Biophysics, Vol. 48, Suppl. 1, 2003, pp. S116–S122.
Original Russian Text Copyright © 2003 by Merkeev, Novichkov, Mironov.

TOGs vs COGs: a Database of Supergenomes Built from Proteome Complements

I. V. Merkeev, P. S. Novichkov, and A. A. Mironov

State Scientific Center GosNIIGenetica, Moscow, 113545 Russia; E-mail: ivmerkeev@rambler.ru

Received November 10, 2003; in final form, December 20, 2003

Abstract—Evolutionary forces acting on genomes result in gene duplications, gene losses, and gene acquisitions. Generally, it is difficult to reconstruct the exact history of the evolutionary process because of its complex nature. A widely used approach is to find orthologous groups by comparing completely sequenced genomes. This approach resulted in several databases that helped predict protein functions and provided deep insights into protein evolution. These procedures, however, did not take into account the taxonomy of organisms. Here we present a robust procedure that creates clusters of orthologous groups at each node of the evolutionary tree. As a result of this procedure, a tree of orthologous genes (TOG) is obtained. Each cluster is a "supergene" and is represented by a "consensus" sequence obtained by multiple alignment of orthologous and paralogous genes from one TOG. The procedure has been applied to the tree corresponding to the Bacillus group of bacteria. Protein complements from 26 bacterial species were used to create TOGs at all tree nodes of the tree 5 levels deep. 2520 TOGs were obtained at the root node. The analysis of these TOGs showed that they agree well with COGs from the COG database.

Key words: SNP, supergene, supergenome, cluster of orthologous groups, tree of orthologous groups

INTRODUCTION

Orthologs and paralogs are widely used terms in modern comparative genomics. Orthologs are genes derived from a single ancestral gene as a result of the speciation event, while paralogs are genes that result from gene duplication events.

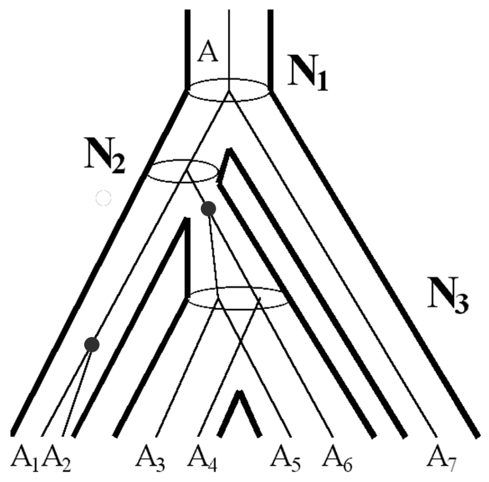

Fig. 1. Evolution by gene duplication. Nodes N_1, N_2, N_3 represent speciation events resulting in orthologs. Filled circles (●) mark gene duplication events resulting in paralogs.

When analyzing orthologous/paralogous relationships between genes from different genomes, one inevitably comes across a very difficult problem of finding paralogs, genes that diverged after the duplication of the ancestral gene. Since such duplications can happen at different evolutionary times, paralogs will create an independent orthologous group if closely related genomes are considered (e.g., strains of the same species as an extreme case), while for analysis of distant genomes, these proteins should be treated as a single paralogous group. Therefore, we suggest that clusters of orthologous genes should be defined at each node of the evolutionary tree.

Figure 1 shows how an ancestral gene A can create a family of genes A_1, A_2, A_3, A_4, A_5, A_6, A_7 by three speciation events N_1, N_1, N_3 and two gene duplication events. The real evolution of gene families is far more complex than this simple example, creating a complex network of orthologs and paralogs. A gene

(a)

```
mevkktswteeedrILYQAHKRLGNRWAEIAKLLp--------grtdna
advkrgniskeeedIIIKLHATLGNRWSLIASHLp--------grtdne
-dlkrgnftadeddLIVKLHSLLGNKWSLIAARLp--------grtdne
-nvnkgnfteqeedLIIRLHKLLGNRWSLIAKRVp--------grtdnq
-dlkrgcfsqqeedHIVALHQILGNRWSQIASHLp--------grtdne
-dlkrgifseaeenLILDLHATLGNRWSRIAAQLp--------grtdne
---shptysemiaaAIRAEKSRGGSSRQSIQKYIkshykvgh--nadlq
---shpkysdmiasALESLKEKKGSSRQAILKYVkanftvgd--nanvh
---ahpsssemvlaAITALKERGGSSAQAIRKYIeknytvdi-kkqaif
---thpptsvmvmaAIKALKERNGSSLPAIKKYIaanykvdv-vknahf
----kpstlsmivaAITAMKNRKGSSVQAIRKYIlannkgintshlgsa
---ahppvidmitaAIAAQKERRGSSVAKIQSYIaakyrcdi-nalnph
---ahppyintikeAIKQLKDRKGSSKQAILKFIsqnyklgd---nviq
---ahppvatmvvtAILGLKERKGSSMVAIKKYIaanyrv-----dvar
---shptyeemikdAIVTLKERTGSSQYAIQKFIeekrkelp-ptfrkl
---nhptyqvmistAIAHYKDRTGSSQPAIIKYIeanynvap-dtfktq
-mrssakqeelvkaFKALLKEEKFSSQGEIVAALqeq-gfd--ninqsk
--------marhrrIVDILNRQPVRSQSQLAKLLadn-gls---vtqat
enlnpvtrtarqalILQILDKQKVTSQVQLSELLlde-gid---itqat
-----mnkgqrhikIREIITSNEIETQDELVDMLkqd-gyk---vtqat
-----mnkgqrhikIREIIMSNDIETQDELVDRLrea-gfn---vtqat
-----mrkrdrhqlIKKMITEEKLSTQKEIQDRLeah-nvc---vtqtt
gpevaanragrqarIVAILSSAQVRSQNELAALLaae-gie---vtqat
```

(b)

```
VTTSFFVLYFKTSEHMEAAYSLAITITMLMTTTLLTYFLIQKGTPKIAIAIISIGLFCI-EGSFFAASLVQFINGAYIVVLIALAIIFV
AVTSCTVLYFRTSAHMEAAYGLAITITMLMTTILLAYYLIKEGV-KPLLASLLMAFFAFIEFIFFLASAVKFMHGGYVVVVLALAIVFV
AITTSIVLLFKTSAHMEAAYGLAITITMLMTTILLSFFLIQKGV-KRGLVLLMMIFFGILEGIFFLASAVKFMHGGYVVVIIAVAIIFI
AVTSCTVLAFRTSAHMEAAYGLAITITMLMTTILLKYYLIKKGT-RPILAHLVMAFFALVEFIFFLASAIKFMHGGYAVVILALAIVFV
ITTIALVLFFRTSAHMEAAYGLSITISMLTTTILLYEWLVLKQGHNLANL-LFVIFFSTINILFMGSSLSKFTHGGYVSLLITLLIASV
AATIGIVFLFRTSEHMEAAYGLAITVTMLMTTVLLFEYLSLRKVNLSLRL-VFLLLFGAIETMFLISSLAKFLHGGYVTVIIAAFIGAI
LAASFIVVYFQSSAHMEAAYGLAITVTMLMTTTLLTVYLSHYQKVKKVLVGLFFTVFIFIEGLFFAASAVKFMHGGYVVVIIAAMILFV
IACSLVVVTFRTSHHMEAAYGLSITVTMLMTTALLYFYLIQNGHS-RWLAYLVTFFFGAIEVVFFISSVVKFFRGGFVAVLIGLIILVV
LACSAIVLAFRTSTHMEAAYGLSITITMLMTTILLLFYLLDK-IP-AWSAYLISLFFAAIEVVFFFSSAAKFFHGGYVAVGMAVFLLCI
AALQRLSQYHLSRFKLYSTYVMTHTILLLLVLLAVSLYLSQPLSLIFYLKSLLLILIYEIGIVFILFHIQTISHRLFMTFIYALAMGIV
APLERLYLVPYSQLKLYLTYISVHVVILMLMLLMISLLMHQPLSILFYLKTLIIVLFYEAGIALLLFKINVLSHRIFMAIIYTVAIGII
AALQRLSQYHLSRFKLYSTYVMTHTILLLLVLLAVSLYLSQPLSLIFYLKSLLLILIYEIGIVFILFHIQTISHRLFMTFIYALAMGIV
Supergene:
AATS-IVLYFRTSAHMEAAYGLAITITMLMTTILLSYYLIQKGT-K-YLASL-MIFFAAIEIVFFLAS-VKFMHGGYVVVIIALAI-FV
```

Fig. 2. (a) An example showing how a common sequence segment produces a similar function for the orthologous family. A multiple alignment shows here its core positions (bold uppercase symbols) and non-homologous positions (lowercase symbols). (b) An example of a TOG together with its "consensus" sequence (supergene). "Garbage" positions in the supergene are marked by hyphens.

can be either an ortholog or a paralog depending on which subtree of the evolutionary tree is being considered. For instance, gene A_3 is an ortholog relative to node N_3, while this gene (together with his ortholog A_5) is a paralog relative to node N_2. Genes A_1, A_4, A6, A_7 are orthologs relative to node N_1. A gene is considered to be an ortholog or a paralog relative to a particular node of the evolutionary tree if its ancestral gene at that node resulted from a speciation event or a gene duplication event correspondingly.

The usefulness of orthologs and paralolgs in modern genomics comes from the fact that orthologs generally perform the same function while paralogs perform similar functions. The reconstruction of orthologs and paralogs at each node of the evolutionary tree can shed light on the evolution of function. We can give several examples how the knowledge of orthologs and paralogs has helped solving some difficult issues. Comparative studies of bacterial transcriptional regulation often use orthologs, assuming

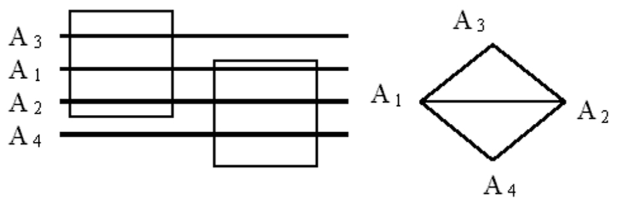

Fig. 3. The triangle rule can lead to the non-transfer of function.

that they tend to be regulated in the same way [1–4]. It is possible to predict functional coupling between genes if orthologs of genes forming a functional cluster in one organism will form a cluster in another organism [5]. Mirny and Gelfand [6] have found specificity-determining positions in the LacI/PuR family of bacterial transcription factors, looking for residues that are conserved among orthologs and are different in paralogs. Orthologs and paralogs also help to understand the evolution by gene duplication, which is thought to be a major force in creating organismic complexity [7, 8].

Although several efforts have been undertaken to create clusters of orthologous groups (COGs) [9–12], no one is capable to create such clusters at each node of the evolutionary tree. We used the COG database [13] as the most complete high-quality database or orthologous genes to tune several key parameters of our procedure.

In building clusters of orthologous groups, one inevitably comes to the problem of the cluster common core, the homologous sequence segment that all orthologous genes are supposed to have in order to perform the same biological function (Fig. 2). The procedure that underlies the COG database [9] uses the triangle pattern to find orthologous genes. If gene A_1 is an ortholog to gene A_2, gene A_2 is an ortholog to gene A_3, and gene A_3 is an ortholog to gene A_1, then all these three genes are orthologs. Triangles that have common sides are then merged. This procedure creates, however, a major difficulty (Fig. 3). Let us assume that genes A_1, A_2, A_4 form another triangle pattern with a common side A_1A_2. Then genes A_3 and A_4 might not have a common homologous sequence segment and in this case it is highly likely that they are not orthologs. This difficulty is resolved in the COG database by the manual revision of COGs.

Our procedure is based on the direct definition of orthologs and paralogs, and makes use of the following simple idea. If we have two species with their protein complements, we can find orthologs by running a similarity search procedure (e.g. BLAST), find bidirectional best hits (BBHs), and choose orthologs from BBHs using some system of rules. Then it is possible to find paralogs in each species by finding genes that are not declared orthologs and which are more similar to orthologs from their own species than orthologs are between themselves. Then we can form a new “genome,” putting into it all orthologous families and genes that did not find any match. Since this new “genome” is an artificial construct and it includes all genes from both species, this new genome is called a supergenome built from protein complements of both species. In the same way, we can also find orthologs and paralogs between two supergenomes and build a next-level supergenome. Repeating the procedure for all nodes of the tree, we will eventually obtain the root-level supergenome. Since clusters of orthologous groups are defined at each node of the evolutionary tree, they are called TOGs (Tree of Orthologous Groups). A supergenome is a collection of TOGs accumulated at a particular node of the evolutionary tree. A supergene is a “consensus” sequence, and is thought to be the ancestral sequence, for a TOG.

There are three fundamental differences between our procedure and the procedure that underlies the COG database: (i) our procedure is completely automated, so it does not require any manual intervention; (ii) our procedure detects orthologs and paralogs basing on their direct biological meaning without going to artificial methods like the triangle pattern; (iii) our procedure creates clusters of orthologous groups at each node of the evolutionary tree and gives clear indication of the timing of gene duplication events that result in paralogs; (iv) the time required to run our procedure depends linearly on the number of genomes.

METHODS

The basic step in the overall procedure to obtain TOGs at all nodes of the evolutionary tree is to compare two supergenomes, find orthologs and paralogs, put them into one TOG and to merge these two supergenomes into the supergenome lying higher in the evolutionary tree. Since each TOG represents a multiple alignment of protein sequences, it has first to be converted into a “consensus” sequence (a supergene), and then consensus sequences from both

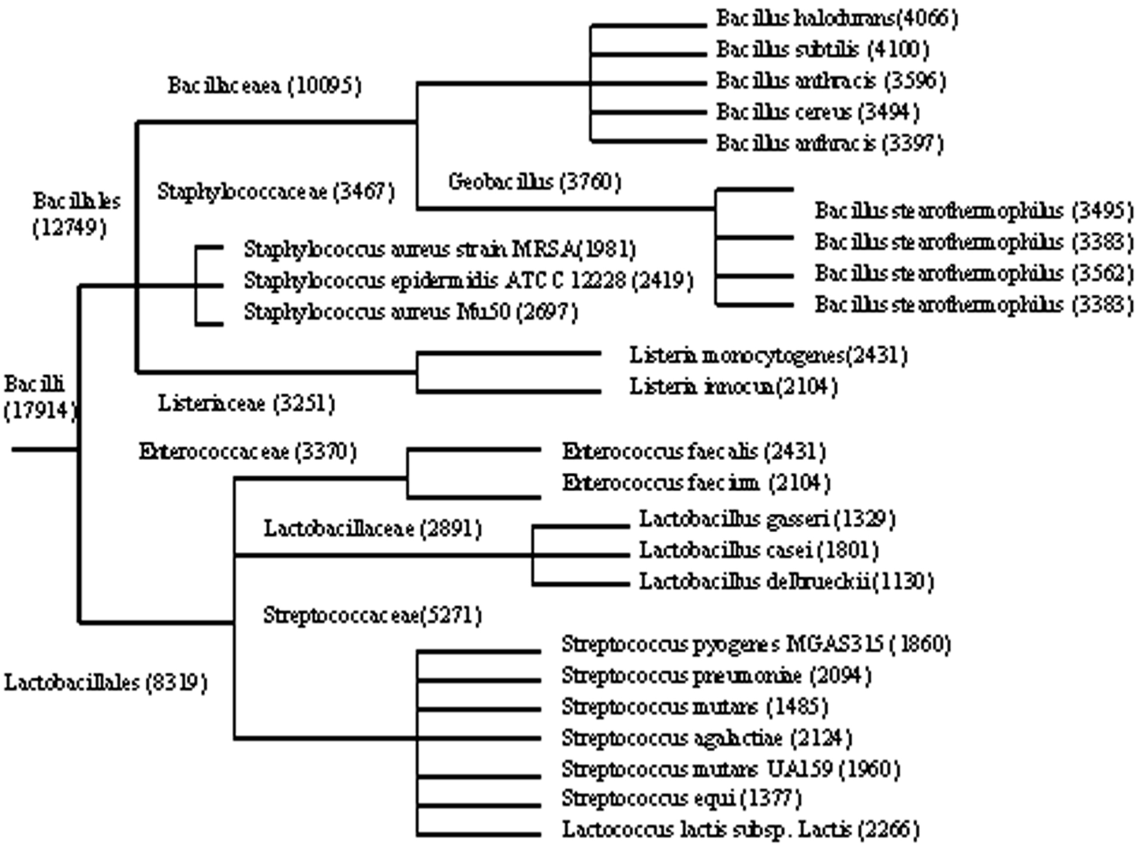
Bacillus halodurans(4066)
Bacillus subtilis (4100)
Bacilhceaea (10095)
Bacillus anthracis (3596)
Bacillus cereus (3494)
Bacillus anthracis (3397)
Bacillales (12749)
Staphylococcaceae (3467)
Geobacillus (3760)
Bacillus stearothermophilus (3495)
Bacillus stearothermophilus (3383)
Staphylococcus aureus strain MRSA(1981)
Bacillus stearothermophilus (3562)
Staphylococcus epidermidis ATCC 12228 (2419)
Staphylococcus aureus Mu50 (2697)
Bacillus stearothermophilus (3383)
Listeria monocytogenes(2431)
Bacilli (17914)
Listeria innocua(2104)
Listeriaceae (3251)
Enterococcaceae (3370)
Enterococcus faecalis (2431)
Enterococcus faecium (2104)
Lactobacillaceae (2891)
Lactobacillus gasseri (1329)
Lactobacillus casei (1801)
Lactobacillus delbrueckii(1130)
Streptococcaceae(5271)
Streptococcus pyogenes MGAS315 (1860)
Lactobacillales (8319)
Streptococcus pneumoniae (2094)
Streptococcus mutans (1485)
Streptococcus agalactiae (2124)
Streptococcus mutans UA159 (1960)
Streptococcus equi (1377)
Lactococcus lactis subsp. Lactis (2266)

Fig. 4. The part of the bacterial taxonomy tree corresponding to the *Bacillus* group. The number of TOGs at each node of the tree is shown in parentheses.

supergenomes are compared to find orthologs and paralogs. Sequences in each newly created TOG are subjected to multiple alignment, and all TOGs are then written to the relational database to launch the procedure at next nodes of the evolutionary tree.

Obtaining Supergenes from a TOG Multiple Alignment

Since each TOG represents a multiple alignment of orthologs and paralogs, it has to be converted into a "consensus" sequence. When the procedure starts from the proteome complements of organisms, it does not need to create consensus sequences, since at this level a supergene is a gene itself.

Each column of the TOG multiple alignment is converted into a consensus amino acid residue which is determined as the most probable ancestral amino acid type in this position:

$$\alpha_{consensus} = \arg\max_{\alpha=1}^{20}\left(p_\alpha = f_\alpha \cdot \prod_{\beta=1}^{20} q_{\alpha\beta}^{n_\beta}\right)$$

where $\alpha_{consensus}$ is the most probable ancestral amino acid, f_α is the background frequency of amino acid α, $q_{\alpha\beta}$ is the transition probability from amino acid α to the amino acid β, n_β is the number of times amino acid β occurs in the column, p_α is the likelihood that amino acid α was the ancestral amino acid. Likelihoods for all 20 amino acids are then sorted in decreasing order: $p_1 \geq p_2 \geq \ldots \geq p_{20}$.

Then the decision is made whether the most probable ancestral amino acid type is left as the symbol representing this column or this column is declared to be a "garbage" symbol [14]. If the following condition holds than the column is converted to a meaningful symbol:

$$\frac{p_1}{p_3} > \lambda + \mu \cdot N_{sign}^2$$

where p_1 is the likelihood of the most probable ancestral amino acid, p_3 is the likelihood of the third most probable ancestral amino acid, N_{sign} is the number of significant amino acids in the alignment column (the number of sequences in the TOG multiple alignment minus the number of gaps in the alignment column), $\alpha = 0.2$ and $\mu = 0.8$ are experimentally found parameters [14]. An example of a TOG multiple alignment together with its "consensus" sequence is given in Fig. 2b.

Running a Special BLAST-like Procedure

After all TOGs from two supergenomes are converted into consensus sequences, a special BLAST-like procedure is run between these two supergenomes which is called TOG-BLAST. TOG-BLAST combines ideas from FASTA[15], original BLAST[16] and dynamic programming. When a supergene from the first supergenome is "tog-blasted" against all supergenes from the second supergenome, all l-tuples from this supergene are hashed to a hash table. When a supergene from the second supergenome is compared with the supergene from the first supergenome using TOG-BLAST, all l-tuples from the supergene from the second supergenome are looked up in the hash table and each l-tuple hit increments the number of hits at a particular diagonal. Only diagonals with N or more l-tuple hits are considered for further processing.

At all diagonals, the leftmost l-tuple hit is then extended leftward until the total score becomes negative. The maximum score achieved during this extension and the left end of the segment corresponding to this score are memorized. In the same way, this l-tuple is extended rightward. The result of this extension is a maximal segment pair (MSP). Then the l-tuple leftmost to the resulting MSP is chosen (if such a tuple can be found). The same extension process is applied to this l-tuple. The diagonal is scanned from left to right until no 2-tuple hits are found anymore. Thus, a diagonal with N l-tuple hits or more can give one or more MSPs. Only MSPs with score 20 or more are considered for further processing.

Computational experiments with different values for N and l have allowed us to determine the optimal value for these parameters: $N = 5$ and $l = 2$ (data not shown). 3-tuples are not conserved in orthologous families from distant organisms, and for $l = 1$ there is a lot of non-homologous diagonals, so it is difficult to choose a cutoff value for N. Choosing $N = 5$ saves time on processing candidate diagonals, since computational experiments showed that the procedure gives the same orthologous families when $N < 5$ and $N = 5$.

MSPs are then ordered by their left ends belonging to one member of the pair. Let us assume that up to this point we have obtained a set of N_{msp} ordered MSPs having scores s_i, left ends in one sequence l_i and left ends in the other sequence m_i. Since MSPs are ordered, then the condition $l_{i+1} \le l_i$ holds for i from 1 to $N_{msp} - 1$.

Using the following recurrent relationship, we can obtain the score of the maximal scoring chain of MSPs:

$$SCORE_1 = s_1$$

$$SCORE_{i+1} = \max\left(s_{i+1} + SCOREj - abs(d_{i+1} - d_j) : j \le i, l_j \ge l_{i+1}, m_j \ge m_{i+1}\right)$$

where d_{i+1}, d_j are diagonals of $(i + 1)$th and jth MSPs, $-\text{abs}(d_{i+1} - d_j)$ is the gap penalty for introducing gaps. To make the TOG-BLAST score significant, an additional requirement is imposed demanding that the total length of segments that form the chain should exceed the half length of the longest protein in order to avoid domain-based and signature-based matches.

After TOG-BLAST scores are obtained for all possible pairs of supergenes from both supergenomes, bidirectional best hits are obtained (BBHs). BBHs are then ordered based on their scores in descending order. Since BBHs with low scores can potentially lead to false-positive orthologs, BBHs with scores less than 20% of the median BBH score are discarded. The justification for this removal comes from the work of Novichkov and co-workers [17] who showed that in non-paralogous COGs the rate of ortholog evolution cannot be 20-fold different between the fast- and slow-evolving genes.

A Smith–Waterman algorithm (SW) [18] was applied to all BBHs, and segments from both supergenes making the BBH and giving the maximal SW score were stored in the program database. These conserved segments are called BBH cores. The core from the first supergene is called the first BBH core, and the core from the second supergene is called the second BBH core.

Finding Paralogs

To find paralogs, the procedure starts from the BBH with the maximal score, and it runs TOG-BLAST with the first BBH core as the query against all supergenes in the first supergenome that are not declared orthologs yet. If a supergene has the score with the first core greater than the BBH score, it is declared as a paralog. To find a paralog core, a Smith-Waterman algorithm is applied to the first BBH core and the found supergene. The supergene segment giving the maximal SW score is declared as the paralog core. The same procedure is applied to find paralogs in the second supergenome for the same BBH pair. In case the new paralog is a part of the BBH pair, this BBH pair is cancelled. The procedure proceeds now to the next BBH pair with lower score and so on until all paralogs are found in both supergenomes.

TOG Multiple Alignment

BBH cores are aligned using the Needleman–Wunsch algorithm [19] with affine gap penalties inside sequences and zero gap penalties at the end. After that, a consensus sequence is formed from this alignment. Then a paralog is aligned with this consensus sequence using the same algorithm. This procedure is repeated for all other paralogs. Since ortholog and paralog cores are parts of their supergenes and each supergene represents generally a multiple alignment, gaps introduced into ortholog and paralog cores are also introduced in their respective multiple alignments. The resulting multiple alignment of multiple alignments is written to the relational database to be used at the next round of the procedure.

RESULTS AND DISCUSSION

The procedure has been applied to the tree corresponding to the *Bacillus* group of bacteria (Fig. 4). Proteome complements for these species were downloaded from the NCBI ftp site[20]. Since the taxonomy tree can have several branches at each node of the tree, this tree was binarized, grouping together the first two branches, then these two branches and the third branch and so on (Fig. 5). Genomes of 26 bacterial species were used to create TOGs at all tree nodes of the tree 5 levels deep.

17 914 TOGs were obtained at the root node of the tree including 2520 TOGs that have the root node as their origin and 8129 TOGs that consisted of only one gene. 7265 TOGs have different intermediary nodes of the tree as their origins.

We used the COG database to test our procedure. Since this database also contains protein sequences for some organisms from the *Bacillus* group, we matched protein sequences in our 2520 root-level TOGs against the COG database. Each matched protein sequence obtained a number corresponding to the number of the COG where this sequence was found. 1925 root-level TOGs contained at least two matched proteins from the COG database. Of these TOGs 1921 contained matched proteins with same COG number. Only 4 root-level TOGs have protein sequences from two different COGs. This test proves that our procedure basically results in the same clusters of orthologous groups, though in our procedure each cluster is split at lower nodes of the tree revealing its fine structure.

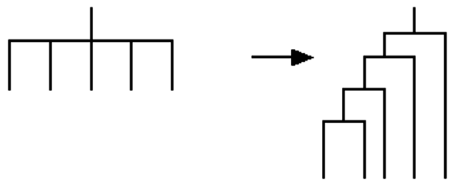
Fig. 5. Tree binarization.

8129 root-level TOGs consisting of orphan genes that did not find their match during the procedure run represent an evolutionary puzzle. Their origin is not clear and further research is needed to look for their homologs in protein sequence databases. 7265 top level TOGs belonging to intermediary nodes might arise either through horizontal transfer or by vertical descent if their orthologs in other lineages are lost during the course of evolution. It is possible that some TOGs, apart from containing ortholog and paralogs that came to them through vertical descent, contain genes that came to them by horizontal transfer events and therefore violate the definition of orthologs and paralogs. A postprocessing is needed to match gene trees against species trees to detect and isolate all horizontal transfer events.

The average length of the root-level TOG is 260 amino acids, which corresponds approximately to two protein domains. As the procedure goes from the leaves to the root of the evolutionary tree, protein sequences are truncated to leave the most conserved evolutionary cores. There is always a possibility that cores be truncated to such an extent that they cannot be used anymore for resolving orthologs and paralogs. When our procedure is applied to deeper trees, we can carry out the final test whether core truncation is a serious threat.

We plan to apply our procedure to all pub-licly available bacterial genomes and post these TOGs to the Web for public access. Such an effort might improve the quality of research in many areas of comparative genomics and can deepen our understanding of the evolution by gene duplications.

ACKNOWLEDGMENTS

We are grateful to M.Gelfand for discussion.

This study was partially supported by grants from HHMI (55000309) and LICR (CRDF RBO-1268).

REFERENCES

1. Mironov, A.A., Koonin, E.V., Roytberg, M.A., and Gelfand, M.S., Computer analysis of transcription regulatory patterns in completely sequenced bacterial genomes, *Nucleic Acid Research*, 1999, vol. 27, pp. 2981–2989.
2. Gelfand, M.S., Koonin, E.V., and Mironov, A.A., Prediction of transcription regulatory sites in Archaea by a comparative genomic approach, *Nucleic Acid Research*, 2000, vol. 28, pp. 695–705.
3. Panina, E.M., Mironov, A.A., and Gelfand, M.S., Comparative analysis of FUR regulons in gamma-proteobacteria, Nucleic Acid Research, 2001, vol. 29, pp. 5195–5206.
4. McCue, L.A., Thompson, W., Carmack, C.S., Ryan, M.P., Liu, J.S., Derbyshire, V., and Lawrence, C.E., Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. Nucleic Acid Research, 2001, vol. 29, pp. 774–782.
5. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G.D., and Maltsev, N., The use of gene clusters to infer functional coupling, *Proc. Natl. Acad. Sci. USA*, 1999, vol. 96, pp. 2896–2901.
6. Mirny, L.A. and Gelfand, M.S., Using orthologous and paralogous proteins to indentify specificity–determining residues in bacterial transcription factors, 2002, *J. Mol. Biol.*, vol. 321, pp. 7–20.
7. Jordan, I.K., Makarova, K.S., Spouge, J.L., Wolf, Y.I., and Koonin, E.V., Lineage–specific gene expansions in bacterial and archaeal genomes, Genome Research, 2001, vol. 11, pp. 555–565.
8. Lynch, M. and Conery, J.S., The evolutionary fate and consequences of duplicate genes, *Science*, 2000, vol. 290, pp. 1151–1155.
9. Tatusov, R.L., Koonin, E.V., and Lipman, D.J., A genomic perspective on protein families, *Science*, 1997, vol. 278, pp. 631–637.
10. Uchiyama, I., MBGD: microbial genome database for comparative analysis, Nucleic Acid Research, 2003, vol. 31, pp. 58–62.
11. Remm, M., Storm, C.E.V., and Sonnhammer, E.L.L., Automatic clustering of orthologs and in–paralogs from pairwise species comparisons, *J. Mol. Biol.*, 2001, vol. 314, pp. 1041–1052.
12. Yuan, Y.P., Eulenstein, O., Vingron, M., and Bork, P., Towards detection of orthologues in sequence databases, *Bioinformatics*, 1998, vol. 14, pp. 285–289.
13. Tatusov, R.L., Natale, D.A., Garkavtsev, I.V., Tatusova, T.A., Shankavaram, U.T., Rao, B.S., Kiryutin, B., Galperin, M.Y., Fedorova, N.D., and Koonin, E.V., The COG database: new developments in phylogenetic classification of proteins from complete genomes, *Nucleic Acid Research*, 2001, vol. 29, pp. 22–28.
14. Merkeev, I.V. and Mironov, A.A., Profile clusters derived from BLOCKS suggest a simple model of column evolution in multiple alignments of protein families, 2003, Article in press.
15. Lipman, D.J. and Pearson, W.R., Rapid and sensitive protein similarity searches, *Science*, 1985, vol. 227, pp. 1435–1441.
16. Altschul, S., Gish, W., Miller, W., Myers, E.W., and Lipman, D., A basic local alignment search tool, *J. Mol. Biol.*, 1990, vol. 215, pp. 403–410.
17. Novichkov, P.S., Gelfand, M.S., and Mironov, A.A., Relative mutation rate of bacterial proteins and prediction of the distance between orthologous genes. Proceedings of the third international conference on bioinformatics of genome regulation and structure. Novosibirsk, Russia, 2002, pp. 177–180.
18. Smith, T.F. and Waterman, M.S., Identification of common molecular subsequences, *J. Mol. Biol.*, 1981, vol. 147, pp. 195–197.
19. Needleman, S.B. and Wunsch, C.D., A general method applicable to the search for similarities in the amino acid sequence of two proteins, *J. Mol. Biol.*, 1970, vol. 48, pp. 443–453.
20. Benson, D.A., Karsch–Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A., and Wheeler, D.L., GenBank, *Nucleic Acid Research*, 2002, vol. 30, pp. 17–20.