SUPPORTING TEXT.

Inference of gains and presence of genes on branches of the tree.

To estimate the probability that specific genes were gained or present on each branch of the tree, we chose a simple heuristic, based on the joint probability of the states of the ancestor and descendant nodes (Methods). We chose this approach because we are not concerned with *any* gain, but rather with gains that are retained until the end of a branch. For example, any gain at all is to be expected at some rate more or less without regard to genome content of the host, due to phage infection or DNA in the environment. However, given that the vast majority of these gains are followed closely by losses (Baltrus 2013), they are not as biologically interesting as genes gained and retained adaptively, and they are also mostly unobserved. Additionally, our approach allows us to consider the probability of steady presence across a branch. Furthermore, our approach considers the average reconstruction at each node to compute the probability of gain or presence of genes on branches, rather than summing across each possible reconstructed scenario in the stochastic mapping procedure (for instance weighted by the likelihood of each possible scenario). While using all possible mappings could, in principle, reduce the numerical error of our probability estimates, it would entail an onerous and potentially intractable computation. Moreover, the biological (Figure 2) and statistical (Figures 5, S8) validations we have performed suggest that our results are robust.

Our method of inferring gains is also different from the probabilities of gains (or, similarly, the expected number of gains) that are computed by the *gainLoss* software (Cohen and Pupko 2010), using a previously-developed continuous-time Markov chain (CTMC) model to count the number of gains on each branch (Minin and Suchard 2008). These models solve the problem of counting the number of one-way transitions between two states (say, presence and absence) given transition rates, states at the start and end of the interval, and a set amount of time in the interval. Thus, the CTMC implemented in *gainLoss* is capable of estimating the expected number of gains of a given gene on a given branch, with knowledge of gain and loss rates. However, this approach can lead to problematic cases in which a gene can be absent in ancestor and descendant nodes, and yet, given a very long branch, is inferred to be gained on this branch. While such scenarios may have statistical support, in practice they are very hard to interpret and compare to other events that more obviously support a gain. Given the presence of Archaea in our phylogeny, which are a dramatically divergent outgroup, this was a cause for concern.

Indeed, the CTMC estimated that the median gene was gained more than twice along the long branch connecting Archaea to Bacteria, with some genes gained more than 10 times on this branch alone (data not shown). This result is almost certainly artefactual, but has the potential to substantially skew the overall appraisal of gains for a given gene. This problem is probably exacerbated by overfitting, given that *gainLoss* assigns each gene a potentially unique mixture of gain and loss rates, in addition to re-estimating the branch lengths of the tree. All of these parameters are used in computing an expected number of gains for a gene on a branch in addition to the reconstruction. It is possible that such methods are inappropriate for large phylogenies with highly heterogeneous branch lengths. For these reasons and those stated above, we chose to ignore the *gainLoss* CTMC estimates in favor of the less sophisticated but more interpretable gain/presence inference method described above.

Simulation of gene gain/loss evolution.

Previous attempts to use the *gainLoss* software to make inferences about horizontal gene transfer and detect coevolution used a parametric bootstrapping approach, simulating the evolution of genes to obtain null expectations for testing hypotheses (Cohen et al. 2011, 2012). While the use of exact parametric methods to estimate this null distribution is possible in principle (Maddison 1990), these methods rely upon a single binary reconstruction of ancestral states. Clearly, our probabilistic reconstruction is unsuited for such an analysis. Again, one could in principle enumerate all possible reconstructions, and estimate the null distribution exactly as a weighted sum across each reconstructions, but developing this method for large trees lies outside the scope of this paper.

In our simulations, we therefore followed the example of others with certain modifications. The simulation procedure implemented in the *gainLoss* program was too memory-intensive to be feasible for a sufficiently large number of genes. Consequently, we took the gain and loss rates inferred by *gainLoss* for the real genes and used their distribution to simulate the evolution of genes using the function *rTraitDisc()* in the APE library. Briefly, we fit gamma distributions to the rates of gain and the rates of loss across all genes, and used the resulting parameters to define sampling distributions for gain and loss rates of simulated genes (see Methods). We found that using these distributions inferred relatively few gains compared to the gains of observed genes (compare Figure S1A and Fig S1C). We speculated that the rate mixture

model employed by *gainLoss* has difficulties accommodating the upper tail of the distribution of gain rates (roughly, those genes gained >50 times in this tree), given that the vast majority of genes are gained relatively few times (Fig S1A). Consequently, we adjusted the shape parameters of the gain and loss rate distributions heuristically to find values that gave distributions of simulated gains that included genes that are gained sufficiently many times. We found that multiplying the shape parameter of the gain rate by 3 and the shape parameter of the loss rate by 1.5 gave reasonably wide distributions of gains among simulated genes (Fig S1E). It is important to note that the shape of the distribution from which rates are drawn does not affect the simulated evolution of a given gene with single sampled gain and loss rates. Furthermore, because we are not using the entire distribution of simulated genes but only those most appropriate to each gene as a null distribution, any differences in the distributions of gain counts between simulated and real genes are unlikely to affect results.

Power of the PGCE detection method.

One of our observations is that there are weak relationships between the prevalence of a gene, how often it is gained, and its in- and out-degrees in the PGCE network (Fig S4). Given that these values define the null distributions that we use to infer PGCEs, it was possible that our analyses are less sensitive for certain values of these parameters. We considered to what extent a lack of power was affecting our results with a simple power analysis. For genes i and j, the maximum observable value C_{ij} counting the gains of j in the presence of i is $\min(p_i, g_j)$, representing respectively the prevalence of gene i and the number of gains of gene j. For a range of values of these parameters (p_i, g_j), we compared this maximum potential observation to the null distribution from parametric bootstrapping appropriate to these parameter values. This represents the most extreme possible test statistic between the two genes for these parameter values, so in each case the null hypothesis should be rejected if there is sufficient power. We found that power varied substantially across various values of (p_i, g_j) (Figure S2A). Specifically, we were incapable of detecting associations for any combination involving the most-prevalent genes or the least-gained genes. This is unsurprising, given that noise is expected to be high for the former, and signal to be low for the latter. Considering our observed distribution of p-values (Figure S2B), we find the expected spike in frequency near p = 0 (indicating true positive dependencies), but also an unexpected spike in frequency near p = 1, indicating that our

parametric bootstrapping test is underpowered due to the sparsity of gains, as suggested by power analysis (Fig S2A). Consequently, there are likely to be many more PGCEs than we detect in this study. Notably, if we relax our FDR threshold from 1% to 5% in inferring PGCEs, we increase the raw number of edges in our network more than ten-fold (from 8,415 to 86,719). We chose to proceed with the more stringent threshold to focus on the most confident PGCEs, but we use this example to highlight the very large potential for PGCEs structuring genome evolution in prokaryotes.

Processing and analysis of the PGCE network.

After inferring a PGCE network, we post-processed this network to both ease further analysis and to remove potentially spurious edges. First, we removed edges such that the network became a directed acyclic graph (DAG). DAGs are relatively easy to analyze and interpret topologically. We found only one cycle-inducing edge: an obviously spurious self-edge (for gene K07218). The absence of non-spurious cycles may be initially surprising, but can be explained by the relatively small number of genes with in-edges (less than one-third of genes in the network) and the anti-correlation of in-degree and out-degree across genes (Fig S4E). To evaluate whether the lack of cycles is attributable to degree distribution, we randomly rewired the DAG five times while preserving degree distribution, and in each of these five cases the result was still a DAG. This analysis indicates that this acyclic topology is a simple consequence of degree distribution, rather than a biological property of specific PGCE relationships. Together, these results indicate that few cycles are expected for a network with such properties. However, one might still expect some number of true cycles from a biological point of view, even if the network itself is biased against them. We believe that such cycles likely exist, but we are not detecting them because of our relatively low power, and the stringency of our threshold for assigning edges (Fig S2, see above section).

Next, we removed potentially spurious edges in the network that might have been introduced by indirect transitive effects. For example, if gene A encourages the gain of gene B, and gene B encourages the gain of gene C (A→B→C), we might also infer that there is a direct A→C PGCE, even if such a PGCE does not actually exist. Consequently, we performed a transitive reduction of our DAG to obtain a "minimal equivalent graph" (Hsu 1975), or a DAG with all potentially indirect interactions (such as the A→C example above) removed. While

potentially removing true PGCEs, we thus enrich our PGCE network for the most confident interactions. This procedure removed 186 potentially indirect PGCEs. It is this DAG, with all cycles and indirect edges removed, that we used for all downstream analyses.

The degree distributions for this network indicated that a slight majority of genes (nodes) are disconnected, and we omitted these genes from further analyses. Furthermore, the distribution of in-degrees was more unequal than that of out-degrees across nodes (Fig S4A, S4B). The degree distributions showed weak relationships with the prevalence and gain count of genes, but these do not appear to be primary determinants of network structure (Fig S4C, S4D).

Dependencies among pathways.

The *urtA-rbsL* PGCE (Figure 3) highlighted the potential importance of inter-pathway PGCE dependencies. To understand the structure of such pathway-pathway dependencies, we tested for associations between genetic pathways within the PGCE network, compared to a null distribution of rewired networks. We detected 93 pathway-pathway dependencies (each $p < 0.001$, compared to the rewired null distribution), which we modeled as a directed network among 65 pathways (Figure S6). Unlike the PGCE network, the pathway-pathway dependency network has many cycles. Related pathways showed many dependencies and clustered with each other, most strikingly for the metabolism of aromatic compounds. Consequently, we expect that PGCE dependencies, rather than only representing one-to-one interactions between genes, also reflect functional relationships between whole genetic pathways.

Algorithms.

Feedback arc set (FAS) identification algorithm (Hausmann and Korte 1978; Hassin and Rubinstein 1994)**.**

1) Start with an empty DAG and an empty FAS;
2) Select a random edge *E* from our PGCE network, add it to the DAG;
3) If adding *E* to the graph adds a cycle, remove *E* again and add it to the FAS, else accept *E* in the DAG;
4) If there are more edges that are neither in the DAG nor in the FAS, go to 2

Transitive reduction of a DAG algorithm (Hsu 1975)**.**

1) Convert the network into an adjacency matrix representation;

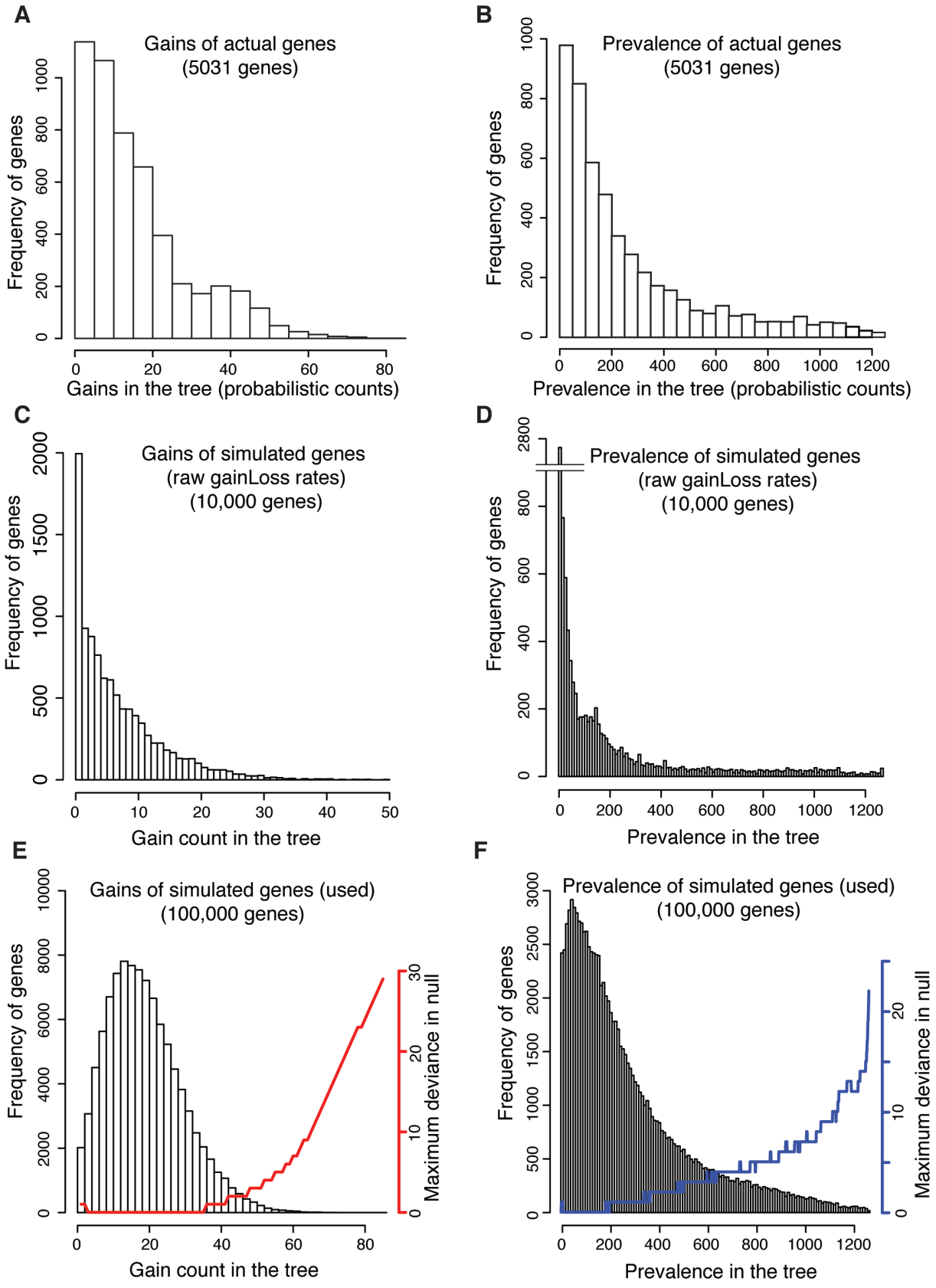
2) Convert the adjacency matrix into a path matrix;
3) Remove all edges in the path matrix that can be explained by other paths, by iterating over all groups of 3 nodes.

Topological sort with grouping algorithm (Knuth 1973)**.**

We used the following procedure to perform a topological sort of a DAG:

(1) Initialize the rank count with "rank" = 1;
(2) Identify the set of nodes in the DAG with in-degree = 0 (these occupy the first position in a sort);
(3) Label these nodes with the current "rank" (1 in the first step);
(4) Remove these nodes and their edges from the DAG (some new nodes will now have in-degree = 0;
(5) if there are still nodes in the DAG, increment "rank" by 1 and go to step 2.

The resulting labeled groups constitute the ordered ranks of the topological sort.


A
Gains of actual genes
(5031 genes)
Frequency of genes
Gains in the tree (probabilistic counts)
B
Prevalence of actual genes
(5031 genes)
Frequency of genes
Prevalence in the tree (probabilistic counts)
C
Gains of simulated genes
(raw gainLoss rates)
(10,000 genes)
Frequency of genes
Gain count in the tree
D
Prevalence of simulated genes
(raw gainLoss rates)
(10,000 genes)
Frequency of genes
Prevalence in the tree
E
Gains of simulated genes (used)
(100,000 genes)
Frequency of genes
Maximum deviance in null
Gain count in the tree
F
Prevalence of simulated genes (used)
(100,000 genes)
Frequency of genes
Maximum deviance in null
Prevalence in the tree

1

Figure S1. Distributions of total gains (A) and prevalence (B) estimated for empirical genes by the *gainLoss* program. *gainLoss* rate estimates lead to underestimation of gains (C) and prevalence (D) in the tree: gene gain counts across 10^4 genes simulated according to gain/loss rates directly estimated by *gainLoss* for empirical genes. Gene gain (E) and prevalence (F) counts across genes simulated for use in null distributions. Red (gain) and blue (prevalence) line plots indicate, for each value of gain count or prevalence, the absolute difference of the least similar gene in its null distribution from that value (maximum deviance). For instance, in (E), a gene with 40 gains will be compared to a null distribution of simulated genes with as few as 39 gains and as many as 41 gains (deviance of one). Relative to (A) and (B), parameters of the underlying distributions of gain and loss rates were heuristically adjusted to provide acceptable coverage of the gain/prevalence values observed for empirical genes in (E) and (F).

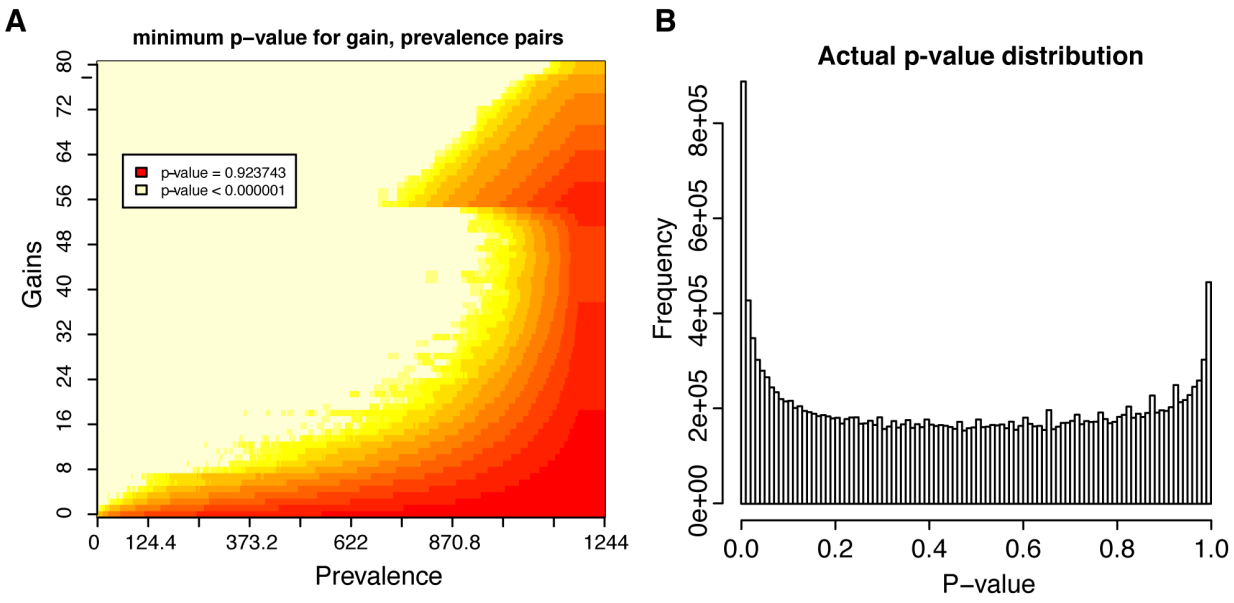


Figure S2. (A) Power analysis of the parametric bootstrapping hypothesis test for detecting PGCEs. X and Y axes represent, respectively, total prevalence and total gains for a hypothetical pair of genes with a strong PGCE (maximum observable test statistic). Colors represent the (log10-scaled) minimum possible p-value that can be attained for such a gene pair using the relevant null distribution of simulated genes. Areas that are not white/pale yellow are underpowered for detecting PGCEs. (B) The distribution of empirical p-values observed for testing hypotheses of no PGCE in the evolution of pairs of genes, according to parametric bootstrapping. The spike at p = 1.0 in (B) indicates that sparsity in the data detracts from power, as predicted in (A), even after filtering pairs of genes with C_{ij} <= 1.

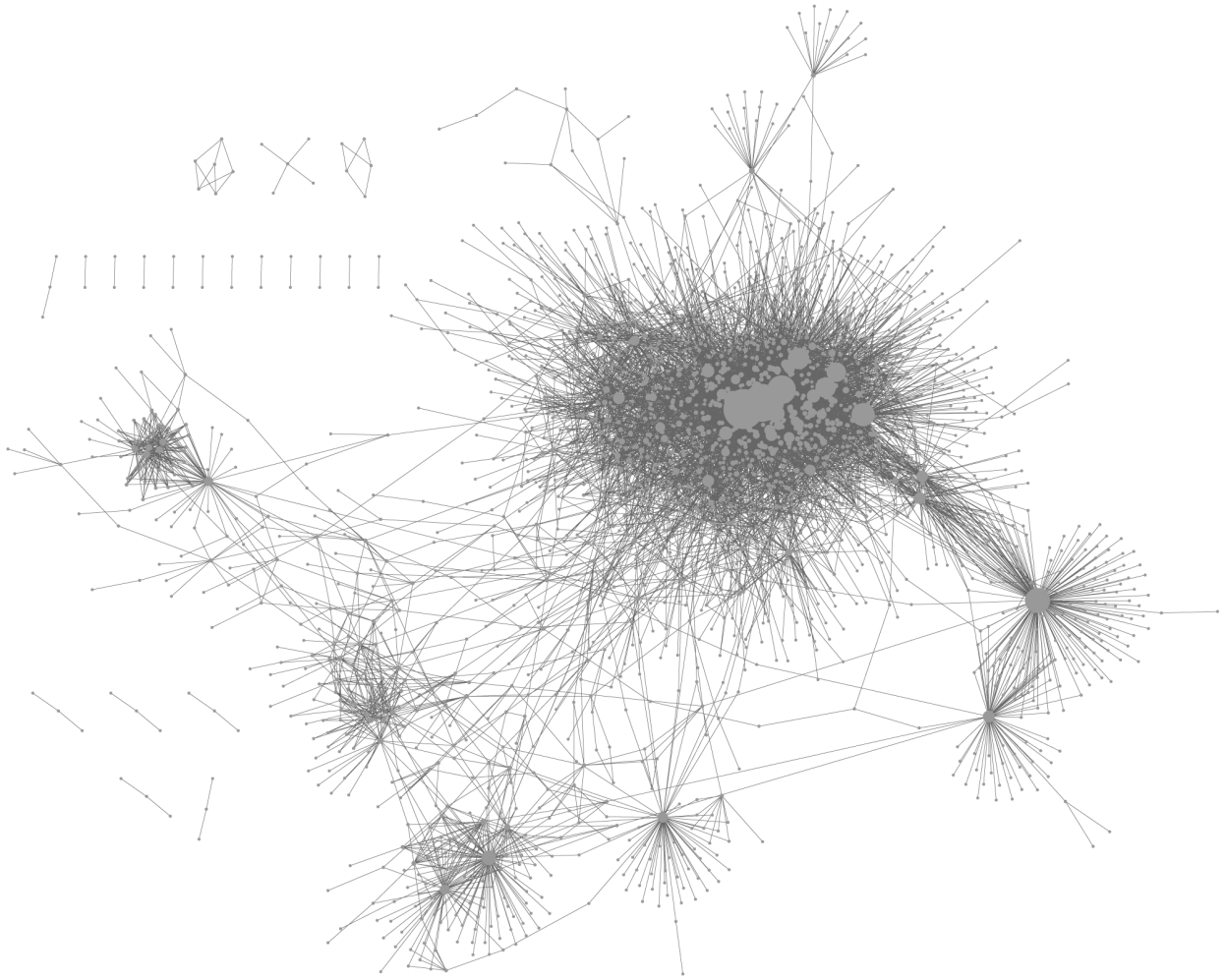


Figure S3. A global network of directional dependencies between prokaryotic genes (PGCEs). Node size is scaled to total edge count for each node (and see also Figure S4).

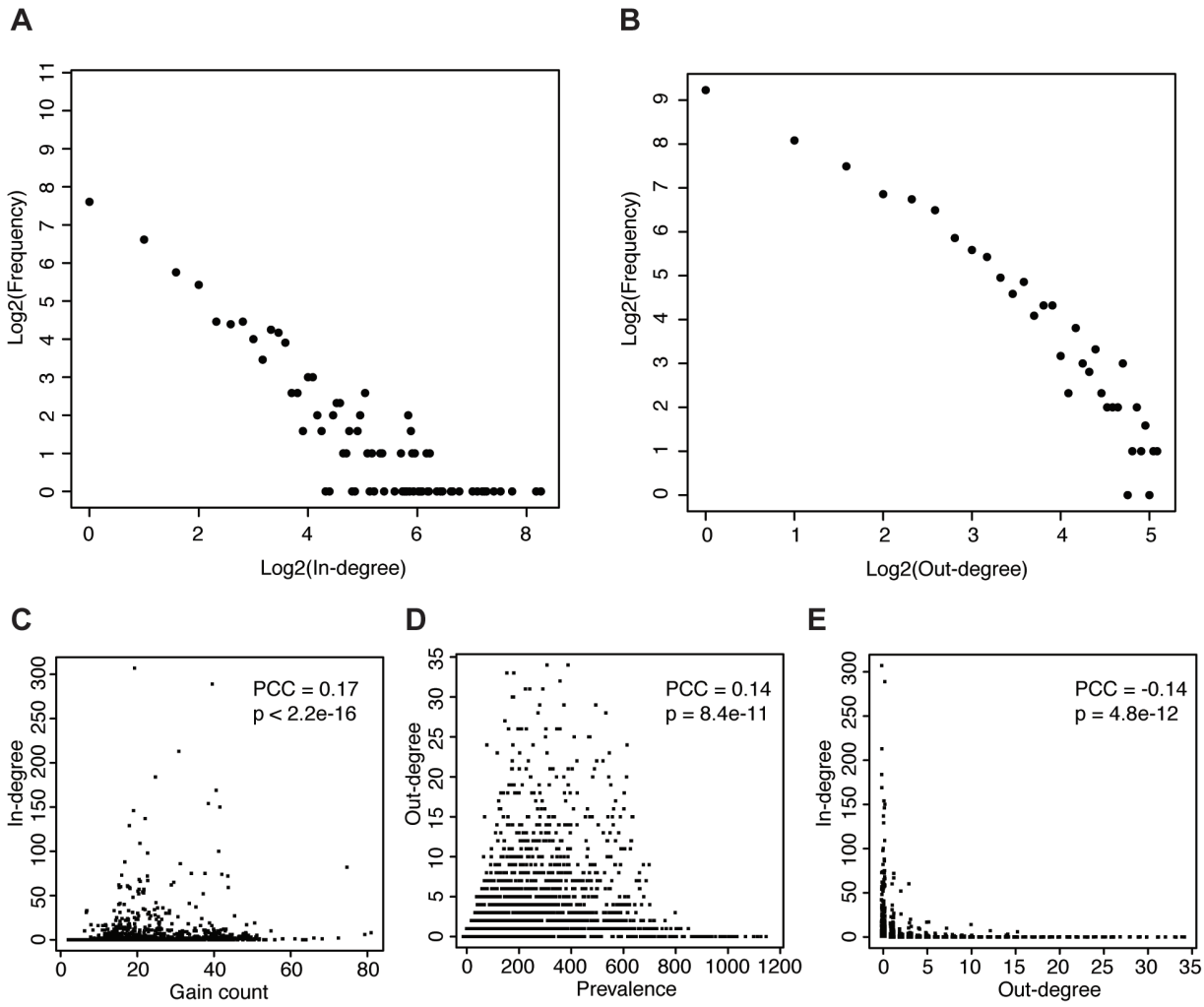


Figure S4. (A) Out-degree distributions of the final PGCE network (nodes with out-degree equal to zero are omitted). (B) In-degree distributions of the final PGCE network (nodes with in-degree equal to zero are omitted). (C-E): Prevalence and gain counts of genes only weakly affect their PGCEs. The degrees of each gene (node) in the PGCE network are plotted against its prevalence (C) and counted gains (D) throughout the tree, and the degrees are plotted against each other (E). Pearson correlations between the plotted variables are indicated above each plot. PCC = Pearson correlation coefficient, p-value is from a correlation test.

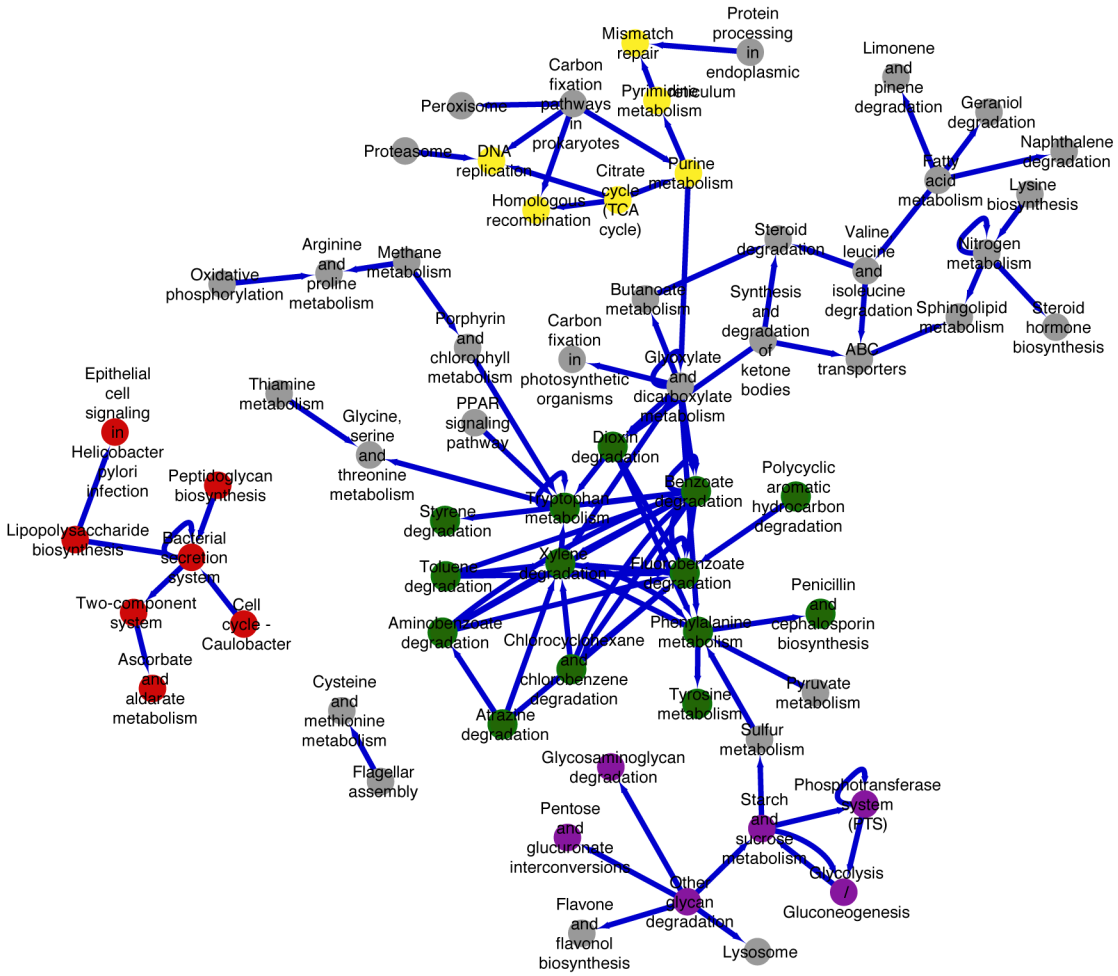


Figure S5. A network of evolutionary dependencies between functional pathways. Overall structure of the evolutionary pathway-pathway dependency network. Directed edges indicate that the source pathway and the sink pathway are connected by more PGCEs between individual genes in those pathways than expected from a rewired null distribution ($p < 0.001$). Colors indicate selected pathway clusters of similar functions (green: aromatic compound secondary metabolism; red: pathogenesis; purple: carbohydrate metabolism; yellow: DNA metabolism).

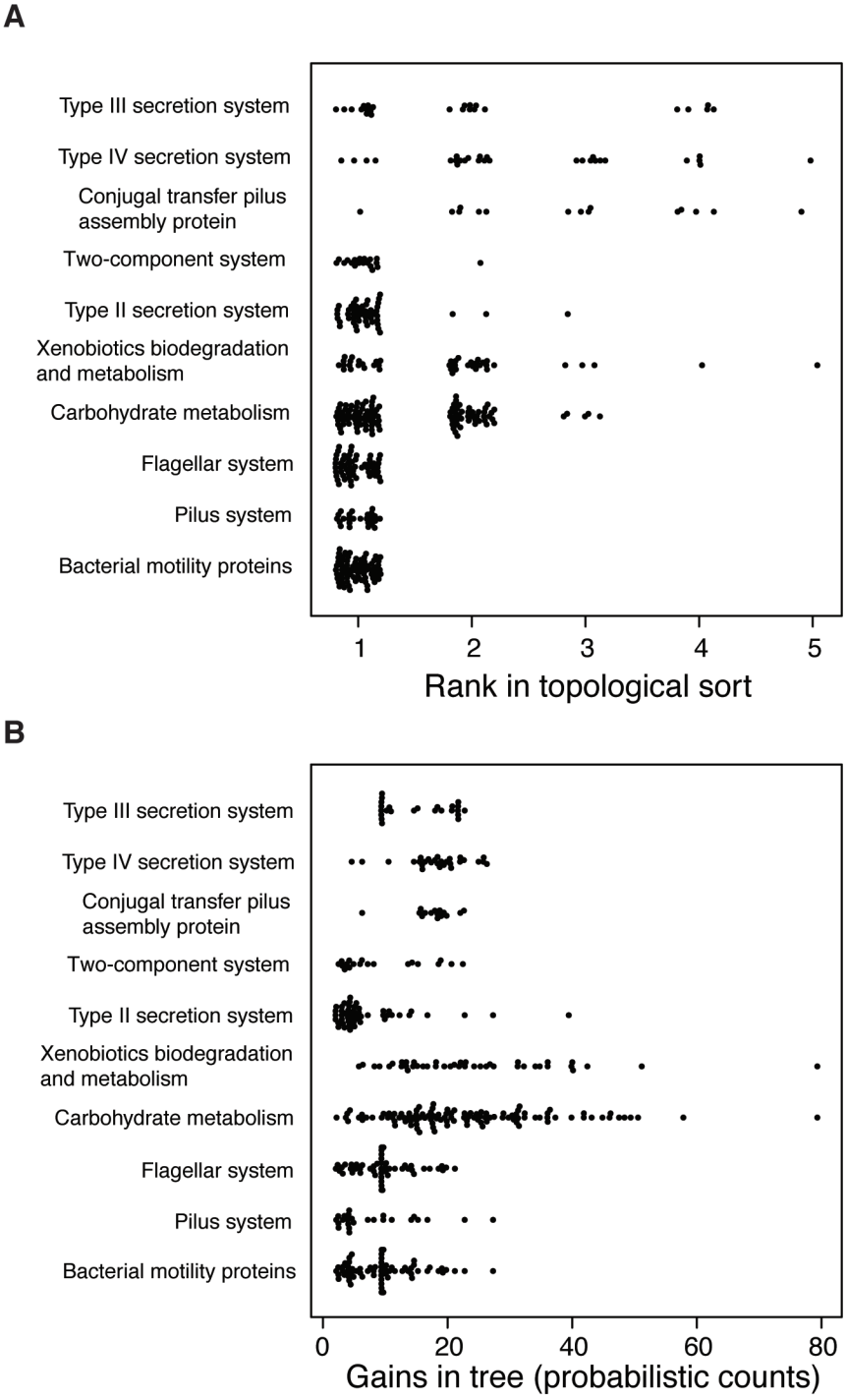


Figure S6. Differences in gain counts do not explain differential sorting of genes in different functional groups. (A): Variation in ranks of the sort across functional categories. (B): Total branches in which gains have occurred ("gains in tree") across genes in various functional categories that are differentially ranked in a topological sort of the PGCE network. Note that the categories with the highest average gain (Carbohydrate and Xenobiotics metabolism) are ranked in the middle of the sort. See Table 1.

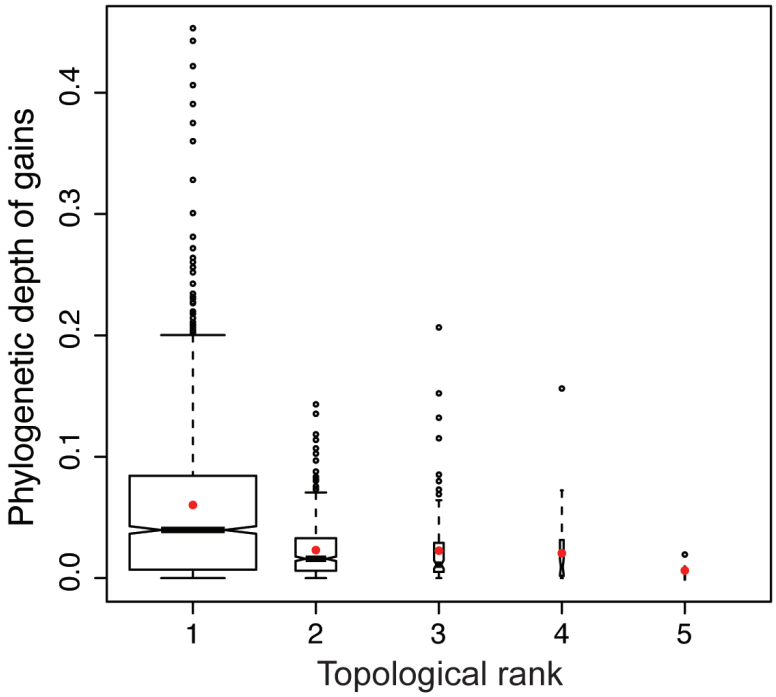


Figure S7. Phylogenetic depth of gene gains in bacteria decreases with rank in the topological sort. Phylogenetic depth of the gains of genes are weakly negatively correlated with their ranks in the sort (Spearman's $r = -0.24$, $p < 2.2 \times 10^{-16}$). For each rank, we plot the distribution of the phylogenetic depths (distance of gain branch from root) of the average depth of confident gains ($Pr(gain) > 0.6$) of each gene in that rank. The mean of each distribution is plotted as a red point. Branches leading to Archaea and archaeal genomes are omitted from the analysis. Boxplot widths are scaled to the number of genes in each rank of the sort. The tree was converted to an ultrametric tree for the purpose of this analysis (the root is separated from all tips by a total branch length of 1.0).

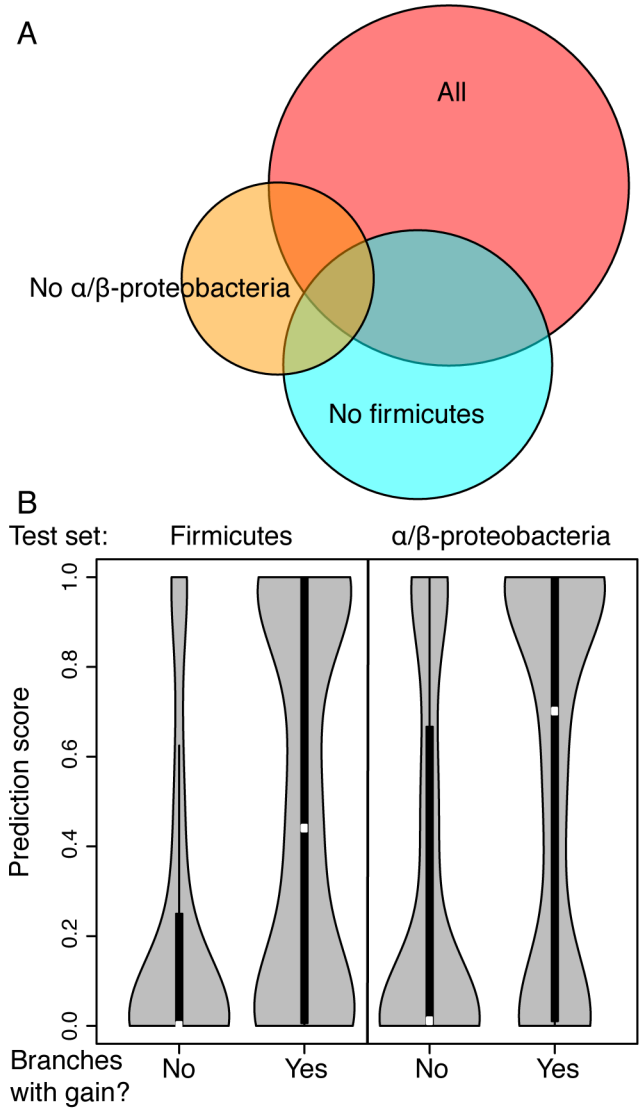


Figure S8. (A) Overlap of edges in PGCE networks inferred from different subsets of the data. See also Table S4. All overlaps are highly statistically significant ($p < 2.2E-16$, hypergeometric test). (B) Distribution of prediction scores for gene acquisition on each branch in the test set clades. Branches with a gain ($Pr(gain) > 0.5$)) have a higher score than branches without a gain ($Pr(gain) < 0.5$) for predictable genes ($p < 2.2E-16$ for each, U-test). Predictable genes are the affected genes in at least one PGCE, i.e. they have at least one in-edge in the trained PGCE model. Violin plots show density of each distribution, with an inset boxplot (white box is median of distribution). Each violin plot shows the distribution of prediction scores for branches in one test set for one category (gene gained/gene not gained).

Table S1. Genes which influence the gain of *rbsS*, gene encoding the RuBisCO small chain.

KEGG Orthology (KO)	Description
K02584	Nif-specific regulatory protein
K06139	pyrroloquinoline quinone biosynthesis protein E
K06138	pyrroloquinoline quinone biosynthesis protein D
K06137	pyrroloquinoline-quinone synthase [EC:1.3.3.11]
K06136	pyrroloquinoline quinone biosynthesis protein B
K09165	hypothetical protein
K03809	Trp repressor binding protein
K13483	xanthine dehydrogenase YagT iron-sulfur-binding subunit
K13481	xanthine dehydrogenase small subunit [EC:1.17.1.4]
K02448	nitric oxide reductase NorD protein
K02597	nitrogen fixation protein NifZ
K02596	nitrogen fixation protein NifX
K02595	nitrogenase-stabilizing/protective protein
K02593	nitrogen fixation protein NifT
K02592	nitrogenase molybdenum-iron protein NifN
K02022	HlyD family secretion protein
K11811	arsenical resistance protein ArsH
K08973	putative membrane protein
K12511	tight adherence protein C
K08995	putative membrane protein
K07506	AraC family transcriptional regulator
K10778	AraC family transcriptional regulator, regulatory protein of adaptative response / methylated-DNA-[protein]-cysteine methyltransferase [EC:2.1.1.63]
K07165	transmembrane sensor
K07161	NA
K00830	alanine-glyoxylate transaminase / serine-glyoxylate transaminase / serine-pyruvate transaminase [EC:2.6.1.44 2.6.1.45 2.6.1.51]
K01266	D-aminopeptidase [EC:3.4.11.19]
K05559	multicomponent K+:H+ antiporter subunit A
K02278	prepilin peptidase CpaA [EC:3.4.23.43]
K02279	pilus assembly protein CpaB
K02276	cytochrome c oxidase subunit III [EC:1.9.3.1]
K02274	cytochrome c oxidase subunit I [EC:1.9.3.1]
K02275	cytochrome c oxidase subunit II [EC:1.9.3.1]
K02305	nitric oxide reductase subunit C
K13924	two-component system, chemotaxis family, CheB/CheR fusion protein [EC:2.1.1.80 3.1.1.61]
K13926	ribosome-dependent ATPase
K09924	hypothetical protein
K10764	O-succinylhomoserine sulfhydrylase [EC:2.5.1.-]
K07157	NA

K03188	urease accessory protein
K01067	acetyl-CoA hydrolase [EC:3.1.2.1]
K01797	NA
K00824	D-alanine transaminase [EC:2.6.1.21]
K00685	arginine-tRNA-protein transferase [EC:2.3.2.8]
K09796	hypothetical protein
K11177	xanthine dehydrogenase YagR molybdenum-binding subunit [EC:1.17.1.4]
K11178	xanthine dehydrogenase YagS FAD-binding subunit [EC:1.17.1.4]
K00329	NADH dehydrogenase [EC:1.6.5.3]
K09008	hypothetical protein
K09005	hypothetical protein
K05563	multicomponent K+:H+ antiporter subunit F
K01800	maleylacetoacetate isomerase [EC:5.2.1.2]
K00253	isovaleryl-CoA dehydrogenase [EC:1.3.8.4]
K02258	cytochrome c oxidase assembly protein subunit 11
K11962	urea transport system ATP-binding protein
K11963	urea transport system ATP-binding protein
K11960	urea transport system permease protein
K11961	urea transport system permease protein
K05973	poly(3-hydroxybutyrate) depolymerase [EC:3.1.1.75]
K07102	NA
K00023	acetoacetyl-CoA reductase [EC:1.1.1.36]
K15866	2-(1,2-epoxy-1,2-dihydrophenyl)acetyl-CoA isomerase [EC:5.3.3.18]
K04561	nitric oxide reductase subunit B [EC:1.7.2.5]
K05564	multicomponent K+:H+ antiporter subunit G
K05562	multicomponent K+:H+ antiporter subunit E
K05561	multicomponent K+:H+ antiporter subunit D
K05560	multicomponent K+:H+ antiporter subunit C
K02533	tRNA/rRNA methyltransferase [EC:2.1.1.-]
K15011	two-component system, sensor histidine kinase RegB [EC:2.7.13.3]
K03200	type IV secretion system protein VirB5
K07303	isoquinoline 1-oxidoreductase, beta subunit [EC:1.3.99.16]
K07302	isoquinoline 1-oxidoreductase, alpha subunit [EC:1.3.99.16]
K07234	uncharacterized protein involved in response to NO
K00303	sarcosine oxidase, subunit beta [EC:1.5.3.1]
K02651	pilus assembly protein Flp/PilA
K01055	3-oxoadipate enol-lactonase [EC:3.1.1.24]
K02502	ATP phosphoribosyltransferase regulatory subunit
K03325	arsenite transporter, ACR3 family
K02225	cobalamin biosynthetic protein CobC
K01991	polysaccharide export outer membrane protein

K04748	nitric oxide reductase NorQ protein
K00304	sarcosine oxidase, subunit delta [EC:1.5.3.1]
K00305	sarcosine oxidase, subunit gamma [EC:1.5.3.1]
K01429	urease subunit beta [EC:3.5.1.5]
K05343	maltose alpha-D-glucosyltransferase/ alpha-amylase [EC:5.4.99.16 3.2.1.1]
K06044	(1->4)-alpha-D-glucan 1-alpha-D-glucosylmutase [EC:5.4.99.15]
K13766	methylglutaconyl-CoA hydratase [EC:4.2.1.18]
K01430	urease subunit gamma [EC:3.5.1.5]
K11959	urea transport system substrate-binding protein
K15012	two-component system, response regulator RegA
K00457	4-hydroxyphenylpyruvate dioxygenase [EC:1.13.11.27]
K00104	glycolate oxidase [EC:1.1.3.15]
K04756	alkyl hydroperoxide reductase subunit D
K03519	carbon-monoxide dehydrogenase medium subunit [EC:1.2.99.2]
K09983	hypothetical protein
K06995	NA
K00119	NA
K00449	protocatechuate 3,4-dioxygenase, beta subunit [EC:1.13.11.3]
K00114	alcohol dehydrogenase (cytochrome c) [EC:1.1.2.8]
K05524	ferredoxin
K02282	pilus assembly protein CpaE
K02280	pilus assembly protein CpaC
K03153	glycine oxidase [EC:1.4.3.19]
K09959	hypothetical protein
K00050	hydroxypyruvate reductase [EC:1.1.1.81]
K08738	cytochrome c
K07018	NA
K00126	formate dehydrogenase, delta subunit [EC:1.2.1.2]
K14161	protein ImuB
K11902	type VI secretion system protein ImpA
K07246	tartrate dehydrogenase/decarboxylase / D-malate dehydrogenase [EC:1.1.1.93 4.1.1.73 1.1.1.83]
K03198	type IV secretion system protein VirB3
K11472	glycolate oxidase FAD binding subunit
K11473	glycolate oxidase iron-sulfur subunit
K11475	GntR family transcriptional regulator, vanillate catabolism transcriptional regulator
K07649	two-component system, OmpR family, sensor histidine kinase TctE [EC:2.7.13.3]
K07395	putative proteasome-type protease
K07028	NA
K02391	flagellar basal-body rod protein FlgF
K01601	ribulose-bisphosphate carboxylase large chain [EC:4.1.1.39]
K03821	polyhydroxyalkanoate synthase [EC:2.3.1.-]

K07168	CBS domain-containing membrane protein
K06923	NA
K00411	ubiquinol-cytochrome c reductase iron-sulfur subunit [EC:1.10.2.2]
K01941	urea carboxylase [EC:6.3.4.6]
K17226	sulfur-oxidizing protein SoxY
K11897	type VI secretion system protein ImpF
K10125	two-component system, NtrC family, C4-dicarboxylate transport sensor histidine kinase DctB [EC:2.7.13.3]
K10126	two-component system, NtrC family, C4-dicarboxylate transport response regulator DctD
K04090	indolepyruvate ferredoxin oxidoreductase [EC:1.2.7.8]

1
2

Table S2. Enrichment analysis of genes influencing the gain of *rbsS*.

Annotation label	p-value[1]	test set[2]	background set[3]	Enrichment[4]
Nitric oxide reductase (Nor) complex	6.73E-05	4	5	12.83018868
Urea transport system (Urt)	8.62E-07	5	5	16.03773585
Purine degradation, xanthine=>urea	0.00042	4	7	9.164420485
Photorespiration	8.49E-05	5	9	8.909853249
Type IV secretion system	0.0031	4	11	5.831903945

1: from a hypergeometric test.
2: the number of genes with this annotation appearing in Table S1 (out of 129 genes).
3: the number of genes with this annotation appearing in the set of all genes in the PGCE network (out of 2472 genes).
4: The ratio of the observed proportion of genes with this label to the expected proportion.
5: The annotation of these genes to the same pathway is not present in KEGG, so this enrichment is derived from our manual annotation.

Table S3. Summary of nodes (genes) ranked by their order in a topological sort.

Rank	Number of genes	Total out-degree	Total in-degree
1	1593	7792	0
2	498	357	2512
3	118	73	2348
4	46	6	2992
5	5	0	376

Table S4. Characteristics of PGCE network models inferred from data subsets.

Dataset[1]	# PGCEs	ROC AUC[2]	Predictable / Total[3]
All (predicting Firmicutes)[c]	8,228	0.80	667 / 3281
Lacking Firmicutes	3,703	0.73	394 / 3281
Lacking A/B-proteobacteria	1,726	0.68	204 / 3505

1: The dataset used to train the PGCE model in question. Predictions are made concerning the test set (dataset lacking Firmicutes predicts Firmicutes).

2: Area under the curve of the receiver operating characteristic curve; a random prediction is 0.5, a perfect prediction is 1.0.

3: The number of genes that are predictable using each dataset to train PGCE models, compared to the total number of genes that are actually gained at least once (defined as Pr(gain) > 0.5) in the test set clade.

Supporting References

Baltrus DA. 2013. Exploring the costs of horizontal gene transfer. *Trends Ecol Evol* **28**: 489–95.

Cohen O, Ashkenazy H, Burstein D, Pupko T. 2012. Uncovering the co-evolutionary network among prokaryotic genes. *Bioinformatics* **28**: i389–i394.

Cohen O, Gophna U, Pupko T. 2011. The complexity hypothesis revisited: connectivity rather than function constitutes a barrier to horizontal gene transfer. *Mol Biol Evol* **28**: 1481–9.

Cohen O, Pupko T. 2010. Inference and characterization of horizontally transferred gene families using stochastic mapping. *Mol Biol Evol* **27**: 703–13.

Hassin R, Rubinstein S. 1994. Approximations for the maximum acyclic subgraph problem. *Inf Process Lett* **51**: 133–140.

Hausmann D, Korte B. 1978. K-greedy algorithms for independence systems. *Zeitschrift für Oper Res* **22**: 219–228.

Hsu HT. 1975. An Algorithm for Finding a Minimal Equivalent Graph of a Digraph. *J ACM* **22**: 11–16.

Knuth DE. 1973. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, Reading, Mass.

Maddison WP. 1990. A Method for Testing the Correlated Evolution of Two Binary Characters: Are Gains or Losses Concentrated on Certain Branches of a Phylogenetic Tree? *Evolution (N Y)* **44**: 539–557.

Minin VN, Suchard MA. 2008. Counting labeled transitions in continuous-time Markov models of evolution. *J Math Biol* **56**: 391–412.