent positions in the alignment are corrected by a directed edge $e = (v(i,T), v(i+1,T'))$.

A *most parsimony* set of trees for M is given by a path P of minimum weight, then recursion method could be used, see below:

$$W(i,T) = \begin{cases} min_{T'} W(i-1,T') + w(v(i-1,T'), v(i,T)) + w(v(i,T)) & \text{if i >1} \\ w(v(1,T)) & \text{else} \end{cases} \tag{12}$$

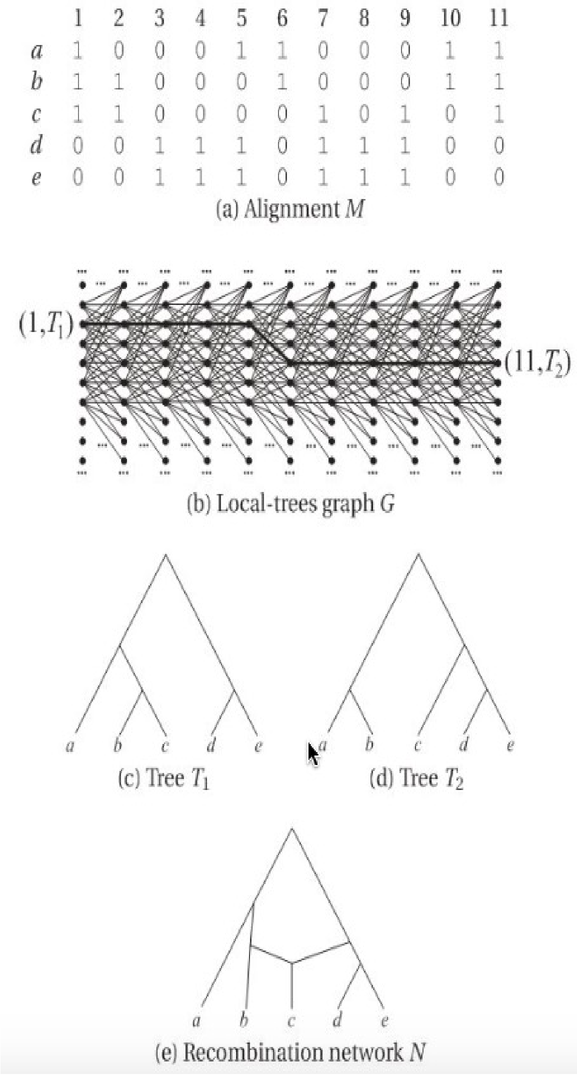
An example is shown in the figure. Unfortunately, this method is not practical, since computing a suitable phylogenetic network representing all rooted trees in second part is an NP-hard problem, besides that, computation of edges weight is also NP-hard.

Next is the heuristic for the local tree approach.The first simplification is to use unrooted phylogenetic trees rather than rooted trees, main simplification is to consider a small part of full local tree graph.

The heuristic starts by computing a phylogenetic tree for M using maximum parsimony method. Then we compute the SPR-neighborhood S of T that only one SPR modification of T, the main recursion is the same as previous section. The resulting set of trees might depend quite strongly on the initial tree, so once completing the first pass of this algorithm, a second pass is needed, start tree will be replaced by the tree from last position. However, this step is still challenging in practice.

Actually, in paper"Computing recombination networks from binary sequences" Daniel H.Huson 2005, researchers proposed a method to draw recombination network from aligned binary sequences. There are some important and meaningful concepts before the description of core algorithm. Let's first introduce these concepts.

Splits can be compatible, circular, and weakly compatible. The last one is of interest because these splits can be efficiently computed using split



	1	2	3	4	5	6	7	8	9	10	11
a	1	0	0	0	1	1	0	0	0	1	1
b	1	1	0	0	0	1	0	0	0	1	1
c	1	1	0	0	0	0	1	0	1	0	1
d	0	0	1	1	1	0	1	1	1	0	0
e	0	0	1	1	1	0	1	1	1	0	0

(a) Alignment M

(b) Local-trees graph G

(c) Tree T_1

(d) Tree T_2

(e) Recombination network N

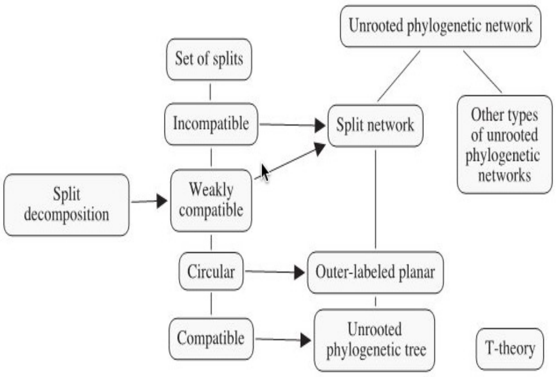
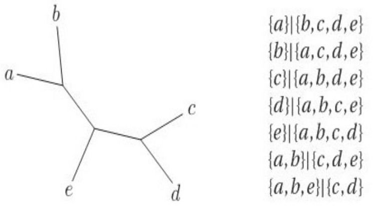


Figure 5.1 Overview of the main concepts introduced in this chapter. On the left, we list the different properties that a set of splits can have, in order of decreasing generality, and in the middle, we list the corresponding types of split networks.

decomposition algorithm.



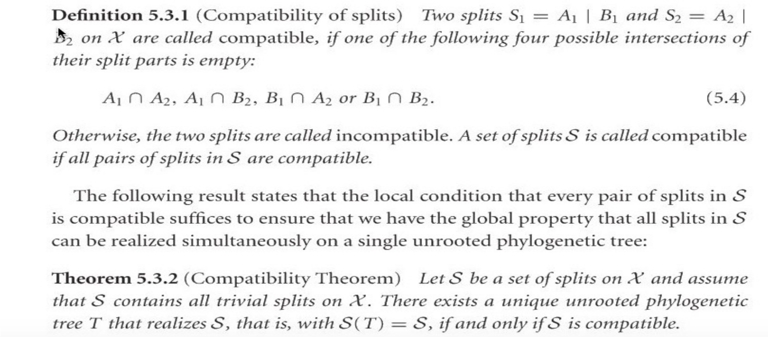
$\{a\}|\{b,c,d,e\}$
$\{b\}|\{a,c,d,e\}$
$\{c\}|\{a,b,d,e\}$
$\{d\}|\{a,b,c,e\}$
$\{e\}|\{a,b,c,d\}$
$\{a,b\}|\{c,d,e\}$
$\{a,b,e\}|\{c,d\}$

(a) Unrootedtree T (b) Split encoding of T

Figure 5.2 (a) An unrooted phylogenetic tree T on $\mathcal{X} = \{a, \ldots, e\}$. (b) The seven splits represented by T.

Here the term compatible, incompatible and split network all occur in above mentioned paper.

Given the following unrooted phylogenetic tree, there are seven edges in this tree, it gives rise to seven split, called split encoding of T. Suppose we are given an arbitrary set of splits S, we would like to know whether S can be represented by some unrooted phylogenetic tree T with $S = S(T)$. The answer is given by compatibility theorem.



Definition 5.3.1 (Compatibility of splits) *Two splits $S_1 = A_1 \mid B_1$ and $S_2 = A_2 \mid B_2$ on $\mathcal{X}$ are called* compatible, *if one of the following four possible intersections of their split parts is empty:*

$$A_1 \cap A_2,\ A_1 \cap B_2,\ B_1 \cap A_2 \text{ or } B_1 \cap B_2. \tag{5.4}$$

Otherwise, the two splits are called incompatible. *A set of splits $\mathcal{S}$ is called* compatible *if all pairs of splits in $\mathcal{S}$ are compatible.*

The following result states that the local condition that every pair of splits in $\mathcal{S}$ is compatible suffices to ensure that we have the global property that all splits in $\mathcal{S}$ can be realized simultaneously on a single unrooted phylogenetic tree:

Theorem 5.3.2 (Compatibility Theorem) *Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$ and assume that $\mathcal{S}$ contains all trivial splits on $\mathcal{X}$. There exists a unique unrooted phylogenetic tree T that realizes $\mathcal{S}$, that is, with $\mathcal{S}(T) = \mathcal{S}$, if and only if $\mathcal{S}$ is compatible.*

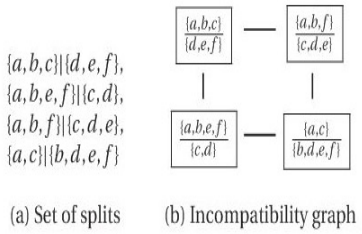


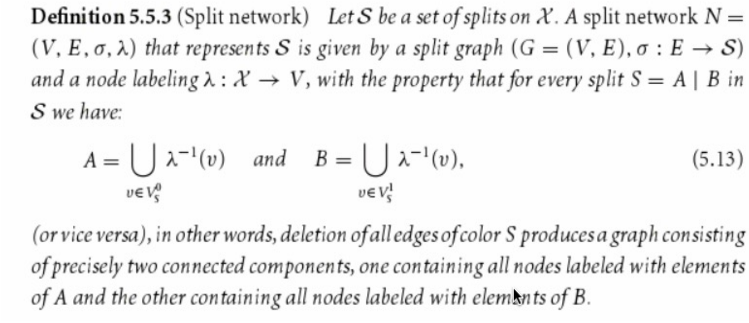
Figure 5.3 (a) A set of four splits $\mathcal{S}$ on $\mathcal{X} = \{a, \ldots, f\}$. (b) The corresponding incompatibility graph $IG(\mathcal{S})$, which has four nodes and four edges.

It is also useful to represent the incompatibilities among a set of splits S by a graph called **Incompatibility graph**, the formal definition is here:

Incompatibility graph: The Incompatibility graph IG(S) of a set of splits S is the graph (V, E) that has node $V = S$ and edge set $E = \{(s_1, s_2)|s_1 and s_2 are compatible\}$, one example is shown in figure 5.3.

From the definition of incompatibility graph, we could know that if one split in S is compatible with other splits in S, then this split will be isolated node in incompatibility graph.The formal definition of split network is shown below.

If we want to compare two split network N_1 and N_2, we compare the



Definition 5.5.3 (Split network) *Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. A* split network *$N = (V, E, \sigma, \lambda)$ that represents $\mathcal{S}$ is given by a split graph $(G = (V, E), \sigma : E \to \mathcal{S})$ and a node labeling $\lambda : \mathcal{X} \to V$, with the property that for every split $S = A \mid B$ in $\mathcal{S}$ we have:*

$$A = \bigcup_{v \in V_S^0} \lambda^{-1}(v) \quad \text{and} \quad B = \bigcup_{v \in V_S^1} \lambda^{-1}(v), \tag{5.13}$$

(or vice versa), in other words, deletion of all edges of color S produces a graph consisting of precisely two connected components, one containing all nodes labeled with elements of A and the other containing all nodes labeled with elements of B.

corresponding set of splits S_1 and S_2, using the Robinson-Foulds distance:

$$d_{RF}(S_1, S_2) = \frac{|S_1 \Delta S_2|}{2} \tag{13}$$

Robinson-Foulds distance is also a proper metric for comparing tree distance.

Actually, there could be a tree or other graph to represent a same set of splits. There is also an important lemma: a set of splits are compatible if and only if there exists a split network N representing S that is a tree. Hence, given our input is a set of splits, how to construct a split network N as output is our first challenge. Below are two available approaches.

The first is the *convex hull algorithm* that computes the Buneman graph and can be applied to any set of splits, using an exponential number of nodes and edges in the worst case; the second is the *circular network algorithm* , which can be applied to any set of **circular** splits and produces an outer-labeled planar network with only a quadratic number of nodes and edges.

Both algorithms proceed in two steps. In the first step, all trivial splits in $S = \{S_1, ..., S_m\}$ are processed to obtain a star network consisting of a central node and one leaf per taxon, noted S^O (outer). Then, in the second step, the remaining splits are inserted one by one , noted S^I (inner) so as to obtain the final network.

11 RDP4 note

RDP4 is the latest version of recombination detection program in a set of aligned sequences. RDP4 could not do multiple sequence alignment, but from its instruction manual, it provides three reliable sequence alignment tools: ClustalX/W, MUSCLE, POA for small, medium and large datasets respectively(small means fewer than 100 sequences, large means more than 100 sequences).

11.1 Compile a good dataset

Firstly, we need to compile a good dataset. Although there is no formula to tell us the optimal numbers and lengths of sequences for optimal recombination detection, some procedures are needed to ensure a reasonable good dataset.

11.2 Make a good alignment

After getting a suitable datasets, next step is to make a good alignment, which is essential for recombination analysis. It is not recommended that any pair of sequences share less than 60% nucleotide sequence identity, ideally this value is greater than 70%. Multiple sequence alignment tools will occasionally make some alignment errors, that's why we need to realign subsections of alignment to rectify these errors.

In general, after making a preliminary alignment of the sequences, if these are small sequences, we could use an alignment editor such as MEGA or IMPALE to check the accuracy of completed alignment by eyes, if they are large datasets, using the sub-sequence realignment tool in MEGA or IMPALE with different alignment parameter settings. It is strongly recommended that any unalignable (or just barely alignable) tracts be either deleted from the alignment or shifted/staggered.

11.3 Prelimnary scan

We can click "open" and load alignment files, RDP4 could recognize these different file formats: FASTA, PHYLIP, GDE, CLUSTAL, GCG,NEXUS,MEGA, DNAMAN, .pdb.

We need to set up some parameters before general scanning. First one is to specify whether the sequence being examined is linear or circular. The following picture is an example of circular sequence.

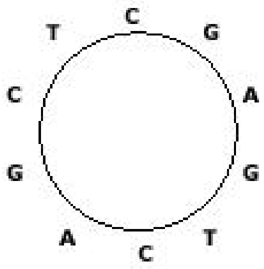


Figure 15: This graphic is originally from here

We always read a circular sequence in the clockwise direction, then it should be “CGAGTCAGCT” in the above example. Of course there can be many linear sequences that are obtained from such a circular sequence, by cutting any place of the circular sequence.

Move to “Analysis Sequences Using” in “general ” button, it is strongly recommended to use the default setting. Pay attention to the Botscan and Siscann, there are two boxes in front of it separately. By default these two will automatically check the recombination signals detected by other methods,if we click the first box, then it will force their use to explore new signals, but be warned that it will increase analysis time dramatically. By the way, LARD method is also only used for checking signals and suitable for less than pretty small dataset (less than 20 sequences).

After general settings, click “Run” button in the command button panel.

11.4 Refine Preliminary recombination

It is important to be aware that RDP4 can get things horribly wrong, such as inaccurate identification of breakpoint positions. Unfortunately, there is no automated tools to judge whether our results are true.

In order to check the accuracy of estimation breakpoints estimation, best graphs are shown in the bottom left panel by choosing “Check using ” “MAXCHI”and “CHIMERA”. Breakpoints are displayed in the peak of these two curves. However, if the peaks doesn’t match the border of recombinant, this doesn’t mean this inferred positions are wrong, it does mean there is a degree of uncertainty regarding this position. By the way, considering “Confirmation Table” in Recombination Info at top right panel is also a good way to assess the accuracy of estimation.

Remember this program could crash at any time so we should regularly save our results.

References

[1] Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications.* Cambridge University Press, 2010.

[2] Daniel H Huson and Celine Scornavacca. “A survey of combinatorial methods for phylogenetic networks”. In: *Genome biology and evolution* 3 (2011), pp. 23–35.

[3] Na Li and Matthew Stephens. “Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data”. In: *Genetics* 165.4 (2003), pp. 2213–2233.