Online Supplementary Material

Coevolution of gene families in prokaryotes.

Otto X. Cordero, Berend Snel and Paulien Hogeweg

Theoretical Biology and Bioinformatics, University of Utrecht,

Padualaan 8, 3584 CH Utrecht, The Netherlands

Corresponding author: Otto X. Cordero. (o.x.corderosanchez@bio.uu.nl)

Supplementary Figure S1

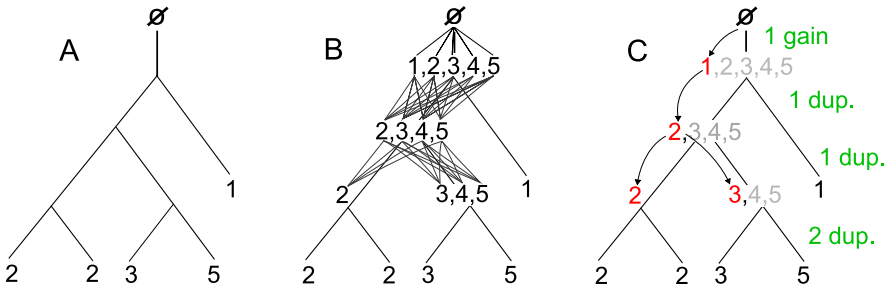

Figure 1: **Illustration of the reconstruction algorithm. A** The mapping of a hypothetical COG species distribution onto the leaves of a species tree. Our method adds an empty ancestor before the root of the tree to include the cost of gene creation and duplication when a family is present in LUCA (Last Universal Common Ancestor). **B** First step of the reconstruction. Starting from the leaves and going to the root, all possible paths are calculated. **C** Second and final step. Starting from the root, the path that minimize the cost function is selected. The process is repeated for all COGs.

S1. Reconstruction of ancestral gene content. The reconstruction of ancestral gene content is performed per COG with maximum parsimony, i.e. finding the evolutionary scenario which renders the least number of (weighted) events. This is done by minimizing the cost function $S = \delta + \lambda d + \gamma g$, where δ is the number of deletions, d number of duplications, λ duplication cost, g number of gains and γ gain cost (see (Mirkin *et al.*, 2003) for related methods). This is implemented as an extension of PAUP's generalized parsimony algorithm (Swofford, 1998; Mirkin *et al.*, 2003). Fig S1 provides an illustration of the steps in the algorithm.

The gain-loss reconstruction is that described in (Mirkin *et al.*, 2003). This is simply the same method as our full reconstruction but using 0 or 1 in the leaves to specify presence or absence of the family, instead of using the actual family sizes.

S2. Evaluation of reconstruction results.

Based on our simulation of genome evolution, we can evaluate how the reconstruction algorithm approximates the real (simulated) values of the ancestors in the tree. For this we calculate the average percentage error, based on 100 simulations and 100 corresponding reconstructions with

Gain cost	Size of LUCA	Percentage of events with HGT
3	1900	5%
5	2400	2%
10	2934	0.0009%

Table 1: **Comparison of different gain costs.** The percentage of events is calculated relative to HGT, duplications and deletions. We see also that even with gain cost 3 the size of LUCA is close to that of an average bacterium species. This suggests that our reconstructions already underestimate the amount of transfer which may have happened at early stages of cellular life.

cost of duplication 2 and cost of innovation 3. The calculated percentage error is only 1.3%, which we think is a reasonable error margin.

We have used different costs of innovation and duplication to validate our results. The costs used in the reconstruction follow the logic that deletions could be more favored than duplications, and that gene creation is less likely than duplication. However, we must warn the reader that there is no known estimate of the real values. Probably the most controversial parameter is the gain penalty, since it represents the likelihood of horizontal gene transfer (HGT) events. In the next section we study the effects of changing this important parameter.

S3. Different gain costs and function prediction.

We have checked the effect of the different gain costs on the function prediction. While the partial correlation score seems to be insensitive to differences in gain penalty, reconstructions with a higher gain cost give a somewhat better prediction performance when using the sign score (Fig S2).

What happens when the cost of a gain is high enough is that creation events gets pushed down to the last common ancestor, adding losses in the intermediate branches to fill the 'gaps' caused by this shift. Changes in gain costs affect only the reconstruction of gain and loss events. Accordingly, a similar increment in performance is observed in the sign score for the gain-loss reconstruction.

The better function prediction achieved with higher gain costs should not be confused as a

Supplementary Figure S2.

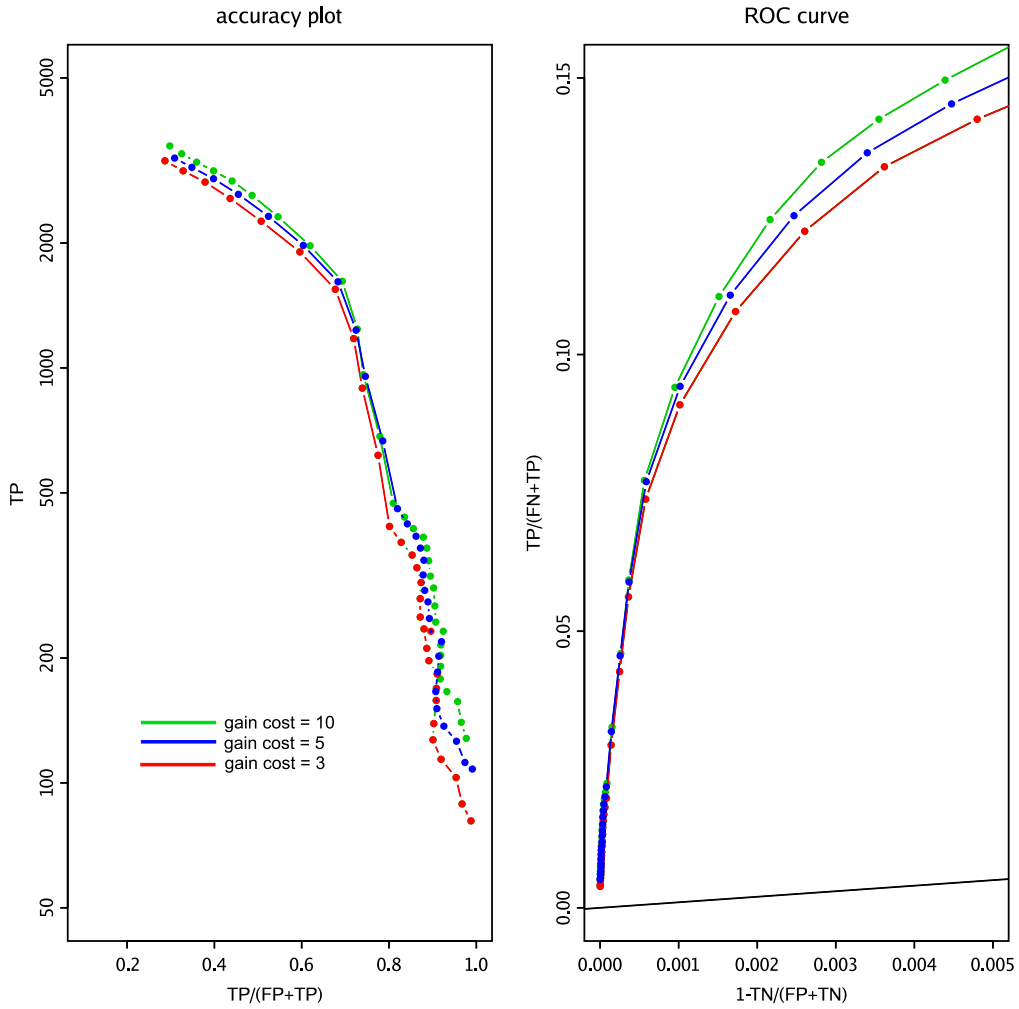

Figure 2: **Comparison of different gain costs.** The picture shows the prediction performance for different gain penalties with the sign score.

symptom of higher reconstruction accuracy. As seen in Table 1, reconstructions with higher cost increase the size of LUCA and decrease the number of horizontal transfer events (HGT) to extremely unrealistic levels. In fact, the reconstruction with gain cost 10 wipes out almost all HGT event from the reconstruction. Therefore, accepting such a reconstruction as more accurate equals saying that HGT does not exist.

In fact there is a simpler explanation. The better prediction performance reflects that there is a better way to score the presence-absence information using a tree. What the sign score with high gain penalty does is similar to measuring the similarity in COG species distribution by finding maximum subtrees where one or both COGs are absent. One can emulate the scores obtained with high gain penalty by finding these maximum subtrees and performing the following calculation (see Fig S3 for an illustration.):

score = - number of maxsubtrees with only 1 COG
foreach: *maximum empty subtree*
if: *sibling subtree contains only 1 COG*
$score = score - 1$
else:
$score = score + 1$
endfor

This is actually a way of measuring the structural similarity of the COG species distribution. It identifies those parts of the tree that reveal coordinated absence and those revealing independent absence. Notice that this is independent of the ancestor reconstruction. It is a different way to interpret the information contained on the COG-species distribution and the phylogenetic tree.

There are least two reasons why this is a better way to measure the similarity of presence-absence patterns of characters related by a tree: a) two COGs occurring only in the same two distant lin-

Supplementary Figure S3.

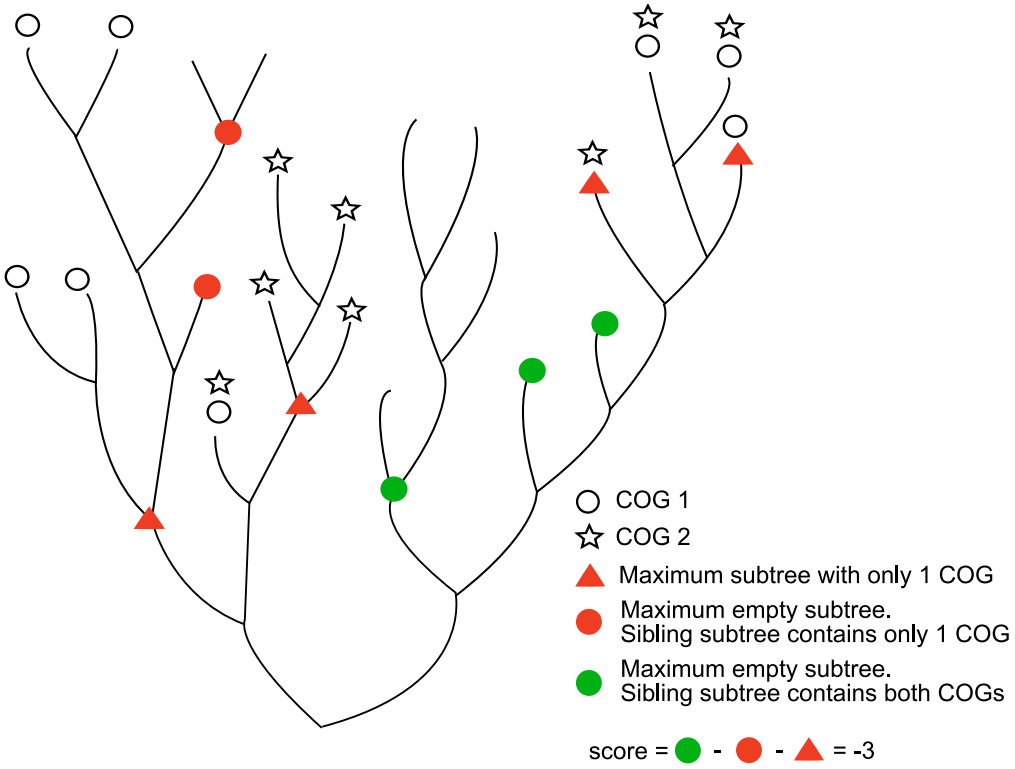

Figure 3: **Gain-loss scoring scheme based on maximum subtrees.** if both COGs are created in the same branch, the sign score with high gain penalty equals 1 + the score produced by this algorithm. In this formalism a leave is also a maximum subtree.

eages may get a low sign score if their gains are not exactly coupled. The score based on maximum subtrees, however, will get higher as the distance between these two lineages increases, because intermediate clades will get positively scored for their lack of both families. b) The maximum subtree based score will be proportional to the depth of the clade where the gain occurred, which means that gains happening in peripheral clades, such as Enterobacteria, are considered more significant than those occurring in, for example, the ancestor of all bacteria.

An example that illustrates this is given by COGs COG2037 and COG3252. Both COGs are exclusively archaeal, except for an HGT in *Rhodopirellula baltica*. With gain cost 3, the sign score on the gain-loss reconstruction gives a score of 2 for these COGs. The maximum tree variant gives instead a score of 9, highlighting the remarkable cooccurrence of these two families in two distant and deep lineages.

A number of possible variants to the above-mentioned scheme are possible. However, a broader analysis of alternative scoring techniques is out of the scope of this article and is left for future research.

S4. Distinguishing gains from duplications.

The sign score mentioned in the main text does not distinguish between gain-gain coocurrence and gain-duplication coocurrence, since both represent an increment in family size. We have implemented a variant in which only gain-gain and duplication-duplication events contribute positively to the score, finding that the results are exactly the same to our sign score as presented in the main text. This is also the case even when gain-duplication occurrences decrease the score.

S5. Degree distribution of coevolution networks.

Supplementary Fig S4 shows the degree distributions for coevolution networks built with partial correlation and with sign score for different thresholds.

S6. Best fit to degree distribution.

We have fitted the cumulative degree distribution of the coevolution network to an exponential,

Supplementary Figure S4.

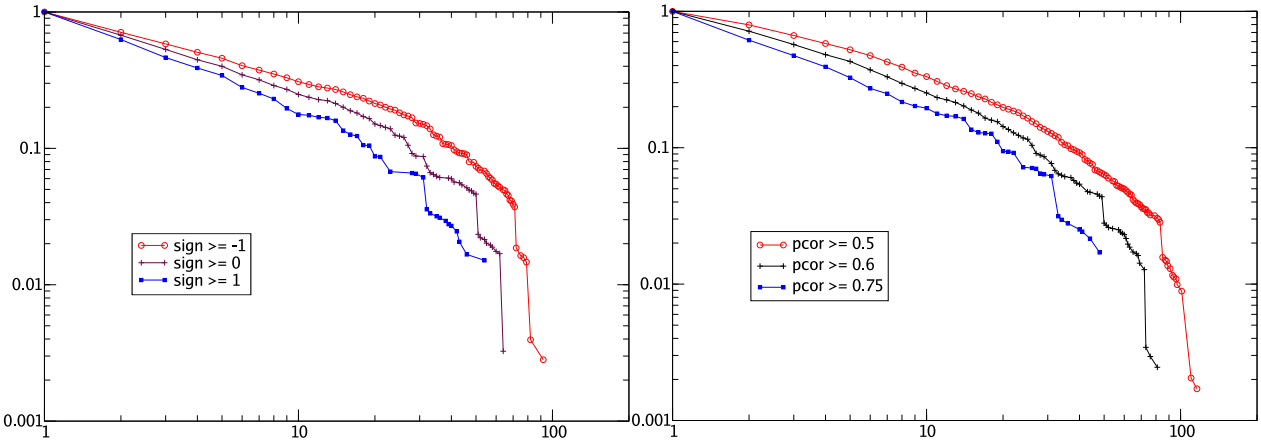

Figure 4: **Degree distribution of coevolution network.** The left panel shows the degree distribution for different thresholds for the sign score based coevolution network. The right panel shows this for the partial correlation based coevolution networks.

Supplementary Figure S5.

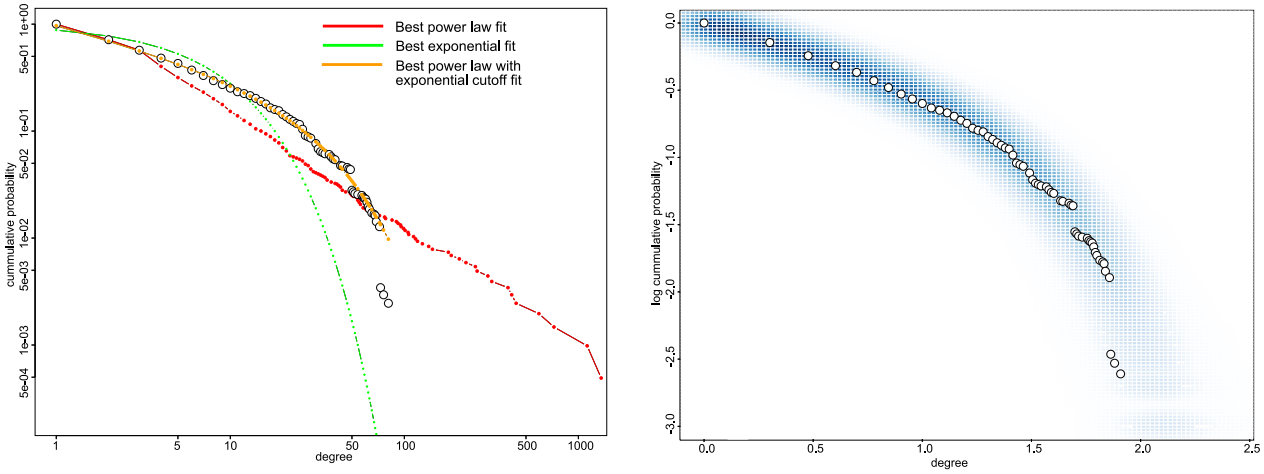

Figure 5: **Best fits to coevolution network degree distribution.** The left panel shows the cumulative degree distribution (pcor threshold = 0.6) and the best fit for different distributions. We see that a power law with exponential cutoff describes the data better than the other alternatives. The right panel shows the cumulative degree distribution in comparison with 100 distributions obtained by random sampling from the best fit power law with exponential cutoff. The smoothed 2D histogram shows the distribution of the synthetic data.

power law and power law with exponential cutoff distribution. The power law fit was done as described in Clauset *et al.* (2007); the other two where done by least square minimization using R's function 'nls'. As seen in Fig S5 the power law with exponential cutoff describes the data better than the rest. This is confirmed by generating 100 distributions sampled from the derivative of the fitted cumulative. We use the method described in Clauset *et al.* (2007) to generate random numbers from a power law with cutoff.

S7. Alternative benchmarks.

In our main text we use the KO based KEGG network to benchmark function prediction. KO is an scheme of ortholog identification based on computational analysis and manual curation which can be mapped to COGs. The mapping is available at the KEGG database website. The KEGG based COG network is constructed by establishing a link between two COGs when their corresponding KOs participate in the same pathway. The resulting network contains 1137 COGs and 20943 links. 72% of these COGs are mapped by a 1-1 mapping to a KO, and 16% by a 2-1 or 1-2 mapping. Alternatively, one can build the COG network directly from the cooccurrence of COG members in KEGG pathways, which results in larger coverage of COGs by the addition of less 'significant' links (1933 COGs and 78874 links). Supplementary Fig 5 show that the results based on this latter benchmark are consistent with those based on the KO KEGG network.

We see that for low percentage of true positives the sign score losses its high performance, while the partial correlation keeps a high accuracy. This could be caused by the inclusion of extra COGs where a filtering of large genome expansions and contractions is needed.

S8. Histories with large expansion and contraction events.

Supplementary Fig. 6 shows in detail the histories of COGs involved in the cluster of ABC transporters.

Supplementary Figure S6.

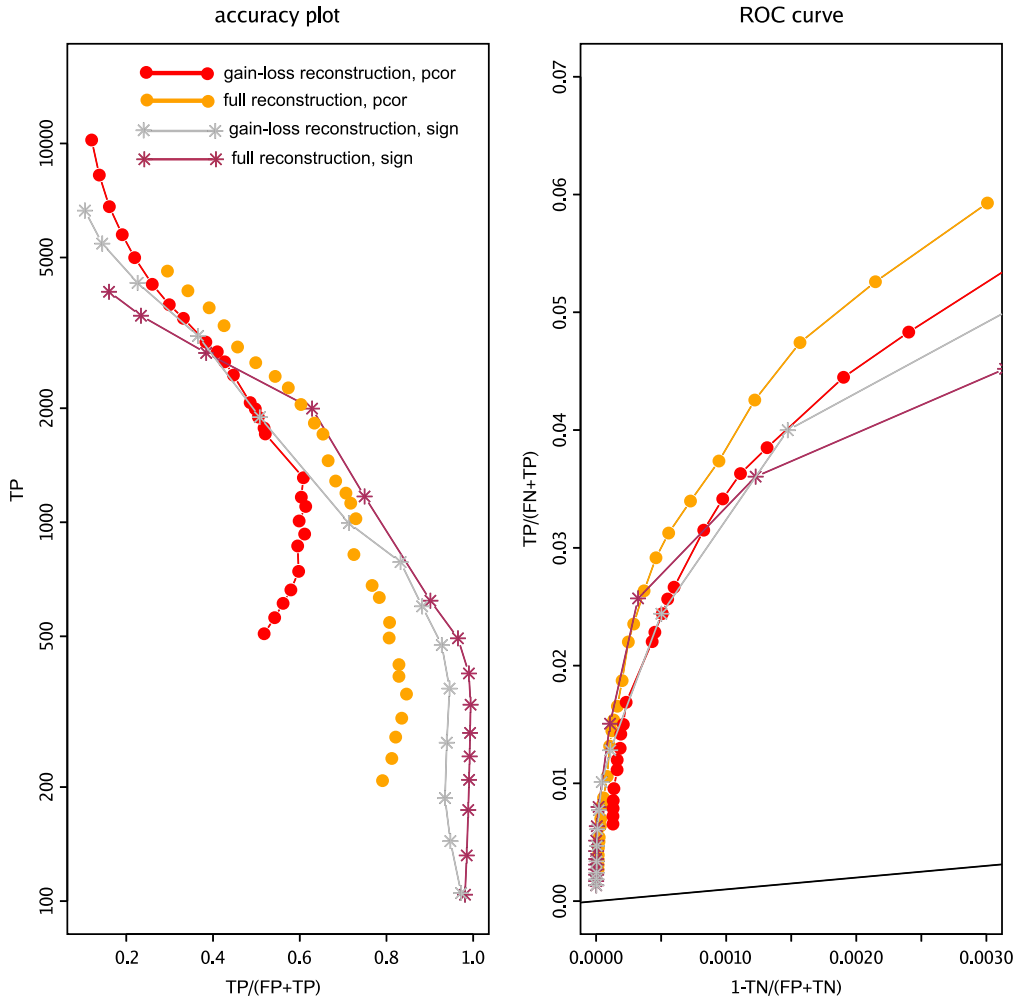

Figure 6: **Prediction performance with alternative KEGG benchmark.** Accuracy-Coverage plot and ROC curve based on gene based KEGG network.

Supplementary Figure S7.

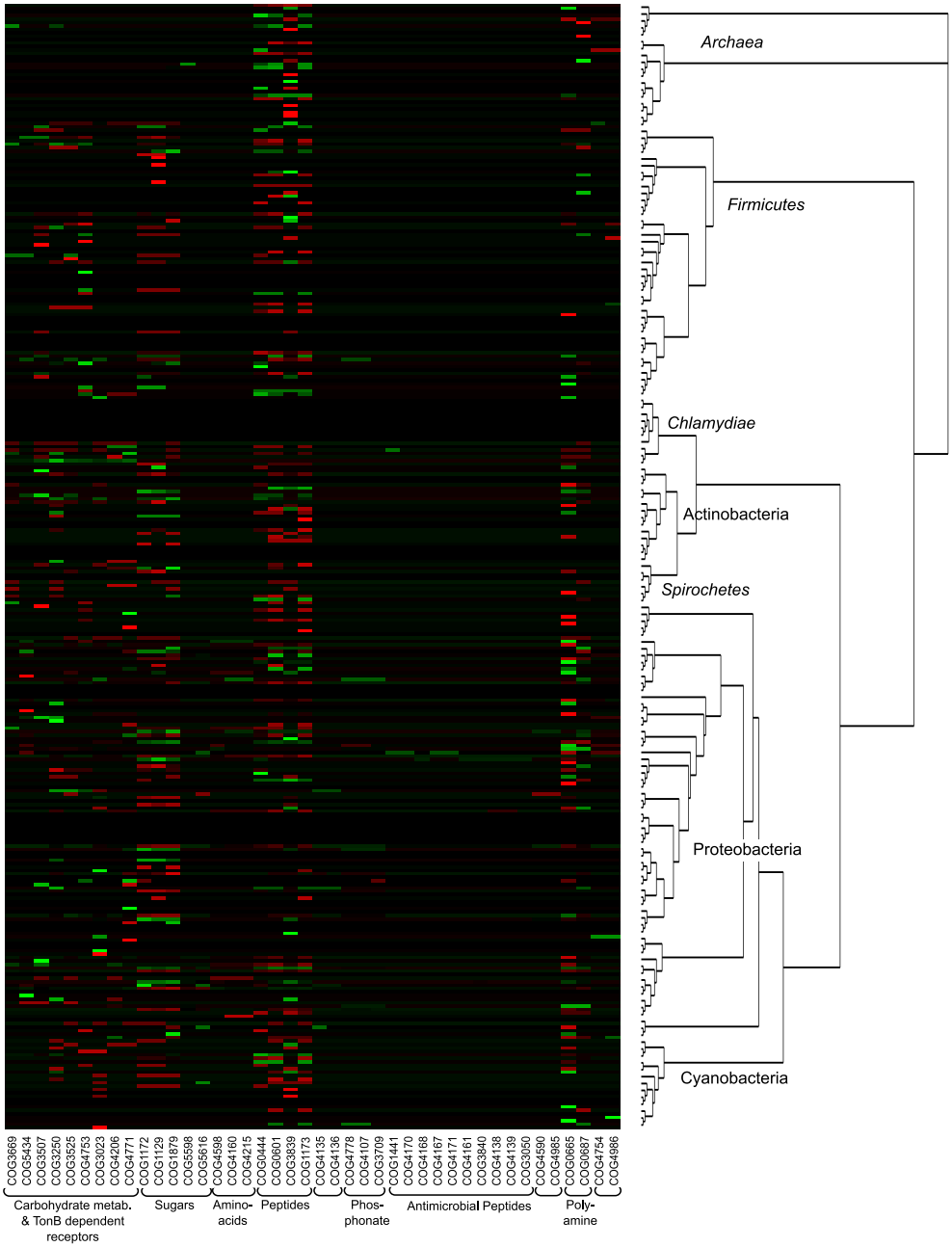

Figure 7: **Evolutionary history of ABC transport families.** The heat map shows the duplications and deletions per branch on the whole tree, normalized per row. We see that some groups of families have similar histories of events and that the high correlation is not caused by one concerted gain but by many gene duplication and deletion events. Groups without label contain uncharacterized COGs. COG3839 is shared by sugars and peptides clusters. The colors show the intensity of duplications (red) and deletions (green). In the heatmap, inner branches are placed in between their daughter subtrees.

S9. Supplementary data.

Reconstructed histories, species list and coevolution maps are available online at http://bioinformatics.bio.uu.nl/otto/coevolution

References

Clauset, A., Shalizi, C., and Newman, M., 2007. Power-law distributions in empirical data. *arXiv: 0706.1062* .

Mirkin, B. G., Fenner, T. I., Galperin, M. Y., and Koonin, E. V., 2003. Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. *BMC. Evol. Biol.* **3**:2.

Swofford, D., 1998. *Phylogenetic Analysis Using Parsimony (PAUP), Version 4.0b10*. Sinauer, Sunderland, MA.