Journal of Immunological Methods 338 (2008) 67–74

Contents lists available at ScienceDirect

Journal of Immunological Methods

journal homepage: www.elsevier.com/locate/jim

Computational modeling

IgTree©: Creating Immunoglobulin variable region gene lineage trees

Michal Barak, Neta S. Zuckerman, Hanna Edelman, Ron Unger, Ramit Mehr *

The Mina and Everard Goodman Faculty of Life Sciences, Bar-Ilan University, Ramat-Gan 52900, Israel

ARTICLE INFO

Article history:
Received 23 April 2008
Received in revised form 21 June 2008
Accepted 23 June 2008
Available online 14 August 2008

Keywords:
B lymphocyte
Immunoglobulin variable region gene
Lineage tree
Phylogenetic tree
Germinal center

ABSTRACT

Lineage trees describe the microevolution of cells within an organism. They have been useful in the study of B cell affinity maturation, which is based on somatic hypermutation of immunoglobulin genes in germinal centers and selection of the resulting mutants. Our aim was to create and implement an algorithm that can generate lineage trees from immunoglobulin variable region gene sequences. The IgTree© program implements the algorithm we developed, and generates lineage trees. Original sequences found in experiments are assigned to either leaves or internal nodes of the tree. Each tree node represents a single mutation separating the sequences. The mutations that separate the sequences from each other can be point mutations, deletions or insertions. The program can deal with gaps and find potential reversion mutations. The program also enumerates mutation frequencies and sequence motifs around each mutation, on a per-tree basis. The algorithm has proven useful in several studies of immunoglobulin variable region gene mutations.

1. Introduction

The adaptive immune system has the remarkable ability of recognizing and responding to new pathogens. B lymphocytes are part of this system. The B cell receptor (BCR), also known as immunoglobulin (Ig), is pivotal in the recognition process: it has a variable part whose gene is rearranged from many segments during B cell development, creating a repertoire of B cells whose receptors can recognize all possible antigens (foreign or self molecules). When B cells encounter an antigen to which their BCRs bind, and become activated, most of them differentiate into plasma cells that secrete their Igs (also known as antibodies), while some B cells undergo the following evolution process. Proliferating B cells and other immune system cells form structures called germinal centers (GCs), in which the variable regions of the expressed Ig genes are subjected to rapid mutation. B cells are subsequently selected depending on their receptors' affinity to the antigen. Most mutations reduce the BCR's affinity to the antigen, or render the gene or the protein nonfunctional. However, the few B cells that improve their BCR affinity to the antigen, successfully pass selection and differentiate into Ig-secreting plasma cells. Memory B cells are also created from the successful mutant B cells, so upon second infection by the same pathogen, the immune response is faster and more efficient.

This affinity maturation of Igs can be explored by examining the Ig gene sequences of the responding B cells (Kochs and Rajewsky, 1988; Manser, 1989; Jacob et al., 1991; Jacob and Kelsoe, 1992; Shannon and Mehr, 1999). The original sequence ("root") can easily be found by sequence alignment with the known germline segments in the genome, thus identifying clonal relationships among Ig gene mutants. The process of affinity maturation is stochastic, such that cells at different stages can exist simultaneously. Lineage trees, constructed from experimental sequence data obtained from several cells of the same clone, trace the most probable evolution from the known root to the experimentally identified sequences (which may include both leaves and internal nodes in the tree, as diversification is still ongoing at the time of sampling). Note that the procedure does not intend to re-construct the particular sequence of events that actually took place. Rather, the procedure aims to find the minimal sequence of events that could have led to the

Abbreviations: BCR, B cell receptor; GCs, germinal centers; Ig, immunoglobulin; AL, light chain amyloidosis.

* Corresponding author. Tel.: +972 3 531 7990; fax: +972 3 736 2560. *E-mail address:* prof.ramit.mehr@gmail.com (R. Mehr).

0022-1759/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jim.2008.06.006

observed sequences assuming that the minimal sequence of events is the most probable natural scenario. Such trees can be used to explore the dynamical properties of Ig affinity maturation. By exploring the characteristics of these trees, we can gain insights into the diversification and selection processes that take place in the GC (Dunn-Walters et al., 2002). Using graph-theoretical measures of lineage trees, we found correlations between some of these measures and dynamic parameters of the GC response that generated the tree. Properties such as degree of branching (measured, e.g. by the average number of descendants per node) can point to the strength of selection, the mutation rate and initial affinity to the antigen (Shahaf et al., in press). B cell lineage trees, constructed manually by counting mutations, have provided insights into these processes, for example, Ig gene micro-evolution after the first immune response (Dunn-Walters et al.,. 2004). By analyzing tree properties, selection in the GC can be quantified (Banerjee et al., 2002; Dunn-Walters et al., 2003), and the diversification of Ig genes in various pathological conditions, such as chronic inflammatory diseases, autoimmune diseases or B cell malignancies, can be elucidated (Steiman-Shimony et al., 2006a,b;Manske et al., 2006; Abraham et al., 2007; Tabibian-Keissar, in press).

The computational process in which lineage trees are constructed is based on the principle of creating phylogenetic trees, which describe the evolution of related nucleic acid sequences or protein sequences in various species. The sequences are the outer leaves of the tree, while the inner structure of the tree reflects relationships between the sequences. Unlike phylogenetic trees, however, the root of the lineage tree is known in Ig gene sequences. In addition, the sequences represented in the tree are not necessarily leaves, as intermediate sequences may be included in the sample (Fig. 1). An Ig lineage tree is not necessarily a binary tree, because a population of cells with the same Ig gene sequence may generate many different mutations. Hence, the existing algorithms, and software implementing them, are not directly applicable for the construction and analysis of Ig gene lineage trees, and using these algorithms requires considerable manual correction.

NP-Complete is a technical term in Computer Science denoting a set of problems for which it is strongly believed that no efficient (i.e. that have polynomial run-time) algorithm that guarantees optimal solution for all instances can be found. It is important to establish if a specific problem belongs to this set because then the emphasis should be shifted to finding practical heuristic algorithms that can supply good solutions for most cases. Phylogenetic tree reconstruction is a known NP-complete problem (Day et al., 1986; Chor et al., 2005). Creating rooted phylogenetic tree with a known root is also NP-complete, since to find the minimal un-rooted tree we need only to create a tree for every sequence as the root, and from the created trees choose the minimal one. This will multiply the complexity of the rooted tree creation algorithm by N, and since multiplication by N does not reduce a problem from the NP-complete class, the rooted tree creation problem is also NP-complete.

The creation of the trees with internal nodes that can come from the input sequences also does not remove the problem from the NP-complete class. The tree can be converted to a rooted phylogenetic tree in a few steps. The first step is adding an additional node for every input sequence that is an internal node of the tree. The new node will be a leaf descending from the node, with a distance of zero. Both nodes will have the same sequence, but only the leaf will be marked as an input value. The next stage is to remove all the internal nodes that have only one descendant, connecting for every node their descendant with their parent and increasing the distance between the parent and the descendant by one. The resulting tree is the rooted phylogenetic tree with distances. The conversion between the trees can be done in linear time. If the initial tree is minimal, then the converted tree is also a minimal tree, since the distances in the tree – which represent the number of mutations – remain the same. If the creation of the initial tree is not NP-complete then so is the creation phylogenetic tree. Since this is not so, the creation of lineage trees with the internal nodes is still an NP-complete problem.

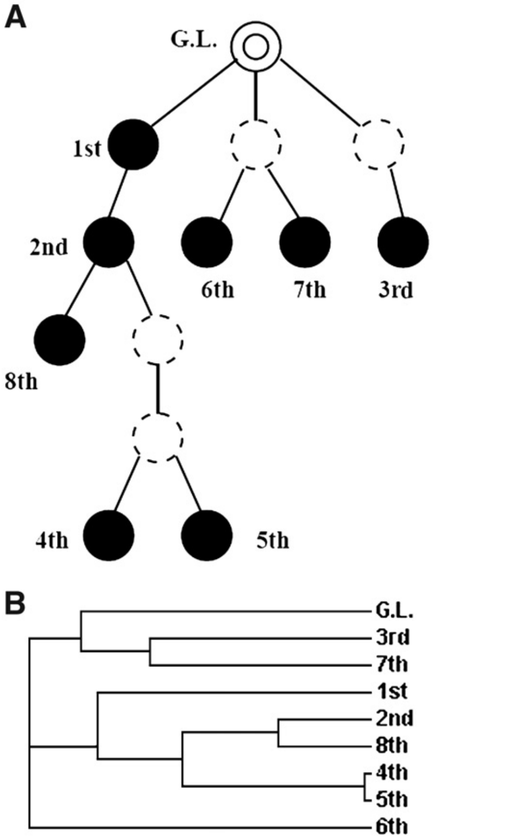

Fig. 1. Differences between phylogenetic trees and Ig lineage trees. (A) A lineage tree is composed of all internal nodes, where every node represents a single mutation. The root of the tree (double circle) is known, and sampled data (filled nodes) may represent internal nodes in the tree. Deduced split and pass-through nodes (dashed circles) are added to the tree if needed. (B) A phylogenetic tree constructed by ClustalW (Larkin et al., 2007; www.ebi.ac.uk/Tools/clostalw) from the same alignment as the Ig lineage tree in A, using the default parameters of ClustalW. In such trees, the root may be known or unknown, sampled data can only be represented as leaves in the tree, and internal nodes are not calculated nor shown on the tree. In the lineage tree, the 6th and 7th sequences are closer than in the phylogenetic tree, as an intermediate mutation was added, causing the 7th sequence to be closer to the new node than to the 3rd sequence. Thus, using phylogenetic tree algorithms such as ClustalW for constructing Ig gene lineage trees requires much manual correction to get from A to B.

Thus, this study presents a heuristic algorithm that was tailored to handle the construction of Ig gene lineage trees, where all the mutations are represented as individual nodes, and observed sequences are not necessarily leaves. The program can deal with sequence gaps, and also finds potential reversion mutations. Additionally, the tree does not necessarily have to be binary, and the output tree is formatted as an adjacency list, ready to serve as input for our program MTree©, which measures the characteristics of trees (Dunn-Walters et al., 2002, 2004). The created lineage tree is also used for mutation analysis. This analysis is done for all the mutations in the tree and not only with the experimental sequences. Lineage tree generation makes it possible to deduce information about the most likely order of groups of mutations — e.g., mutations that appear in the tree's trunk have most likely occurred before those occurring in any branch, and so on. This way, the search for sequence motifs is more accurate, as each mutated sequence is compared to its most probable parent, rather than to the germline sequence.

2. Methods

2.1. Input: sequence alignments

Ig gene sequences sampled from B cells in tissues such as GCs or peripheral blood are supplied by our collaborators or collected from the immunological literature (Steiman-Shimony et al., 2006a,b; Manske et al., 2006). The sequences to be analyzed are aligned with sequences from rearranged germline Ig genes to find the original sequence of the Ig using IgBLAST (www.ncbi.nlm.nih.gov/igblast), VBASE (vbase.mrc-cpe.cam.ac.uk) or SoDA (dulci.org/soda). We have tried all three germline gene alignment tools, and found the following. First, SoDA is currently the most useful, as it gives the whole putative germline sequence, including the whole CDR3 sequence (it simply inserts the N-nucleotides in their appropriate places after identifying the V, D and J segments). Second, however, SoDA gives only one germline segment per sequence, and sometimes because of small differences (usually in the D segment) it splits a group of sequences that clearly came from the same clone into two or more. Hence, we checked the choices made by SoDA against those give by IgBlast or VBASE. Then, the sequences belonging to the same clone are aligned using ClustalW (www.ebi.ac.uk/Tools/clostalw). Our algorithm uses the output of the latter alignment to construct the tree from the aligned sequences, including mutations from all locations in the variable region sequence. Since sequence length must be the same for all aligned sequences, sequences may be cut to match shortest one.

2.2. Algorithm

Trees that genuinely represent Ig gene evolution are not necessarily binary, include all internal nodes, and allow representation of sampled data as internal nodes (Fig. 1). The algorithm described in this paper complies with all the requirements for Ig gene lineage trees. In addition, the algorithm can handle a succession of mutations (gaps or adjacent point mutations) as one mutational event, thus enabling it to also handle gene conversion events. The goal of the algorithm is to create a lineage tree with the minimal number of mutations (nodes in the tree), where every node is separated from its immediate ancestor by only one mutation.

The tree-constructing algorithm performs the following set of steps:

1) Read the data, including the root sequence, and marking duplicate sequences — sequences that were derived from different B-cells but are identical.
2) Calculate the distance between each pair of the distinct input sequences and find possible ancestor–progeny relationships.
3) Construct the preliminary tree.
4) Add internal nodes to the tree to represent all individual mutations, thus creating the full tree.
5) Check whether inclusion of reversions can improve the tree.
6) Change the internal nodes to follow the changes suggested in step 5.
7) Repeat the reversion correction until the first of the following two conditions is met:
    a. No improvement in the tree can be gained by adding more reversions.
    b. An upper limit to the number of reversion cycles (typically 5) was reached.
8) Output the tree
9) Perform mutation analyses and output the results.

2.3. Reading the data

The input of the algorithm is a NBRF/PIR file (one of the standard output formats of ClustalW), containing the results of sequence alignment. The aligned sequences all have the same length. The GL gene, which is one of the sequences in the input, is marked as the root of the tree in the input. If there are duplicate sequences with different names, only one of them is used in the construction of the tree.

2.4. Calculating inter-sequence distances and finding possible ancestor relationships

Our algorithm is a modification of the distance method concept (Li, 1981; Saitou and Nei, 1987), based on a heuristic approach for tree construction. The distance between two sequences is the edit distance, representing the minimal number of mutations separating the two sequences. In instances of more than one possible mutation at the same location, for example, choosing between two consecutive deletions with length of one or one deletion of two bases, the longer mutation is used and the other mutations are discarded such that we use the minimal number of distinct mutational events — and hence the edit distance. The number of mutations used after the elimination of overlapping mutations is defined as the distance. The mutations between the sequences are compared against the root. If one of the sequences includes all the mutations separating another sequence from the root, it is marked as farther from the root and a possible descendant of that other sequence.

2.5. Constructing the preliminary tree

The preliminary tree (Fig. 2A) is constructed using the aligned sequences. The distance is calculated for every pair of

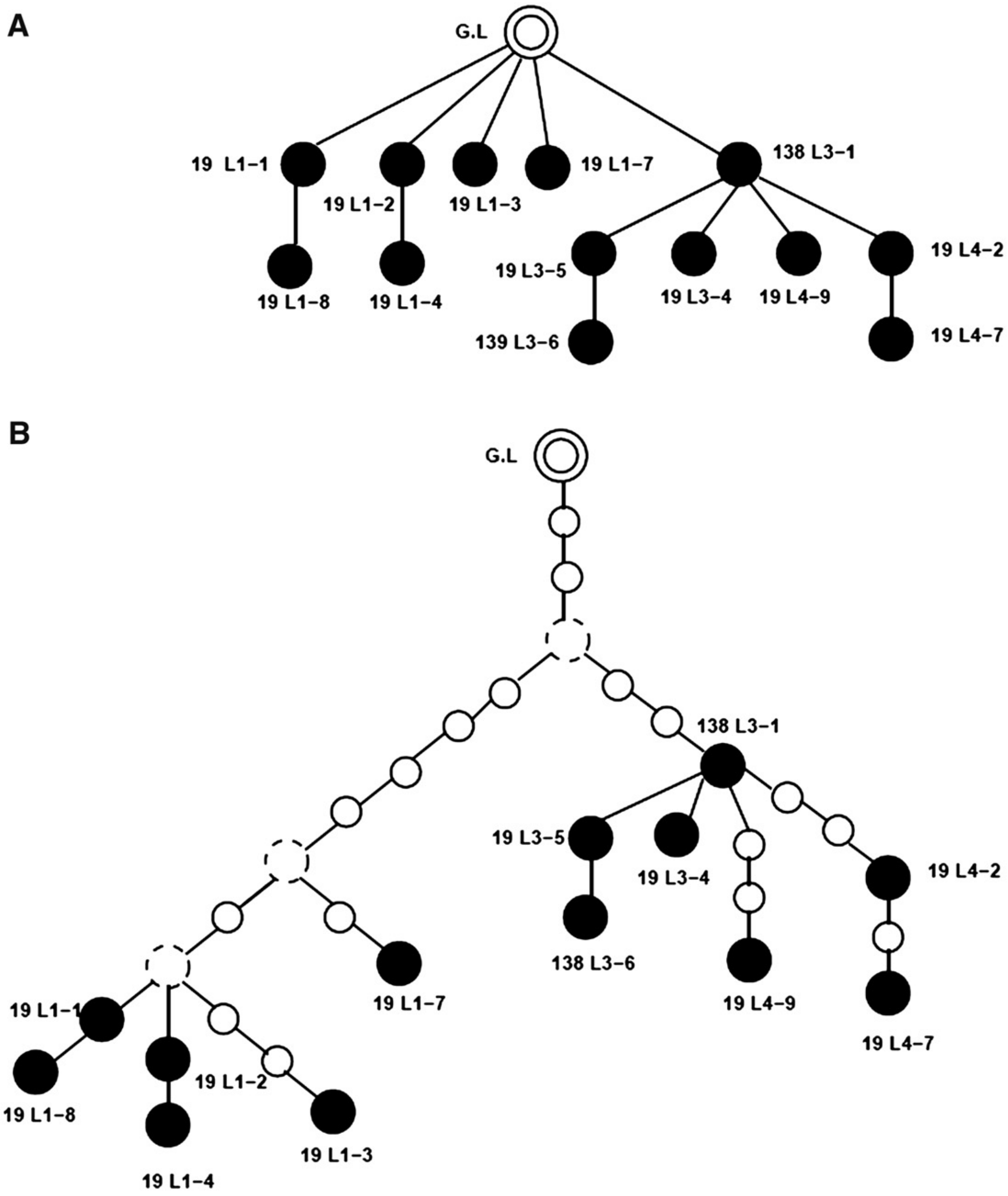

Fig. 2. Lineage tree constructed by IgTree from Ig gene sequences taken from a light chain amyloidosis patient (Manske et al., 2006). The root of the tree is the GL sequence, marked by a double circle. Filled circles represent sampled sequences (from the alignment). Dashed circles represent split nodes, i.e., nodes that have more than one descendant, added by IgTree©. Solid white cycles represent pass-through nodes that have only one descendant representing one mutation, added by IgTree©. The name of the sequences appears next to the sampled nodes. (A) The initial tree constructed by IgTree©. (B) The final tree, after split nodes and pass-through nodes are inserted.

two sequences, including the root. Every sequence becomes a node in the tree. For every sequence, excluding the root, the initial parent is set to be the root. Potential parental relationships are examined in all the nodes excluding the root, which does not have a parent. The actual parent of a node is found by reviewing all the sequences that were marked as possible parents of the node's sequence. If there are no possible ancestors, then the root is set as the parent of the node. If there are possible ancestors nearer to the root from the reviewed node, the one with the shortest distance from the node is chosen as its parent. (Fig. 2B). Thus, putative intermediate split nodes (nodes with more than one descendant) may be created. When this procedure is completed for all nodes, the preliminary tree has been constructed.

2.6. Constructing the full tree

After the preliminary tree has been constructed, the next stage is to fill all the missing internal nodes in the tree. The goal is to construct a tree in which every node is separated from its immediate ancestor by no more than one mutation, therefore all nodes, including the new nodes that were added in the process, are analyzed. If the node has no descendants, it is set as a leaf in the tree. If a node has descendants separated by only one mutation from it, then it meets the required conditions to be represented as an internal node in the tree. On the other hand, if the node has at least one descendant separated from it by more than one mutation, intermediate nodes are added to the tree, as described below, until the one mutation separation rule is enforced. When all the nodes comply with this rule, the full tree is completed. It is important to note that in Ig lineage trees, all mutations have equal weights — no mutation is considered to be more important than any other.

To create intermediate nodes, a new sequence has to be created for each intermediate node. Only one mutation is permitted to separate the parent from its new descendant. To find this mutation, a list of all the mutations separating the parent from its existing descendants is created. A score is calculated for each mutation, as described below. The sequence of the new node is created from its parent's sequence, by applying to the latter the mutation with the highest score. The new node is added as a descendant of the parent node. The descendants of the parent that have the mutation are moved in the tree to become the descendants of the new node (Fig. 2). It is important to note that we do not make any claims regarding the order of mutations between two adjacent defined points (root, split, leaf or an internal node corresponding to an experimentally observed sequence). We assign them to nodes in random order, as from the point of tree measurements only the number, and not the identity, of the mutations between two adjacent defined points matters.

The mutation list is created by collecting all the descendants' mutations. Each mutation in the mutation list is given a modular score, composed of the following elements: (1) the number of descendants sharing the mutation, (2) the number of base substitutions/insertions/deletions in the mutation, (3) the number of mutations shared by descendants that have this mutation, (4) the Root of the Mean of the Squared deviations (RMS) of the number of mutations needed to create a lineage tree from the root and the pair of descendants for all descendant pairs that have this mutation, (5) the average distance between pairs of descendants, that have this mutation and (6) The average distance from the root for all descendants that have this mutation. Each component of the score has a different priority according to the order of the calculation, such that each component of the score acts as a tie breaker when the previous score components are equal. For example, a mutation with a higher number of descendants will always have a higher score than a mutation with fewer descendants but more base changes. From this list, mutations in the same location in the sequences are eliminated save one, since only one mutation per location can separate adjoining nodes. This is done by choosing only the mutation with the highest score to be included in the mutation list.

2.7. Identifying reversions

Reversions are mutations that replace an already-mutated nucleotide by the original pre-mutation nucleotide. Although the algorithm attempts to create the minimal tree that includes all the mutations, reversions can be found in Ig lineage trees, just as they can occur in real B cell clones. Reversions cannot be identified in a straightforward way, since it is unknown whether any given position had a reversion mutation or did not have a mutation at all. However, some reversions can be deduced from lineage trees. Because the tree construction algorithm first deals with mutations that have received a high score, it may assign a sequence to a separate branch because it is lacking one of those first mutations, even when it shares a – potentially large – number of other mutations with an existing branch. Such cases are identified and resolved as follows (Fig. 3).

A “skeleton node” is defined as a node that has more than one immediate descendant, or a sequence that was included among the experimentally sampled sequences. Every node representing a sequence from the original dataset is tested for reversions against all other nodes in the tree. In addition, the distance between the tested nodes and their closest skeleton ancestor is also measured. If the distance between the tested node and another node (referred to as the “found” node) is shorter than the distance between the tested node and its skeleton ancestor, a reversion is suspected. The common ancestor of the found and tested nodes is then retrieved. If the distance between the common ancestor and the tested node is bigger than the distance between the tested node and the found node, the reversion did occur. The new parent of the tested node is then set to be the skeleton ancestor of the found node (if the found node is a skeleton node, then it is set to be the new parent of the tested node). In case there are several found nodes, i.e., they have the same minimal distance from the tested node, then the found node that is chosen is the one that belongs to the original dataset. If none or all of the nodes belong to the original dataset, then the chosen node is the one that is closest to the root.

After the reversion procedure is finished, the new tree is analyzed. Branches with no input nodes are removed from the tree, and new nodes are inserted to enforce the one mutation rule between nodes. If no reversions were found, the program exits the tree creation stage. If reversions were found, a loop is

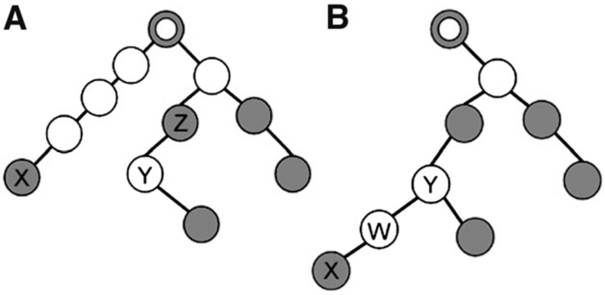

Fig. 3. Finding reversion mutations in a tree. (A) The original tree, before reversion handling is performed. Node X is suspected to have a reversion, as the distance from its skeleton ancestor, the root, is larger than the distance between X and a “found” node, Y. The distance between X and Y was found to be shorter than the distance between X and the root, which is X and Y's common ancestor, meaning, that a reversion in node X occurred. Node Z is chosen as node X's new ancestor. (B) In the process of rebuilding the tree with these ancestral relationships, node X was reassigned as node W's descendant, as two mutations were found to separate it from node Y. The new tree (B) is now smaller by two nodes than tree (A).

carried out for the reversion and tree reformatting stages until there are no more reversions, or until an upper limit for the number of cycles is reached.

2.8. Mutation analyses

After the lineage tree is completed, a list is created for all the mutations in the tree. The list includes the sequence from 3 positions upstream to 3 positions downstream to each mutation, before and after the mutation. This list is used to find DNA motifs that might increase or decrease the probability of mutations, or the probability of a specific mutation (Zuckerman et al., manuscript in preparation). An example of motif analysis in normal human peripheral blood B cell clones is shown in Fig. 4.

In addition, the amino acid encoded by the mutated codon before and after the mutation can also be deduced from the list of mutations. This information can be used to find if there are different probabilities for silent mutations or for mutations that change the physical properties of the Igs. This analysis also explores the different regions of the Igs – CDRs and framework – and can compare the mutations in each region, e.g. for performing a corrected variation of the Replacement-to-Silent (R:S) ratio analysis. Since each dataset contains many trees, plus single sequences in some cases, each dataset typically includes hundreds or even thousands of mutations for analysis, and this often yields statistically significant results.

An executable version of the IgTree program can be obtained from the authors upon request. A pseudo-code outline of the algorithm is given in the supplementary text.

3. Results

3.1. IgTree©

As mentioned above, existing phylogenetic tree constructing algorithms are not optimal for creating Ig gene micro-evolution lineage trees (Fig. 1). The algorithm described in this paper complies with all the requirements for Ig lineage trees. IgTree© is the implementation of the above-described algorithm for constructing a lineage tree from an Ig variable region gene sequence alignment. The input for the program is a file with the sequences after the alignment in NBRF/PIR format. The output is the tree file, in a format that is readable by the MTree© program (Dunn-Walters et al., 2002), with an option for an additional output file in dot language for viewing the tree graphically (dot available from www.graphviz.org; Emden Gansner and Stepthen North, 2000).

3.2. Gaps in the lineage tree

Alignment gaps are created when insertions or deletions occur. Unlike point mutations, mutations that create gaps are not necessarily limited to one nucleotide per mutation. (In in-frame sequences, gap lengths will be multiples of 3 nucleotides. In out-of-frame sequences, this limitation does not apply.) To include gaps, the algorithm assigns an affine gap penalty to the length of the mutation, such that a single event that created a long gap will be considered more probable than separated shorter gaps.

3.3. Gene conversion

The algorithm implemented in IgTree© can also accommodate the type of mutation known as gene conversion, an additional diversification mechanism operating on Ig variable region genes. This process does not take place in human or mouse Ig genes, but it is the main method for creating diversity in B cells of other species, including chickens and rabbits. Gene conversion is the nonreciprocal homologous recombination of a gene. The sequence used as a template for gene conversion is mainly from one chosen Ig segment in the genome, where part of the acceptor gene segment is replaced by a sequence from the donor segment. When a lineage tree is constructed, it is based only on the root and sampled sequences, implying that the donor's sequence, if there is one, is unknown. However, it is known that gene conversion results in changes occurring in several successive nucleotides. Therefore, a succession of point mutations with a length typical for gene conversion in the relevant species is considered as one gene conversion, when data from the appropriate species are analyzed.

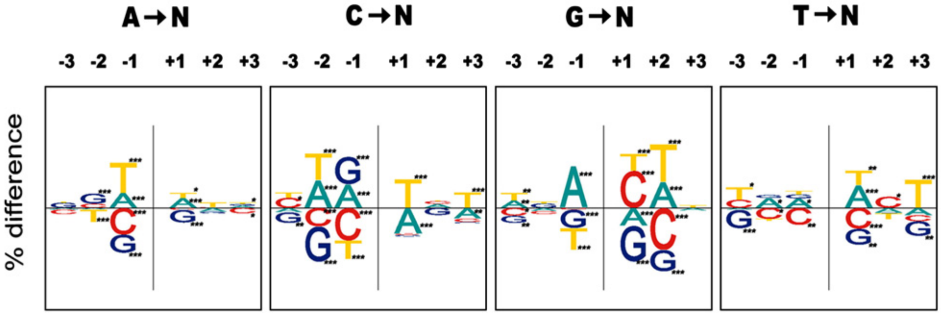

Fig. 4. Mutation analysis of motifs flanking a mutated nucleotide. Nucleotides flanking the mutated nucleotide in six locations were enumerated for each type of mutation in normal human peripheral blood B cell clones (Abraham et al., 2007); −1, −2, −3 denote the three locations flanking the mutation upstream; +1, +2, +3 denote the three locations flanking the mutation downstream. A→N, for example, is mutations from A to any nucleotide. Nucleotides on the positive and negative sides of the axis denote excess or paucity of the nucleotide around the mutation, respectively. % difference is the difference between the baseline percentage composition of the GL and the percentage composition of each base at each position flanking a particular mutation, to show any differences particular to that mutation. The levels of significance after X^2 test are indicated by asterisks: *, $p<0.05$; **, $p<0.005$; ***, $p<0.0005$. Motifs were compared to the motifs reported in the literature: R*G*YW/WR*C*Y (R = A/G, Y = C/T, W = A/T) for mutations from G and C, respectively; and W*A*/*T*W for mutations from A and T, respectively (Odegard et al., 2006). Motifs found were similar to previously reported motifs (Zuckerman et al., manuscript in preparation.).

3.4. Testing IgTree©'s performance

To test the algorithm's performance, we used a simulation of Ig lineage tree generation. The simulation starts with an Ig sequence which is the tree's root. A mutation (nucleotide substitution, insertion or deletion) is created in a random location in the sequence. The new sequence is added to the tree as a node which is a son of the root. The creation of new sequences is repeated for a predefined number of times, where the original sequence for every new node is one of the existing tree nodes, chosen randomly. The constructed tree has one mutation separating every node from its immediate ancestor In the next stage, a sample of N sequences is randomly selected from all the created sequences, and the algorithm is challenged to reconstruct the original tree from this sample. After a tree is reconstructed from the sampled data by IgTree©, it is compared with the original tree (discarding branches that are not represented in the sampled sequences). The two trees are compared by tree size and by other properties measured by MTree© (Dunn-Walters et al., 2004). The comparison is performed for trees with sizes between 20–80 nodes and sample sizes of 2 to 24 sequences, values that correspond to typical biological data. This test was performed for 25061 trees. In 80.4% of the comparisons the two trees were identical, in 16.3% of the comparisons, the IgTree© tree was smaller, and in 3.3% of the comparisons the simulation tree was smaller. The comparisons where the simulation tree was smaller (3.3%) were due to the fact that IgTree© uses a heuristic algorithm and does not always find the minimal tree, and thus the accuracy of about 97% is a reasonable estimate of the performance of the algorithm. Interestingly, in 16.3% of the cases, IgTree© constructed a tree which is more compact than the actual tree that was the source of the sequences. This is probably due to the fact that we are dealing with only a sample of all sequences and thus it is possible (both in the simulation and in real biological experiments) to create smaller trees that can explain the sample.

4. Discussion

The analysis of B cell lineage trees is often used to elucidate and better understand the dynamics of the humoral immune response in both normal and pathological conditions (reviewed below). Automating lineage tree construction from Ig variable region gene sequences, identifying all mutations including gaps and reversions, will aid in facilitating and standardizing the usage of lineage trees, thus advancing the usage of lineage trees in more studies. IgTree©, along with our MTree© program for lineage tree measurement, will contribute to the creation of a fully automated process of lineage tree analysis. In addition, several mutation analyses procedures, such as replacement and silent mutation analysis, mutational motif counts and amino acid substitutions, were integrated into IgTree©, providing much information regarding diversification and selection of Ig genes in the GC. Prior to IgTree©, the construction of lineage trees was performed via different phylogenetic tree-constructing algorithms, which required much manual correcting, or manually. The constructed tree was then manually adjusted for quantitative analyses. IgTree© will eliminate the need for this manual work, which is important as current studies typically generate thousands of sequences.

The problem of creating phylogenetic trees using the approaches of maximum parsimony and maximum likelihood is NP-complete (Day et. al, 1986; Chor et. al, 2005). Hence, IgTree© uses several heuristic procedures to reduce the running time of the program. Even so, it is most useful when applied to short sequences (~300 bases long) that are closely related to each other. In such cases, with numbers of sequences that are typical of the datasets obtained by experimentalists, the run times of the program are less than 1 min.

Unlike other phylogenetic tree-constructing programs, IgTree© is tailored for constructing lineage trees from Ig gene sequences. It can construct non-binary trees, including internal nodes; it considers all the mutations in the tree; and performs mutational analyses as described above.

A version of IgTree© implementing the described algorithm, excluding the usage of mutations with lengths of more than one nucleic-acid and reversions, was used in several studies. One study demonstrated the importance of setting a standard for sequence extraction by comparing between lineage trees from several autoimmune diseases (Steiman-Shimony et al., 2006a). A second study gained insights regarding activation and selection of B cells from autoimmune diseases by constructing lineage tree for B cell clones from autoimmune diseases (Steiman-Shimony et al., 2006b). Autoimmune trees were found to be significantly larger relative to normal controls. In contrast, comparison of the measurements for tree branching indicated that similar selection pressure operates on autoimmune and normal control clones. A third study revealed a common precursor for B cells from the peripheral blood and bone marrow in light chain amyloidosis (AL) patients, by constructing Ig gene lineage trees (Manske et al., 2006). A fourth study gave insight into the process of B cell clonal evolution in Immunoglobulin AL (Abraham et al., 2007). When compared to normal bone marrow and peripheral blood B cells, AL clones showed significant but incomplete impairment of antigenic selection, which could not be detected by conventional R and S mutation analysis. Thus, these studies provide robust examples of the use of lineage trees in delineating the role of autoimmune or neoplastic precursor B cells in the pathogenesis of autoimmunity and hematopoietic malignancies.

Although the IgTree© algorithm was designed specifically for the construction of Ig gene lineage trees, it may be used in other cases as well, given that there is a known root, sequences aligned to the root, and that the length of the sequences is in the order of hundreds of bases. In such cases, IgTree© will provide information regarding the route from the root to the observed sequences.

Acknowledgements

The work was done as part of Michal Barak's studies towards a PhD degree in Bar-Ilan University, and was supported in parts by the following grants (to R.M.): Israel Science Foundation grants 759/01-1 and 546; an Israel Cancer Research Fund project grant; a Systems Biology prize grant from Teva Pharmaceuticals; Human Frontiers Science Program — a Young Investigator Grant and a Research Grant; and a Swedish Foundation for Strategic Research grant funding the IRIS Strategic Research Center, Karolinska Institute, Stockholm, Sweden; The ministry of Science and Technology PhD scholarship for advancing women in science; The Yeshaya Horowitz Association through the Center for Complexity Science PhD scholarship; Bar Ilan University President's PhD scholarship; and a Dean's excellence PhD scholarship of The Mina and Everard Goodman Faculty of Life Sciences (to NSZ). Ron Unger is partially supported by the Israeli Ministry of Science grant number 3-2559.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jim.2008.06.006.

References

Abraham, R.S., Manske, M.K., Zuckerman, N.S., Sohni, A., Edelman, H., Shahaf, G., Timm, M.M., Dispenzieri, A., Gertz, M.A., Mehr, R., 2007. Novel analysis of clonal diversification in blood B cell and bone marrow plasma cell clones in immunoglobulin light chain amyloidosis. J. Clin. Immunol. 27 (1), 69.

Banerjee, M., Mehr, R., Belelovsky, A., Spencer, J., Dunn-Walters, D., 2002. Age- and tissue-specific differences in human germinal center B cell selection revealed by analysis of IgVH gene hypermutation and lineage trees. Eur. J. Immunol. 32 (7), 1947.

Chor, B., Tuller, T., 2005. Maximum likelihood of evolutionary trees is hard. In: Miyano, S., et al. (Eds.), Research in computational molecular biology. Springer Berlin, Heidelberg, p. 296.

Day, W., Johnson, D., Sankoff, D., 1986. The computational complexity of inferring rooted phylogenies by parsimony. Math. Biosci. 81, 33.

Dunn-Walters, D.K., Belelovsky, A., Edelman, H., Banerjee, M., Mehr, R., 2002. The dynamics of germinal centre selection as measured by graph-theoretical analysis of mutational lineage trees. Dev. Immunol. 9 (4), 233.

Dunn-Walters, D.K., Banerjee, M., Mehr, R., 2003. Effects of age on antibody affinity maturation. Biochem. Soc. Trans. 31 (2), 447.

Dunn-Walters, D.K., Edelman, H., Mehr, R., 2004. Immune system learning and memory quantified by graphical analysis of B-lymphocyte phylogenetic trees. Biosystems 76 (1–3), 141.

Emden Ga nsner, R., Stephen North, C., 2000. An open graph visualization system and its applications to software engineering. Soft. Pract. Exp. 30 (11), 1203.

Jacob, J., Kelsoe, G., Rajewsky, K., Weiss, U., 1991. Intraclonal generation of antibody mutants in germinal centres [see comment] Nature 354, 389.

Jacob, J., Kelsoe, G., 1992. In situ studies of the primary immune response to (4-hydroxy-3-nitrophenyl)acetyl. II. A common clonal origin for periarteriolar lymphoid sheath-associated foci and germinal centers. J. Exp. Med. 176, 679.

Kocks, C., Rajewsky, K., 1988. Stepwise intraclonal maturation of antibody affinity through somatic hypermutation. Proc. Natl. Acad. Sci. U. S. A. 85, 8206.

Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., McGettigan, P.A., McWilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J., Higgins, D.G., 2007. ClustalW2 and ClustalX version 2. Bioinformatics 23 (21), 2947.

Li, W.H., 1981. Simple method for constructing phylogenetic trees from distance matrices. Proc. Natl. Acad. Sci. U. S. A. 78 (2), 1085.

Manske, M.K., Zuckerman, N.S., Timm, M.M., Maiden, S., Edelman, H., Shahaf, G., Barak, M., Dispenzieri, A., Gertz, M.A., Mehr, R., Abraham, R.S., 2006. Clonal CD19+B cells skew the Ig VL gene repertoire in the CD138-negative compartment of bone marrow in light chain amyloidosis. Clin. Immunol. 120 (1), 106.

Manser, T., 1989. Evolution of antibody structure during the immune response. The differentiative potential of a single B lymphocyte. J. Exp. Med. 170, 1211.

Odegard, H., Schatz, G., 2006. Targeting of somatic hypermutation. Nat. Rev., Immunol. 6, 573.

Saitou, N., Nei, M., 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4 (4), 406.

Shannon, M., Mehr, R., 1999. Reconciling repertoire shift with affinity maturation: the role of deleterious mutations. J. Immunol. 162, 3950.

Shahaf, G., Barak, M., Zuckerman, N.S., Swerdlin, N., Gorfine, M., Mehr, R., in press. Antigen-driven selection in germinal centers as reflected by the shape characteristics of immunoglobulin gene lineage trees: a large-scale simulation study. J. Theor. Biol.

Steiman-Shimony, A., Edelman, H., Barak, M., Shahaf, G., Dunn-Walters, D., Stott, D., Abraham, R.S., Mehr, R., 2006a. Immunoglobulin variable-region gene mutational lineage tree analysis: Application to autoimmune diseases. Autoimmun. Rev. 5 (4), 242.

Steiman-Shimony, A., Edelman, H., Hutzler, A., Barak, M., Zuckerman, N.S., Shahaf, G., Dunn-Walters, D., Stott, D.I., Abraham, R.S., Mehr, R., 2006b. Lineage tree analysis of immunoglobulin variable-region gene mutations in autoimmune diseases: chronic activation, normal selection. Cell. Immunol. 244, 130.

Tabibian-Keissar, H., Zuckerman, N.S., Barak, M., Dunn-Walters, D., Steiman-Shimony, A., Chowers, Y., Ofek, E., Rosenblatt, K., Schiby, G., Mehr, R., Barshack, I., in press. B cell clonal diversification and gut-lymph node trafficking in Ulcerative Colitis revealed using lineage tree analysis. European Journal of Immunology.