Ancestral Reconstruction of Gene Blocks using an Event-Based Method

Huy N. Nguyen[1,4], Ashish Jain[2,3], Oliver Eulenstein[4], and Iddo Friedberg[1]

[1]Dpt. of Veterinary Microbiology and Preventive Medicine

[2]Dpt. of Genetics, Development, and Cell Biology

[3]Program in Bioinformatics and Computational Biology

[4]Dpt. of Computer Science

Iowa State University, Ames, IA USA

November 1, 2017

Abstract

Motivation: A fundamental attribute of life is complex systems: systems made of parts that together perform functions that a single component, or most subsets containing individual components, cannot. Examples of molecular complexity include protein structures such as the F_1F_o-ATPase, the ribosome, or the flagellar motor. Each one of these structures requires most or all of its components to function properly. Given the ubiquity of complex systems in the biosphere, understanding the evolution of complexity is central to biology. At the molecular level, operons are a classic example of a complex system. An operon's genes are co-transcribed under the control of a single promoter to a polycistronic mRNA molecule. The operon's gene products often form molecular complexes or metabolic pathways. With the large number of complete bacterial genomes available, we now have the opportunity to examine the evolution of operons and identify possible intermediate states.

Results: In this work, we used a maximum parsimony algorithm to reconstruct ancestral operon states, and show a simple vertical evolution model of how operons may evolve from the individual component genes. We offer the software as the **R**econstruction **of** **A**ncestral **G**enomes **U**sing **E**vents or ROAGUE.

Availability and implementation: The software is available on https://github.com/nguyenngochuy91/Ancestral-Blocks-Reconstruction

Contact: {huyn,idoerg}@iastate.edu

1 Introduction

The evolution of complex systems is an open problem in biology[1], and has recently been studied intensively in genomes[2, 3]. To better understand how complex systems evolve, we focus on the problem of the evolution of orthologous gene blocks and operons in bacteria. Orthologous gene blocks or *orthoblocks* are sequences of genes co-located on the chromosome, whose evolutionary conservation is apparent[4]. Operons can be viewed as a special case of gene blocks where the genes are co-transcribed to polycistronic mRNA and are often associated with a single function, such as a metabolic pathway or a protein complex. Several models have been proposed to explain gene block and operon evolution, and it may very well be that the models are not mutually exclusive, and different operons may evolve by different models, or indeed a single operon may be the result of the combination of several models[5, 6, 7, 8].

Previously, we proposed a method that explains the evolution of orthoblocks and operons as a combination of events that take place in vertical evolution from common ancestors. In the evolution of an orthoblock, the different gene blocks may gain or lose genes, have genes duplicated, or have them split off. By determining the frequency of the events for any orthoblock in a studied clade, we can determine a cost for each event, and thus create a cost function to determine an optimal vertical path for the evolution of orthoblocks. We have used the cost function to determine the conservation of some operons and orthoblocks in proteobacteria, and show that orthoblocks that perform cellular information processing (such as mRNA translation) are more conserved than those that are associated with adaptation to specific environments [4].

In this study, we use the orthoblock evolution cost function model to reconstruct ancestral gene blocks. Reconstructing plausible ancestral states of extant complex entities can help us understand how they evolve, and which forces might affect their evolution. The rest of this paper is structured as follows: first, we present two algorithms that reconstruct ancestral states of orthoblocks. We then use these algorithms to reconstruct the ancestral states of orthoblocks in a clade of Gram-negative bacteria and a clade of Gram-positive bacteria. This reconstruction involves orthoblocks comprising genes orthologous to those found in operons in *Escherichia coli* and in *Bacillus subtilis*, respectively. Finally, we present our findings and discuss our results. Our reconstructions of ancestral states show that: (1) some operons can rapidly evolve independently in several branches in their respective clades, suggesting that positive selection plays a major role in the evolution of gene blocks in bacteria; (2) other operons are highly conserved, their evolution predating the last common ancestor of the clades we chose, and (3) some operon conservation is sporadic and cannot be explained solely by vertical transmission suggesting horizontal gene transfer.

2 Methods

2.1 Definitions

2.1.1 Gene block-based evolutionary events, and event-based distances

The terms reference taxa, neighboring genes, gene blocks, events, and orthoblocks are elaborated upon in [4]. Briefly, a **reference taxon** is a taxon where operons have been identified by experimental means. Here we use *E. coli* K-12 MG1655 and *B. subtilis* as reference taxa. The reference taxon serves as a standard of truth to determine if the genes on a suspected orthoblock do indeed reside, at least in one species, in an operon or similar co-regulated gene block. We chose these species because their genomes are expertly and comprehensively annotated, and experimental evidence exists for many of their operons [9]. **Neighboring genes**: two genes are considered neighboring if they are 500 nucleotides or fewer apart and on the same strand. A **gene block** comprises no fewer than two open reading frames of ORFs that are neighboring. **Orthoblocks**, gene blocks that are orthologous, are defined as follows: two organisms have orthoblocks when each organism must have at least two neighboring genes that are homologous to genes in a gene block in the reference taxon's genome. An **event** is a change in the gene block between any two species with homologous gene blocks.

We identify three types of pairwise events between orthoblocks in different taxa: splits, deletions, and duplications. The *event-based distance* between any two orthoblocks is the sum of the minimized count of splits, duplications, and deletions.

2.1.2 Choosing species

The species tree for each clade was built using *rpoB* as the species marker. For the study of Gram negatives with *E. coli* as a reference species, we use the group of taxa from[4]. For the study of Gram positives with *B. subtilis* as the reference species, we use the Phylogenetic Diversity Analysis program (PDA)[10, 11] to select 33 equidistant species.

2.1.3 Orthoblocks in Phylogenetic Trees

For each orthoblock studied, we use a phylogenetic species tree T comprising a set of extant species related to either one of our reference taxa. The topology of T is determined using multiple sequence alignment of gene *rpoB* followed by the Neighbor Joining algorithm as described in [4]. Each leaf node v in T contains the orthologs to the genes in an operon in the reference species (*E. coli* or *B. subtilis*). For any two genes a and b, if the chromosomal distance is less than 500 bp, the genes will be written as ab. If the distance is greater than 500 bp, they are written with the separator character '$|$' thus: $a|b$. For a species tree T, we define the following:

1. $V(T)$: the set of nodes of T.

2. $E(T)$: the set of edges of T.

3. $L(T)$: set of leaf nodes of T.

4. $I(T)$: set of inner nodes of T.

   If a node v is an inner node, it can be one of three types (illustrated in Figure 1):

   (a) v_{dl}: an inner node whose both children are leaf nodes.

   (b) v_{hl}: an inner node that only one immediate child is a leaf node.

   (c) v_{nl}: an inner node that none of its children are leaf nodes.

5. For node $v \in V(T)$, let O be the gene block assigned to v, we define:

   (a) $v.gene[g]$: the set that represents the choice of including of gene g in O. There are only 3 possible cases.

      i. $v.gene[g] = \{1\}$: this means that gene g has to be in O.

      ii. $v.gene[g] = \{0\}$: this means that gene g can not be in O.

      iii. $v.gene[g] = \{0, 1\}$: this means that gene g can either be in O or not in v.

   (b) $I_g(v)$: the identiy function of gene g in O. It can only takes value of 0 for not appearing in O or 1 ortherwise.

   (c) $v.dup[g]$: the set that represents the duplication status of gene g in O. There are only 3 possible cases.

      i. $v.dup[g] = \{1\}$: this means that gene g has to be duplicated in O.

      ii. $v.dup[g] = \{0\}$: this means that gene g can not be duplicated in O.

      iii. $v.dup[g] = \{0, 1\}$: this means that gene g can either be duplicated or not in O.

   (d) $Gene(O)$: the set of gene of O.

   (e) $Dup(O)$: the set of gene that is duplicated in O.

   (f) $HasLeaf(v)$: the set of leaf nodes that can be reached from node v in postorder traversal.

   (g) $FREQ_g(v)$: frequency of gene g in $HasLeaf(v)$.

(h) $DUP_g(v)$: frequency of duplications of gene g in $HasLeaf(v)$.

Figure 1 shows an example of orthoblocks and node types on a phylogenetic tree.

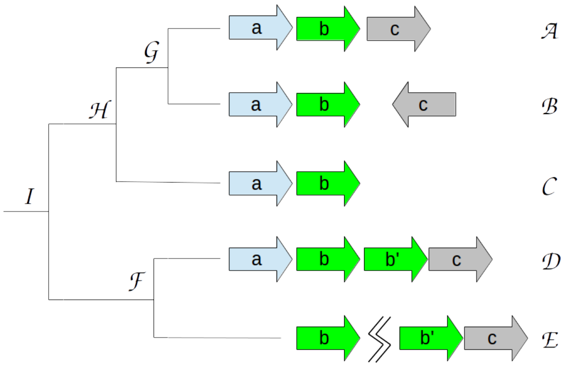

Figure 1: Example of orthoblocks, the event-driven model, and node classifications. $\mathcal{A}$ is the reference taxon, having an experimentally-verified operon composed of genes a, b and c. Nodes $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}$ are leaf nodes: extant species with different orthoblocks whose genes are orthologous to the operon in $\mathcal{A}$. As an example of events, a duplication event (of a gene b homolog) occurs in the pairwise comparison of A and D. Nodes $\mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}$ are internal. $\mathcal{F}$ and $\mathcal{G}$ are v_{dl} type nodes, $\mathcal{H}$ is a v_{hl} type node, and $\mathcal{I}$ is a v_{nl} type node. See Methods for details. (Based on [4])

2.2 Orthoblock distance functions

The distance between any two homologous gene blocks O, O' found in target organisms is defined as in [4]. We provide the definition and the formula to calculate each distance function as follows:

1. *Split distance* (d_s) is the absolute difference in the number of relevant gene blocks between the two taxa. Relevant gene blocks between two taxa can be computed by only including the genes that appear in both taxa. We define $Rel(O, O')$ as relevant gene blocks of O to O' and formalize the split distance as:

$$d_s(O, O') := ||Rel(O, O')| - |Rel(O', O)||$$

Example: for the reference gene block with genes (abcdefg), genome A has blocks $O := ((ab), (def))$ and genome B has $O' := ((abc), (de), (fg))$. We then compute the relevant gene blocks $Rel(O, O') = ((ab), (def))$ and $Rel(O', O) = ((ab), (de), (f))$ (removing genes c, g). Therefore, $d_s(O, O') = |2-3| = 1$.

2. *Duplication distance (d_u)* is the pairwise count of duplications between two gene blocks. We define $Dif(O, O')$ as the set of duplicated genes of gene block O, so that these genes also appear in O' but are not duplicated in O'. We formalize the duplication distance as:

$$d_u(O, O') := |Dif(O, O')| + |Dif(O', O)|$$

Example: For a reference gene block $(abcde)$, genome A has gene block $O = ((abd))$ and genome B has gene block $O' = ((abbcc))$, respectively. The ortholog of gene O_b is duplicated in genome B, creating a duplication distance $d_u(O, O')$ of 1. However, since gene c does not exist in O, it has no bearing on the duplication distance between the homologous gene blocks O and O'. We then compute $Dif(O, O') = \emptyset$ and $Dif(O', O) = \{b\}$. Therefore, $d_u(O, O') = 0 + 1 = 1$

3. *Deletion distance (d_d)* is the difference in the number of orthologs that are in the homologous gene blocks of the genome of one organism, or the other, but not in both. In short, it is the symmetric difference between the set of orthologous genes of the two gene blocks O, O'. We formalize the deletion distance as:

$$d_d(O, O') := |Gene(O) \bigtriangleup Gene(O')|$$

In addition, the deletion distance can also be defined using the identity function:

$$d_d(O, O') := |\sum_g (I_g(O) - I_g(O'))|$$

Example: For a reference gene block $(abcde)$, genome A has gene block $O = ((abd))$ and genome B has gene block $O' = ((abce))$, respectively. Since there are only genes a, b that appear in both genomes, $d_d(O, O') = |\{a, b, d\} \bigtriangleup \{a, b, c, e\}| = |\{d\}| + |\{c, e\}| = 3$

The duplication distance and split distance depend on the deletion distance. Intuitively, the duplication of a gene g in gene block O requires such gene appearing in O. Split distance depends on the relevant gene blocks from two taxa. Hence, it depends on the genes that appear in both taxa. Therefore, the split distance and the duplication distance depend on the deletion distance. Using the three distance functions above, we define theO total distance between any two homologous gene blocks O, O' as:

$$d(O, O') := d_d(O, O') + d_u(O, O') + d_s(O, O')$$

2.3 Problem definition

Let T be a tree, and G be the set of genes in a reference operon. We define Ω as the set of all possible orthoblocks over gene set G. Let $\lambda : L \mapsto \Omega$ be the labeling of L (assign orthoblocks from Ω to the leaf nodes of T, this can include empty orthoblocks). We define the function $\hat{\lambda} : V \mapsto \Omega$ to be an extension of λ on T if it coincides with λ on the leaves of T (assign an orthoblock to each node of T). If $\hat{\lambda}(v) = O$, we say that vertex v is labelled with orthoblock O. Furthermore, given orthoblock O, we define $GeneBlock(O)$ as the set of gene blocks in O. Given a labelling $\hat{\lambda}$ and an edge $(u, 1v) \in E$, we define the distance between the two labellings of the endpoints u, v as $d(u, v) := d(\hat{\lambda}(u), \hat{\lambda}(v))$ and the total distance function as $d(\hat{\lambda}) := \sum_{(u,v)\in E} d(u, v)$.

The Maximum Parsimony problem is now defined as follows: given a tree T, an operon gene set G, the orthoblock set Ω and a leaf labeling λ, find a labeling $\hat{\lambda}$ that minimizes $d(\hat{\lambda})$

3 Approach

Here we explore two related Maximum Parsimony heuristic approaches, local and global, to reconstruct ancestral gene blocks.

3.1 Local Maximum Parsimony

Briefly, the local approach focuses on finding the optimal parent ancestral gene block given its child gene blocks. For each internal node u, let u_1 and u_2 be its 2 direct children. We present a greedy local optimization algorithm.

```
Input: $T$,$G$,$\Omega$, $\lambda$
Result: $\hat{\lambda}$
for internal node $u$ when traversing $T$ in post-order do
    Let $u_1$, $u_2$ to be $u$'s children
    Let $O_1 := \lambda(u_1)$, $O_2 := \lambda(u_2)$
    $initial := GeneBlock(O_1) \cup GeneBlock(O_2)$
    $initial_{gene} := \{g \big| FREQ_g(u) \geq .5\}$
    Remove genes in $initial$ that is not included in $initial_{gene}$
    Remove gene blocks in $initial$ that is a subset of another gene block in $initial$
    Let $U_{1_G} := set()$
    for gene block $b \in GeneBlock(O_1)$ do
        for gene $g$ in $b$ do
            if $g \notin initial_{gene}$ then
                Remove gene $g$ from $b$
            end
        end
        $U_{1_G} = U_{1_G} \cup b$;
    end
    if $|initial| < |U_{1_G}|$ or $|initial| > |U_{1_G}|$ then
        $initial := U_{1_G}$
    end
    for gene block $b \in initial$ do
        if $b$ has a duplication of gene $g$ and $DUP_s(u) \leq .5$ then
            Remove the duplicated gene of $g$ from $b$
        end
    end
    $\hat{\lambda}(u) := initial$
end
Return $\hat{\lambda}$
```

Algorithm 1: Local cost function minimization for reconstructing ancestral nodes

For proof of correctness and runtime, please refer to section 6.1 .

3.2 Global Maximum Parsimony

In section 2.2, we determined that the split distance and duplication distance depend on the deletion distance. While finding the global minimum for each separate distance is simple, this dependency makes finding the global minimum of the aggregate of the three distances challenging. In the following example, we demonstrate the minimization of the deletion distance, and then of the split distance. After that, we provide an optimal solution that minimizes the aggregate sum of the two distances.

Given an inner node v and its two child nodes v_1 and v_2, let O be the gene block to be assigned to v. Consider the orthoblocks O_1 and O_2 of v_1 and v_2 respectively as:

$$O_1 : ab|cd|ef|g|k$$

$$O_2 : bc|de|fb|f|fo$$

We define the set of genes that appear in both O_1 and O_2 as $S = \{b, c, d, e, f\}$, and the union gene set of O_1 and O_2 as $G = \{a, b, c, d, e, f, g, k, o\}$. Any gene $i \in S$ will contribute a deletion distance of 2 to $d_d(O, O_1) + d_d(O, O_2)$ if O does not contain gene i. Any gene $i \in G$ but $i \notin S$ will contribute a deletion distance of 1 to $d_d(O, O_1) + d_d(O, O_2)$ if O either has it or not. Hence, only including all genes from S in O, $d_d(O, O_1) + d_d(O, O_2) = 4$, which is the minimum deletion distance. On the other hand, if we just want to minimize the split distance, the most naive way is not including any genes in O. Then, $Rel(O, O_1) = Rel(O, O_2) = \emptyset$, therefore $d_s(O, O_1) + d_s(O, O_2) = 0$. However, if we choose to do it this way, our deletion distance becomes large ($d_d(O, O_1) + d_d(O, O_2) = 10$). Apparently, decreasing split distance might increase deletion distance and vice versa.

If we focus on minimizing the deletion distance, then $Gene(O) = S$, which means that O has to include all genes in S. Then, the relevant gene blocks between O and its children O_1, O_2 become:

$$Rel(O_1,O) : b|cd|ef$$

$$Rel(O_2,O) : bc|de|fb|f|f$$

Apparently, the split distance of O_1, O_2 is $d_s(O_1, O_2) = |5 - 3| = 2$. If we remove gene f from $Gene(O)$, the relevant gene blocks of the two children of u become:

$$Rel(O_1,O) : b|cd|e$$

$$Rel(O_2,O) : bc|de|b$$

Hence, by setting our gene block O as either $Rel(O, O_1)$ or $Rel(O, O_2)$, the deletion distance increased by 2 since we excluded a gene that is in S; also, the split distance also decreased by 2. Therefore, the new deletion distance is $d_d(O, O_1) + d_d(O, O_2) = 6$, and the new split distance is $d_s(O, O_1) + d_s(O, O_2) = 0$.

Consider another possibility, if we include gene g in $Gene(O)$ (not increasing the deletion distance), the relevant gene blocks to u become:

$$O_1, O : b|cd|ef|g$$

$$O_2, O : bc|de|fb|f|f$$

By setting $O := b|cd|ef|g$, the new split distance is $d_s(O, O_1) + d_s(O, O_2) = 1$ and the deletion distance is $d_d(O, O_1) + d_d(O, O_2) = 4$. Therefore, we achieve a lower aggregate sum of deletion and split distances (5 compared to 6). We can keep on adding, or removing genes that only appear in one taxon. This process requires iterations through all the subsets of the symmetrical difference $Gene(O_1) \triangle Gene(O_2)$ which will take exponential time. We therefore provide a heuristic approach that guarantees minimum deletion and duplication distances, but not split distances.

```
Input: $T, G, \Omega, \lambda$
Result: $\hat{\lambda}$
for gene $g \in G$ do
    for $l \in Leaf(T)$ do
        if gene $g \in Gene(\lambda(l))$ then
            $l.gene[g] = \{1\}$
        else
            $l.gene[g] = \{0\}$
        end
    end
    if gene $g \in Dup(l)$ then
        $l.dup[g] = \{1\}$
    else
        $l.dup[g] = \{0\}$
    end
end
for internal node $u$ when traversing $T$ in post-order do
    Let $u_1, u_2$ be children of $u$
    for gene $g \in G$ do
        if $u_1.gene[g] == u_2.gene[g]$ then
            $u.gene[g] = u_1.gene[g]$
        else
            $u.gene[g] = \{0, 1\}$
        end
        if $u_1.dup[g] == u_2.dup[g]$ then
            $u.dup[g] = u_1.dup[g]$
        else
            $u.dup[g] = \{0, 1\}$
        end
    end
end
for inner node $u \in V(T)$ do
    for gene $g \in G$ do
        if $1 \in u.gene[g]$ then
            $Gene(u).add(g)$
        end
        if $1 \in u.dup[g]$ then
            $Dup(u).add(g)$
        end
    end
end
for internal node $u$ when traversing $T$ in post-order do
    Let $u_1$ be a $u$'s child, $O_1 := \lambda(u_1)$
    Let $U_{1_G} := set()$
    for gene block $b \in GeneBlock(O_1)$ do
        for gene $g$ in $b$ do
            if $g \notin initial_{gene}$ then
                Remove gene $g$ from $b$
            end
        end
        $U_{1_G} = U_{1_G} \cup b$;
    end
    $\hat{\lambda}(u) := U_{1_G}$ for gene $g \in Gene(u)$ do
        if $g \notin Dup(u)$ then
            Remove the duplicated of $g$ from $\hat{\lambda}(u)$
        end
    end
end
Return $\hat{\lambda}$;
```

Algorithm 2: Global approach

For proof of correctness and runtime, please refer to Supplementary Materials.

4 Results and Discussion

We used *E. coli* and *B. subtilis* genomes as gold standards for deriving operons from Gram negative and Gram positive bacteria, respectively. The reason we picked these two species is that they both have well-annotated genomes, including experimentally verified operons. We applied our method to groups of Gram-negative and Gram-negative bacteria, using the operons experimentally identified in *E. coli* K-12 and *B. subtilis* str. 168 for the two groups, respectively.

4.1 Operons from *Escherichia coli*

We chose *E. coli* as a representative of proteobacteria, a major group of Gram-negative bacteria. Here, we examine across 33 taxa of proteobacteria from [4]. Our selection resulted in a set of proteobacteria species comprising three ϵ-proteobacteria, six α-proteobacteria, seven β-proteobacteria and 17 γ-proteobacteria. The latter includes the reference species *E. coli*. Our selection included two γ-proteobacteria insect endosymbionts: *Buchnera aphidicola* and *Candidatus Blochmania*. These two species have unusually small genomes due to their endosymbiotic nature, and display massive gene loss. We reconstructed ancestors for the following operons from *E. coli*: atpIBEFHAGDC, paaABCDEFGHIJK, and the regulon bamA-skp-lpxD-fabZ-lpxAB-rnhB-dnaE.

atpIBEFHAGDC. The *atpIBEFHAGDC* operon codes for F_1F_o-ATPase, which catalyzes the synthesis of ATP from ADP and inorganic phosphate [12]. ATP synthase is composed of two fractions: F_1 and F_o [13]. The F_1 fraction contains the catalytic sites and its proteins are coded by five genes (*atpA, atpC, atpD, atpG, atphH*) [13]. The F_o complex constitutes the proton channel and its proteins are coded by three genes *atpF, atpE, atpB.*

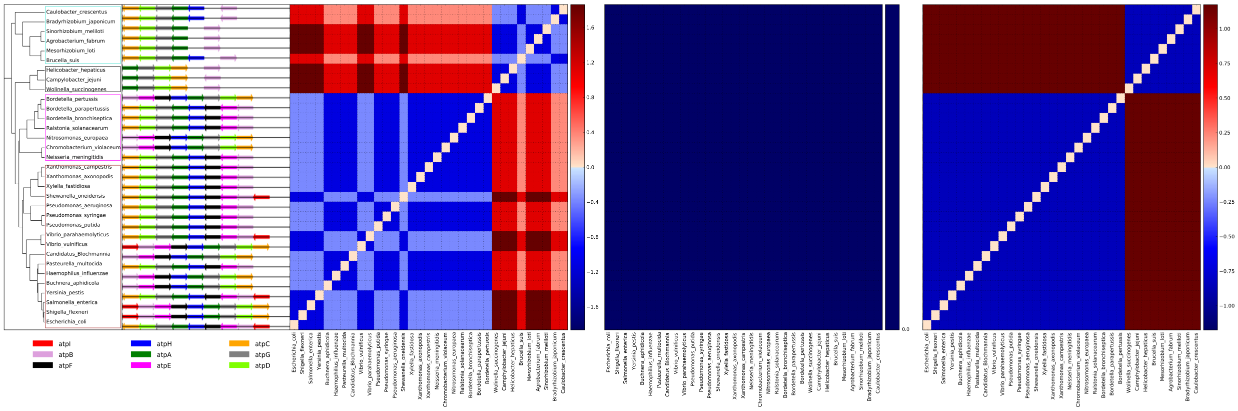

Figure 2: A phylomatrix of gene block atpIBEFHAGDC. Each matrix square depicts the degree of relative conservation of the event between any two species. Blue is more conserved, red is less conserved. Left to right: conservation of deletions, duplications, splits. z-score value calculated as in[4]. As can be seen, there are few deletions and split events, and no duplications events in the pairwise comparison of this gene block, showing a high conservation. Reproduced from [4] under Creative Common CC-BY-NC 4.0. license. A larger version can be found here http://iddo-friedberg.net/operon-evolution/

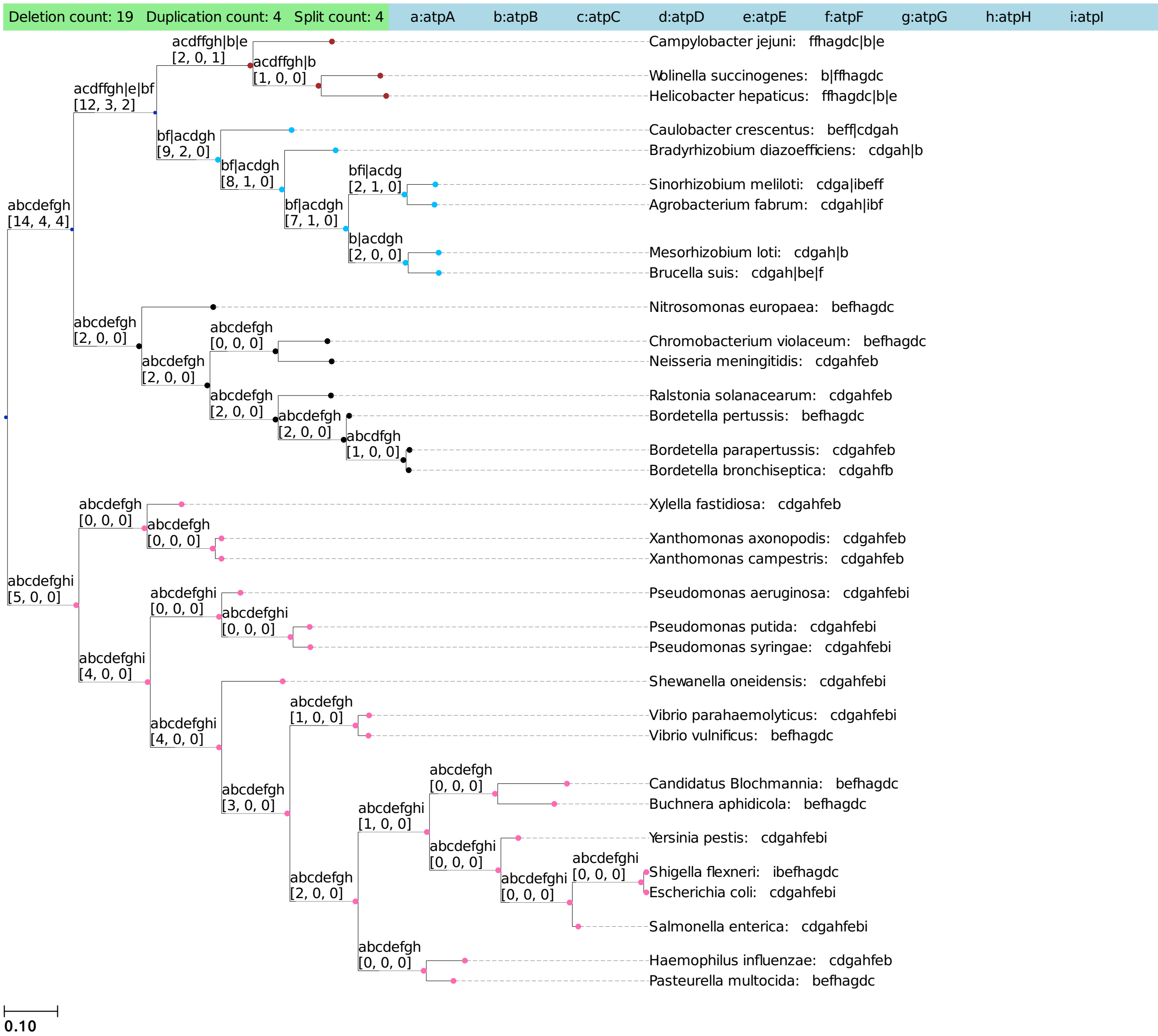
Deletion count: 19
Duplication count: 4
Split count: 4
a:atpA
b:atpB
c:atpC
d:atpD
e:atpE
f:atpF
g:atpG
h:atpH
i:atpI
acdffgh|b|e [2, 0, 1]
acdffgh|b [1, 0, 0]
acdffgh|e|bf [12, 3, 2]
bf|acdgh [9, 2, 0]
bf|acdgh [8, 1, 0]
bfi|acdg [2, 1, 0]
bf|acdgh [7, 1, 0]
b|acdgh [2, 0, 0]
abcdefgh [14, 4, 4]
abcdefgh [2, 0, 0]
abcdefgh [0, 0, 0]
abcdefgh [2, 0, 0]
abcdefgh [2, 0, 0]
abcdefgh [2, 0, 0]
abcdfgh [1, 0, 0]
abcdefgh [0, 0, 0]
abcdefgh [0, 0, 0]
abcdefghi [5, 0, 0]
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 0]
abcdefghi [4, 0, 0]
abcdefghi [4, 0, 0]
abcdefgh [1, 0, 0]
abcdefgh [3, 0, 0]
abcdefgh [0, 0, 0]
abcdefghi [1, 0, 0]
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 0]
abcdefgh [2, 0, 0]
abcdefgh [0, 0, 0]
Campylobacter jejuni: ffhagdc|b|e
Wolinella succinogenes: b|ffhagdc
Helicobacter hepaticus: ffhagdc|b|e
Caulobacter crescentus: beff|cdgah
Bradyrhizobium diazoefficiens: cdgah|b
Sinorhizobium meliloti: cdga|ibeff
Agrobacterium fabrum: cdgah|ibf
Mesorhizobium loti: cdgah|b
Brucella suis: cdgah|be|f
Nitrosomonas europaea: befhagdc
Chromobacterium violaceum: befhagdc
Neisseria meningitidis: cdgahfeb
Ralstonia solanacearum: cdgahfeb
Bordetella pertussis: befhagdc
Bordetella parapertussis: cdgahfeb
Bordetella bronchiseptica: cdgahfb
Xylella fastidiosa: cdgahfeb
Xanthomonas axonopodis: cdgahfeb
Xanthomonas campestris: cdgahfeb
Pseudomonas aeruginosa: cdgahfebi
Pseudomonas putida: cdgahfebi
Pseudomonas syringae: cdgahfebi
Shewanella oneidensis: cdgahfebi
Vibrio parahaemolyticus: cdgahfebi
Vibrio vulnificus: befhagdc
Candidatus Blochmannia: befhagdc
Buchnera aphidicola: befhagdc
Yersinia pestis: cdgahfebi
Shigella flexneri: ibefhagdc
Escherichia coli: cdgahfebi
Salmonella enterica: cdgahfebi
Haemophilus influenzae: cdgahfeb
Pasteurella multocida: befhagdc
0.10

Figure 3: Ancestral reconstruction of operon *atpIBEFHAGDC* using the local optimization approach. Brown: ϵ-protebacteria; blue: α-proteobacteria; black: β-proteobacteria; pink: γ-proteobacteria.

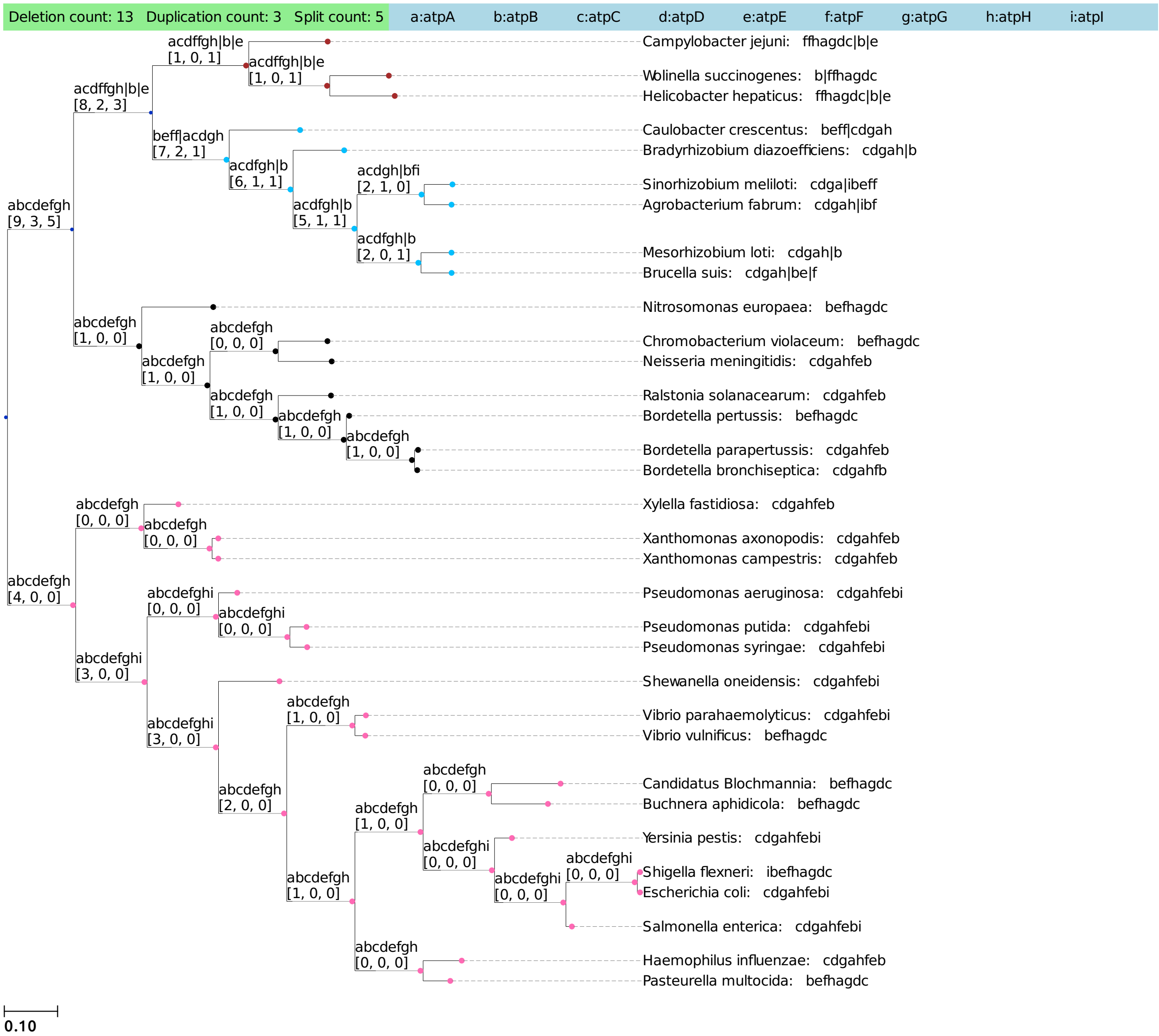
Deletion count: 13
Duplication count: 3
Split count: 5
a:atpA
b:atpB
c:atpC
d:atpD
e:atpE
f:atpF
g:atpG
h:atpH
i:atpI
abcdefgh [9, 3, 5]
acdffgh|b|e [8, 2, 3]
acdffgh|b|e [1, 0, 1]
acdffgh|b|e [1, 0, 1]
Campylobacter jejuni: ffhagdc|b|e
Wolinella succinogenes: b|ffhagdc
Helicobacter hepaticus: ffhagdc|b|e
beff|acdgh [7, 2, 1]
Caulobacter crescentus: beff|cdgah
acdfgh|b [6, 1, 1]
Bradyrhizobium diazoefficiens: cdgah|b
acdfgh|b [5, 1, 1]
acdgh|bfi [2, 1, 0]
Sinorhizobium meliloti: cdga|ibeff
Agrobacterium fabrum: cdgah|ibf
acdfgh|b [2, 0, 1]
Mesorhizobium loti: cdgah|b
Brucella suis: cdgah|be|f
abcdefgh [1, 0, 0]
Nitrosomonas europaea: befhagdc
abcdefgh [1, 0, 0]
abcdefgh [0, 0, 0]
Chromobacterium violaceum: befhagdc
Neisseria meningitidis: cdgahfeb
abcdefgh [1, 0, 0]
Ralstonia solanacearum: cdgahfeb
abcdefgh [1, 0, 0]
Bordetella pertussis: befhagdc
abcdefgh [1, 0, 0]
Bordetella parapertussis: cdgahfeb
Bordetella bronchiseptica: cdgahfb
abcdefgh [4, 0, 0]
abcdefgh [0, 0, 0]
Xylella fastidiosa: cdgahfeb
abcdefgh [0, 0, 0]
Xanthomonas axonopodis: cdgahfeb
Xanthomonas campestris: cdgahfeb
abcdefghi [3, 0, 0]
abcdefghi [0, 0, 0]
Pseudomonas aeruginosa: cdgahfebi
abcdefghi [0, 0, 0]
Pseudomonas putida: cdgahfebi
Pseudomonas syringae: cdgahfebi
abcdefghi [3, 0, 0]
Shewanella oneidensis: cdgahfebi
abcdefgh [2, 0, 0]
abcdefgh [1, 0, 0]
Vibrio parahaemolyticus: cdgahfebi
Vibrio vulnificus: befhagdc
abcdefgh [1, 0, 0]
abcdefgh [1, 0, 0]
abcdefgh [0, 0, 0]
Candidatus Blochmannia: befhagdc
Buchnera aphidicola: befhagdc
abcdefghi [0, 0, 0]
Yersinia pestis: cdgahfebi
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 0]
Shigella flexneri: ibefhagdc
Escherichia coli: cdgahfebi
Salmonella enterica: cdgahfebi
abcdefgh [0, 0, 0]
Haemophilus influenzae: cdgahfeb
Pasteurella multocida: befhagdc
0.10

Figure 4: Ancestral reconstruction of operon *atpIBEFHAGDC* using the global optimization approach. Brown is ϵ-protebacteria, blue is α-proteobacteria, black is β-proteobacteria, pink is γ-proteobacteria.

Figures 3 and 4 show ancestral reconstruction using the local and global maximum parsimony algorithms, respectively. Both local and global reconstructions show a consistency of having orthoblocks *atpACDGH* and *atpBF* in the most common ancestors for different Gram negative bacteria. This finding agrees with the long-standing hypothesis that F_o and the F_1 fractions have evolved separately, with the respective fractions having homologs in the hexameric DNA helicases and with flagellar motor complexes. Although we find the gene *atpI* in several species, the reconstruction predicts that *atpI* is not in the same cluster with other genes. Gene *atpI* is not an essential component of the F_1F_o ATPase[14]. Another interesting finding is the duplication of *atpF* in ϵ-proteobacteria which appears to predate their common ancestor. Note that all genes exist as a gene block even in the endosymbionts *Blochmannia* and *B. aphidicola*.

The ϵ, α, β, and γ -proteobacteria species all have a conserved intact F_1 complex (coded by the *atpACDGH* cluster), which predates their common ancestor. The genes included in the F_o complex in *epsilon*-proteobacteria (gene products *atpB, atpE,atpF*) not in the same cluster as the genes making up F_1. Furthermore, it is unclear whether the gene split that is only found in ϵ-proteobacteria is a split that predates the least common ancestor with the other proteobacteria clades, or whether it is a split introduced in the ϵ-proteobacteria. From the reconstructions provided, the scenario appears to be the latter. Conversely, this observation may also be a result of the small number of species studied here. The species in the ϵ and α-proteobacteria display a known duplication of gene *atpF*. *atpF'* appears as a sister group to *atpF*[15].

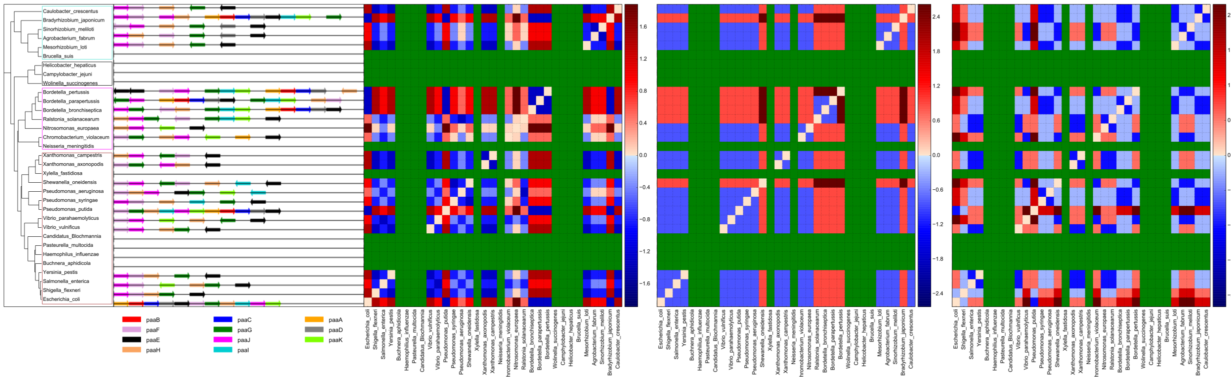

Figure 5: Gene block paaABCDEFGHIJK phylomatrices, each show the degree of relative conservation of the event between any two species. Left to right: Deletions, duplications, splits. Blue to red scale is high-to-low conservation z-score [4]. Larger file can be found here http://iddo-friedberg.net/operon-evolution/

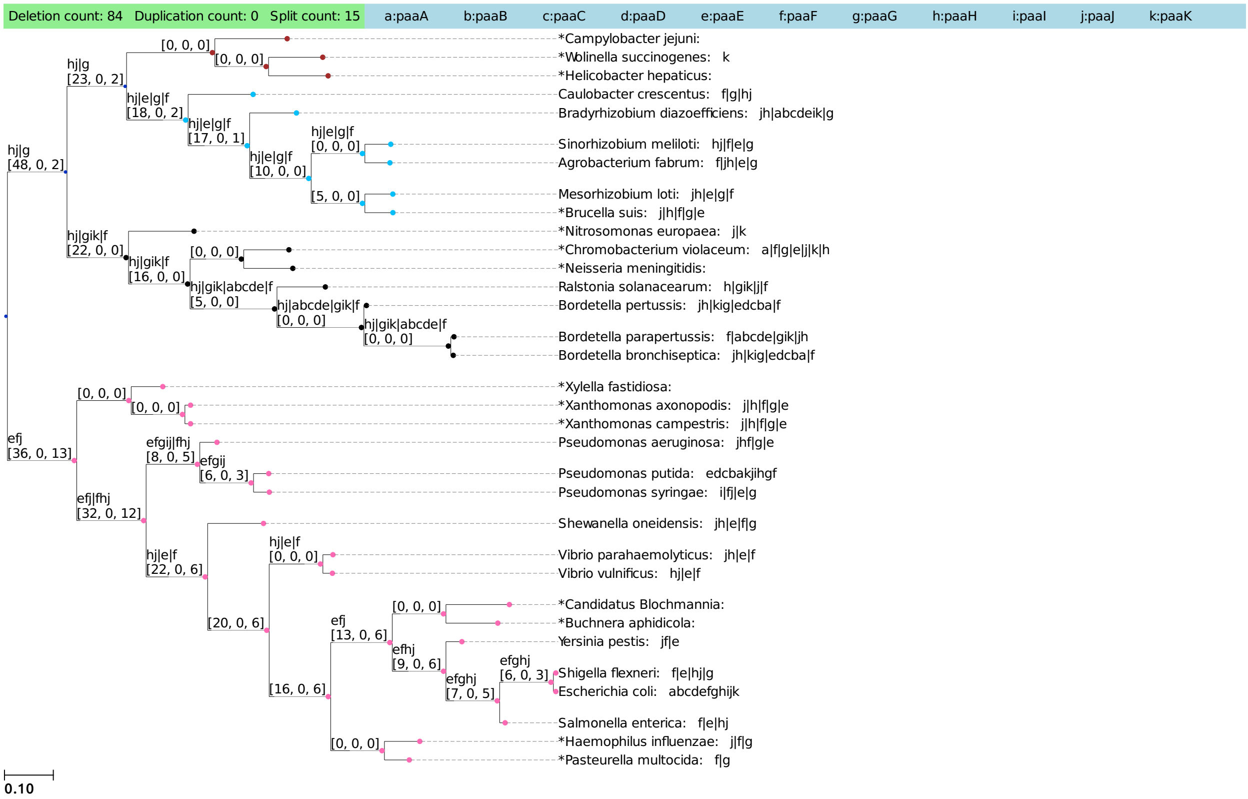
Deletion count: 84
Duplication count: 0
Split count: 15
a:paaA
b:paaB
c:paaC
d:paaD
e:paaE
f:paaF
g:paaG
h:paaH
i:paaI
j:paaJ
k:paaK
[0, 0, 0]
*Campylobacter jejuni:
[0, 0, 0]
*Wolinella succinogenes: k
*Helicobacter hepaticus:
hj|g
[23, 0, 2]
Caulobacter crescentus: f|g|hj
hj|e|g|f
[18, 0, 2]
Bradyrhizobium diazoefficiens: jh|abcdeik|g
hj|e|g|f
[17, 0, 1]
hj|e|g|f
[0, 0, 0]
Sinorhizobium meliloti: hj|f|e|g
hj|g
[48, 0, 2]
hj|e|g|f
[10, 0, 0]
Agrobacterium fabrum: f|jh|e|g
[5, 0, 0]
Mesorhizobium loti: jh|e|g|f
*Brucella suis: j|h|f|g|e
hj|gik|f
[22, 0, 0]
*Nitrosomonas europaea: j|k
[0, 0, 0]
*Chromobacterium violaceum: a|f|g|e|j|k|h
hj|gik|f
[16, 0, 0]
*Neisseria meningitidis:
hj|gik|abcde|f
[5, 0, 0]
Ralstonia solanacearum: h|gik|j|f
hj|abcde|gik|f
[0, 0, 0]
Bordetella pertussis: jh|kig|edcba|f
hj|gik|abcde|f
[0, 0, 0]
Bordetella parapertussis: f|abcde|gik|jh
Bordetella bronchiseptica: jh|kig|edcba|f
[0, 0, 0]
*Xylella fastidiosa:
[0, 0, 0]
*Xanthomonas axonopodis: j|h|f|g|e
*Xanthomonas campestris: j|h|f|g|e
efj
[36, 0, 13]
efgij|fhj
[8, 0, 5]
Pseudomonas aeruginosa: jhf|g|e
efgij
[6, 0, 3]
Pseudomonas putida: edcbakjihgf
Pseudomonas syringae: i|fj|e|g
efj|fhj
[32, 0, 12]
Shewanella oneidensis: jh|e|f|g
hj|e|f
[0, 0, 0]
hj|e|f
[22, 0, 6]
Vibrio parahaemolyticus: jh|e|f
Vibrio vulnificus: hj|e|f
[0, 0, 0]
*Candidatus Blochmannia:
efj
[13, 0, 6]
*Buchnera aphidicola:
[20, 0, 6]
Yersinia pestis: jf|e
efhj
[9, 0, 6]
efghj
[6, 0, 3]
Shigella flexneri: f|e|hj|g
efghj
[7, 0, 5]
Escherichia coli: abcdefghijk
[16, 0, 6]
Salmonella enterica: f|e|hj
[0, 0, 0]
*Haemophilus influenzae: j|f|g
*Pasteurella multocida: f|g
0.10

Figure 6: Ancestral gene block reconstruction of $paaABCDEFGHIJK$ using the local reconstruction approach. Clade colors from top: brown: ϵ-proteobacteria, blue: α-proteobacteria, black: β-proteobacteria, pink: γ-bacteria. Asterisks in front of species names indicate that a minimal orthoblock (two or more proximal orthologs to the reference operon) was not found.

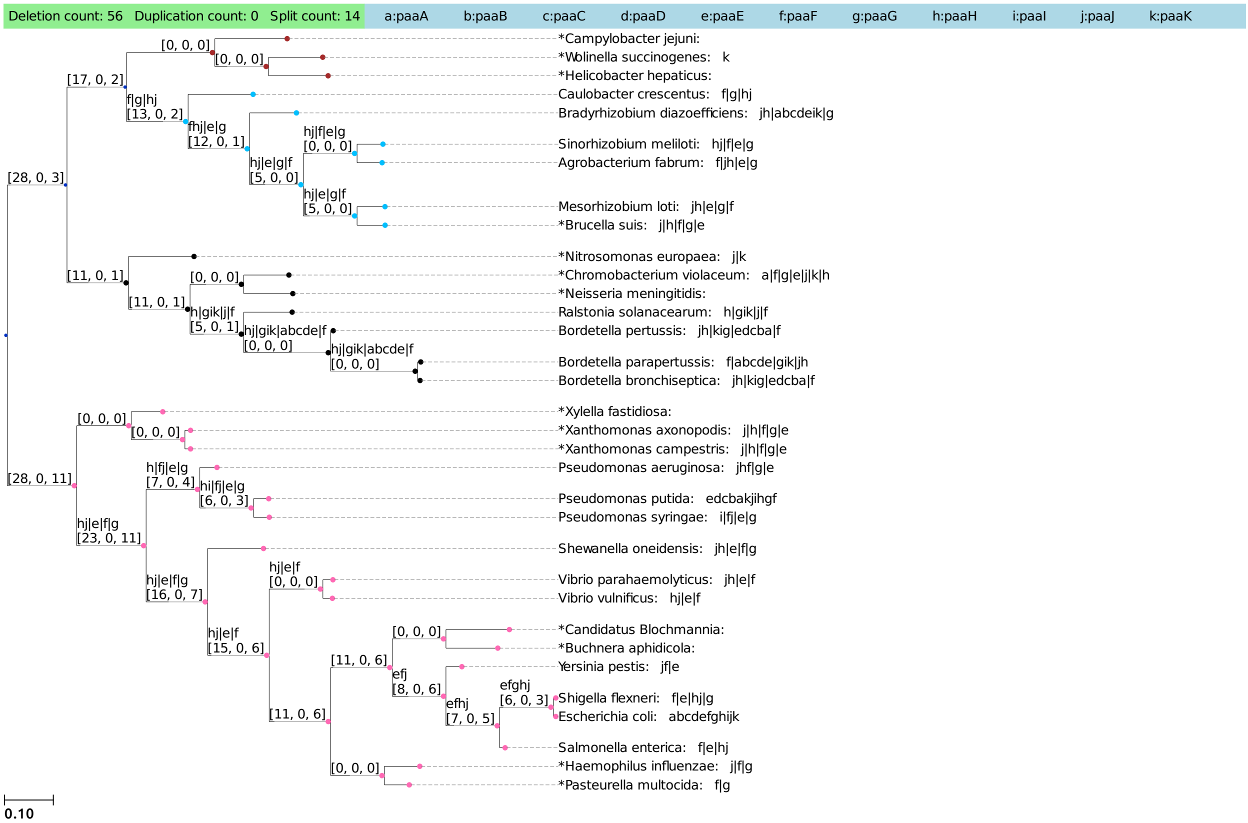
Deletion count: 56 Duplication count: 0 Split count: 14
a:paaA b:paaB c:paaC d:paaD e:paaE f:paaF g:paaG h:paaH i:paaI j:paaJ k:paaK
[0, 0, 0]
*Campylobacter jejuni:
[0, 0, 0]
*Wolinella succinogenes: k
*Helicobacter hepaticus:
[17, 0, 2]
f|g|hj
[13, 0, 2]
Caulobacter crescentus: f|g|hj
fhj|e|g
[12, 0, 1]
Bradyrhizobium diazoefficiens: jh|abcdeik|g
hj|f|e|g
[0, 0, 0]
Sinorhizobium meliloti: hj|f|e|g
Agrobacterium fabrum: f|jh|e|g
[28, 0, 3]
hj|e|g|f
[5, 0, 0]
hj|e|g|f
[5, 0, 0]
Mesorhizobium loti: jh|e|g|f
*Brucella suis: j|h|f|g|e
*Nitrosomonas europaea: j|k
[11, 0, 1]
[0, 0, 0]
*Chromobacterium violaceum: a|f|g|e|j|k|h
*Neisseria meningitidis:
[11, 0, 1]
h|gik|j|f
[5, 0, 1]
Ralstonia solanacearum: h|gik|j|f
hj|gik|abcde|f
[0, 0, 0]
Bordetella pertussis: jh|kig|edcba|f
hj|gik|abcde|f
[0, 0, 0]
Bordetella parapertussis: f|abcde|gik|jh
Bordetella bronchiseptica: jh|kig|edcba|f
[0, 0, 0]
*Xylella fastidiosa:
[0, 0, 0]
*Xanthomonas axonopodis: j|h|f|g|e
*Xanthomonas campestris: j|h|f|g|e
h|fj|e|g
[7, 0, 4]
Pseudomonas aeruginosa: jhf|g|e
[28, 0, 11]
hi|fj|e|g
[6, 0, 3]
Pseudomonas putida: edcbakjihgf
Pseudomonas syringae: i|fj|e|g
hj|e|f|g
[23, 0, 11]
Shewanella oneidensis: jh|e|f|g
hj|e|f
[0, 0, 0]
Vibrio parahaemolyticus: jh|e|f
hj|e|f|g
[16, 0, 7]
Vibrio vulnificus: hj|e|f
[0, 0, 0]
*Candidatus Blochmannia:
hj|e|f
[15, 0, 6]
*Buchnera aphidicola:
[11, 0, 6]
efj
[8, 0, 6]
Yersinia pestis: jf|e
efghj
[6, 0, 3]
Shigella flexneri: f|e|hj|g
efhj
[7, 0, 5]
Escherichia coli: abcdefghijk
[11, 0, 6]
Salmonella enterica: f|e|hj
[0, 0, 0]
*Haemophilus influenzae: j|f|g
*Pasteurella multocida: f|g
0.10

Figure 7: Ancestral gene block reconstruction of operon *paaABCDEFGHIJK* using the global reconstruction approach. Color coding is the same as in Figure 6.

paaABCDEFGHIJK. The operon *paaABCDEFGHIJK* codes for genes involved in the catabolism of phenylacetate[16]. The ability to catabolize phenylacetate varies greatly between proteobacterial species, and even among different *E. coli* K-12 strains. In contrast with *atpABCDEFG* operon which is highly conserved through many species, the operon *paaABCDEFGHIJK* is only found in full complement as an operon in some *E. coli* K-12 strains and some *Pseudomonas putida* strains The orthoblock *paaABCDE* is found in three *Bordetella* species and also in *Bradyrhizobium diazoefficiens.* The products of *paaA*, *paaB*, *paaC* and *paaE* make up the subunits of the 1,2-phenylacetyl-CoA epoxidase, and *paaD* is hypothesized to form an iron-sulfur cluster with the product of *paaE*[17]. We did not find orthologs in the endosymbionts *B. aphidicola* and *Blochmannia.*

In both the local and global reconstructions, only the ancestor of the *Bordetella* species have a combination of *paaABC* complex with *paaE.* According to Grishin *et al* [17], only this combination has full activity. In addition, the global approach only predicts gene blocks for the ancestors of α and most of γ-proteobacteria. Only the common ancestor of the *Bordetella* genus contains the cluster *paaABCE.* It has been confirmed that this cluster of genes is identical to those of *E. coli* [18]. In both approaches, gene *paaF* and *paaG* are not found to be in the same gene blocks, hence the ancestors are most likely missing the hydratase-isomerase complex.*paaJ* thiolase catalyzes two steps in the phenylacetate catabolism[19, 20, 21]. In addition, paaH is the NAD$^+$-dependent 3-hydroxyadipyl-CoA dehydrogenase involved in phenylacetate catabolism[19]. Therefore, it is reasonable that gene *paaJ* and *paaH* appear in most of the ancestral nodes that have gene blocks.

The results from the study of these operons have provided some interesting and valuable understanding of the evolution of the gene blocks. Also, in both cases, the global approach performs better in term of minimizing events. For brevity, we only provide the global ancestral reconstruction henceforth.

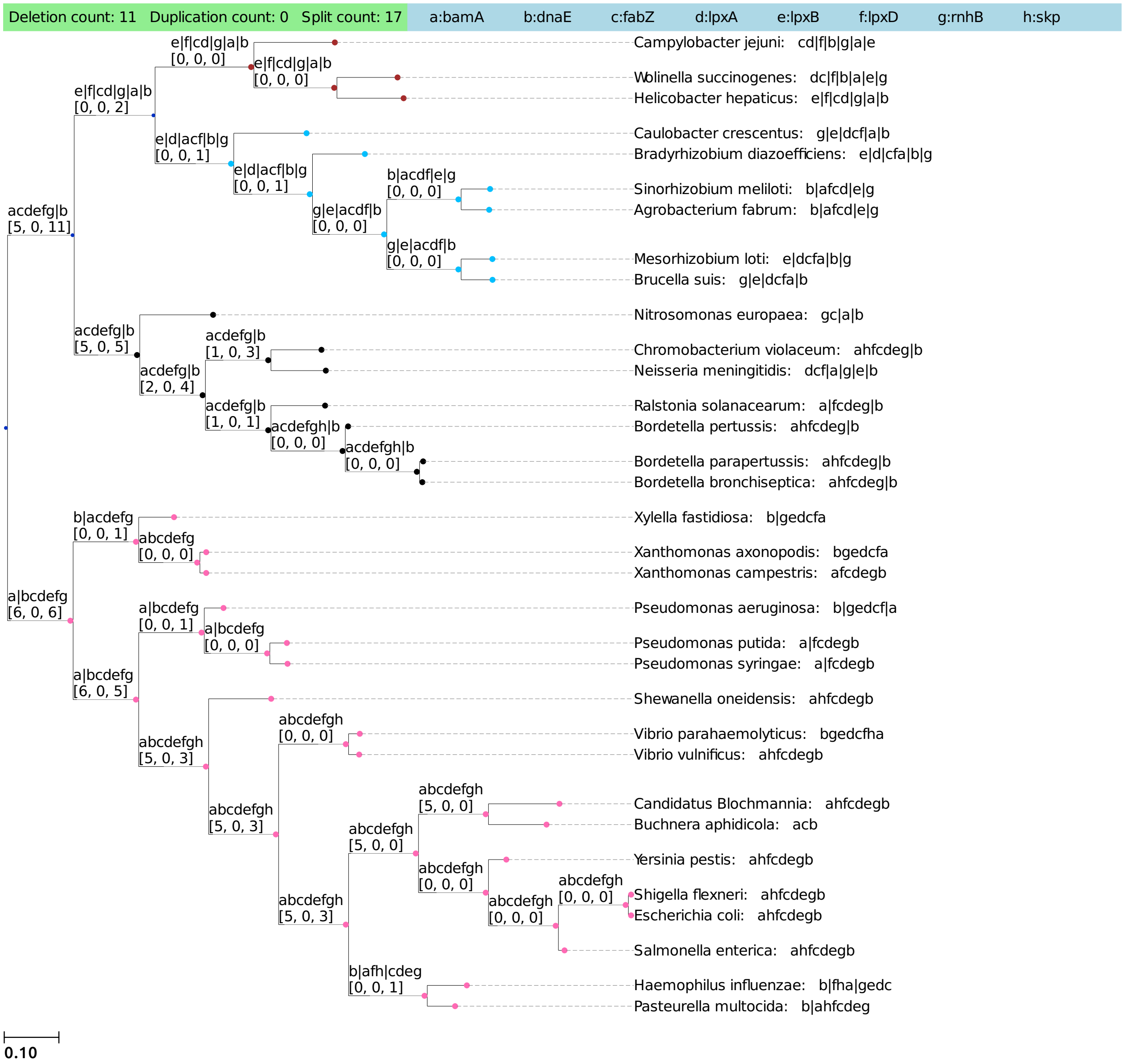
Deletion count: 11
Duplication count: 0
Split count: 17
a:bamA
b:dnaE
c:fabZ
d:lpxA
e:lpxB
f:lpxD
g:rnhB
h:skp
acdefg|b [5, 0, 11]
e|f|cd|g|a|b [0, 0, 2]
e|f|cd|g|a|b [0, 0, 0]
e|f|cd|g|a|b [0, 0, 0]
e|d|acf|b|g [0, 0, 1]
e|d|acf|b|g [0, 0, 1]
g|e|acdf|b [0, 0, 0]
b|acdf|e|g [0, 0, 0]
g|e|acdf|b [0, 0, 0]
acdefg|b [5, 0, 5]
acdefg|b [2, 0, 4]
acdefg|b [1, 0, 3]
acdefg|b [1, 0, 1]
acdefgh|b [0, 0, 0]
acdefgh|b [0, 0, 0]
a|bcdefg [6, 0, 6]
b|acdefg [0, 0, 1]
abcdefg [0, 0, 0]
a|bcdefg [6, 0, 5]
a|bcdefg [0, 0, 1]
a|bcdefg [0, 0, 0]
abcdefgh [5, 0, 3]
abcdefgh [5, 0, 3]
abcdefgh [0, 0, 0]
abcdefgh [5, 0, 3]
abcdefgh [5, 0, 0]
abcdefgh [5, 0, 0]
abcdefgh [0, 0, 0]
abcdefgh [0, 0, 0]
abcdefgh [0, 0, 0]
b|afh|cdeg [0, 0, 1]
Campylobacter jejuni: cd|f|b|g|a|e
Wolinella succinogenes: dc|f|b|a|e|g
Helicobacter hepaticus: e|f|cd|g|a|b
Caulobacter crescentus: g|e|dcf|a|b
Bradyrhizobium diazoefficiens: e|d|cfa|b|g
Sinorhizobium meliloti: b|afcd|e|g
Agrobacterium fabrum: b|afcd|e|g
Mesorhizobium loti: e|dcfa|b|g
Brucella suis: g|e|dcfa|b
Nitrosomonas europaea: gc|a|b
Chromobacterium violaceum: ahfcdeg|b
Neisseria meningitidis: dcf|a|g|e|b
Ralstonia solanacearum: a|fcdeg|b
Bordetella pertussis: ahfcdeg|b
Bordetella parapertussis: ahfcdeg|b
Bordetella bronchiseptica: ahfcdeg|b
Xylella fastidiosa: b|gedcfa
Xanthomonas axonopodis: bgedcfa
Xanthomonas campestris: afcdegb
Pseudomonas aeruginosa: b|gedcf|a
Pseudomonas putida: a|fcdegb
Pseudomonas syringae: a|fcdegb
Shewanella oneidensis: ahfcdegb
Vibrio parahaemolyticus: bgedcfha
Vibrio vulnificus: ahfcdegb
Candidatus Blochmannia: ahfcdegb
Buchnera aphidicola: acb
Yersinia pestis: ahfcdegb
Shigella flexneri: ahfcdegb
Escherichia coli: ahfcdegb
Salmonella enterica: ahfcdegb
Haemophilus influenzae: b|fha|gedc
Pasteurella multocida: b|ahfcdeg
0.10

Figure 8: Ancestral reconstrucion of gene block *bamA-skp-lpxD-fabZ-lpxAB-rnhB-dnaE* Brown: ϵ-protebacteria, blue: α-proteobacteria, black: β-proteobacteria, pink: γ-proteobacteria.

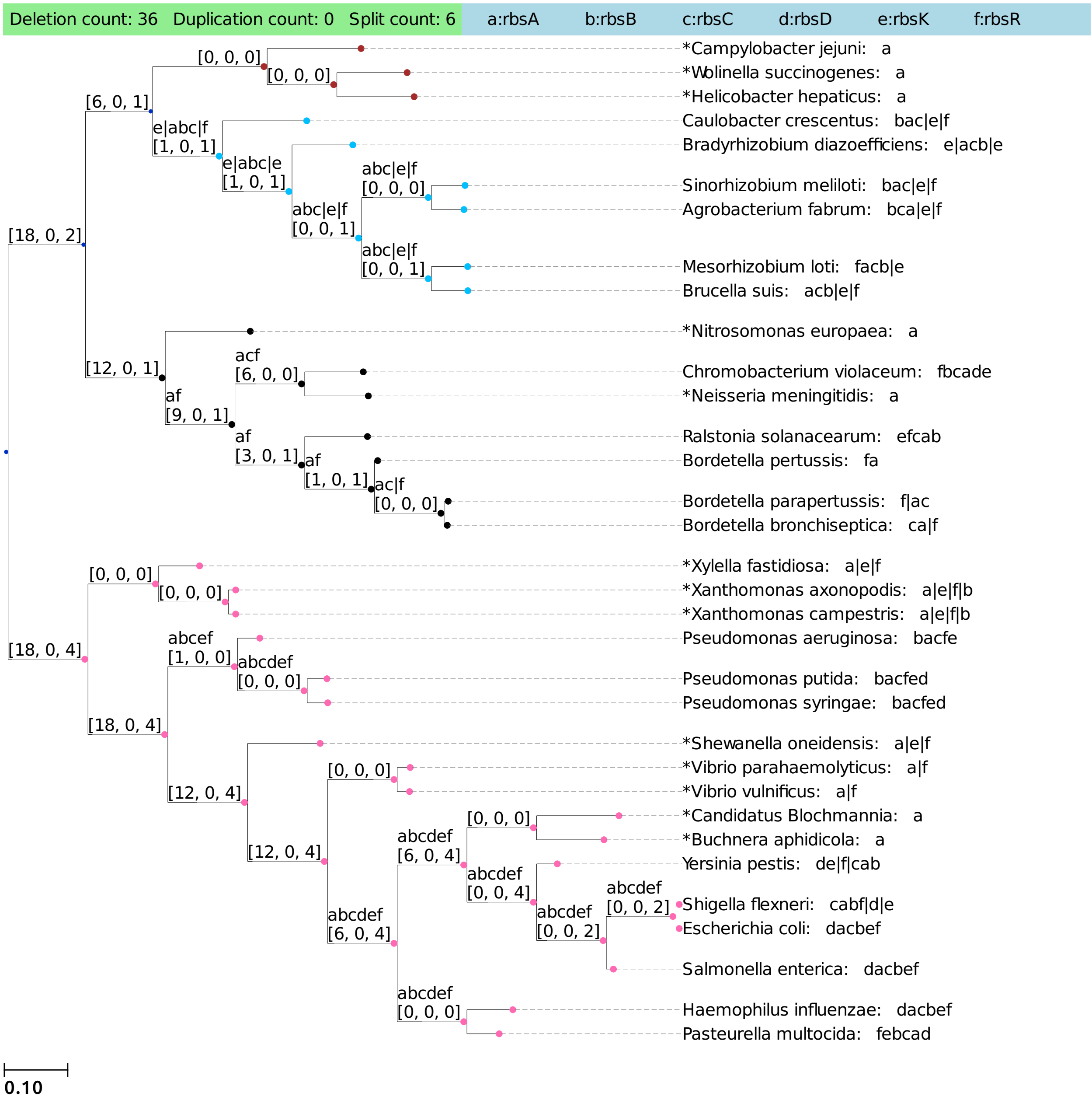
Deletion count: 36 Duplication count: 0 Split count: 6
a:rbsA b:rbsB c:rbsC d:rbsD e:rbsK f:rbsR
*Campylobacter jejuni: a
*Wolinella succinogenes: a
*Helicobacter hepaticus: a
Caulobacter crescentus: bac|e|f
Bradyrhizobium diazoefficiens: e|acb|e
Sinorhizobium meliloti: bac|e|f
Agrobacterium fabrum: bca|e|f
Mesorhizobium loti: facb|e
Brucella suis: acb|e|f
*Nitrosomonas europaea: a
Chromobacterium violaceum: fbcade
*Neisseria meningitidis: a
Ralstonia solanacearum: efcab
Bordetella pertussis: fa
Bordetella parapertussis: f|ac
Bordetella bronchiseptica: ca|f
*Xylella fastidiosa: a|e|f
*Xanthomonas axonopodis: a|e|f|b
*Xanthomonas campestris: a|e|f|b
Pseudomonas aeruginosa: bacfe
Pseudomonas putida: bacfed
Pseudomonas syringae: bacfed
*Shewanella oneidensis: a|e|f
*Vibrio parahaemolyticus: a|f
*Vibrio vulnificus: a|f
*Candidatus Blochmannia: a
*Buchnera aphidicola: a
Yersinia pestis: de|f|cab
Shigella flexneri: cabf|d|e
Escherichia coli: dacbef
Salmonella enterica: dacbef
Haemophilus influenzae: dacbef
Pasteurella multocida: febcad
[18, 0, 2]
[6, 0, 1]
[0, 0, 0]
[0, 0, 0]
e|abc|f
[1, 0, 1]
e|abc|e
[1, 0, 1]
abc|e|f
[0, 0, 1]
abc|e|f
[0, 0, 0]
abc|e|f
[0, 0, 1]
[12, 0, 1]
acf
[6, 0, 0]
af
[9, 0, 1]
af
[3, 0, 1]
af
[1, 0, 1]
ac|f
[0, 0, 0]
[18, 0, 4]
[0, 0, 0]
[0, 0, 0]
[18, 0, 4]
abcef
[1, 0, 0]
abcdef
[0, 0, 0]
[12, 0, 4]
[12, 0, 4]
[0, 0, 0]
abcdef
[6, 0, 4]
abcdef
[6, 0, 4]
[0, 0, 0]
abcdef
[0, 0, 4]
abcdef
[0, 0, 2]
abcdef
[0, 0, 2]
abcdef
[0, 0, 0]
0.10

Figure 9: Ancestral reconstruction of *rbsDACBKR*. Brown: ϵ-protebacteria; blue: α-proteobacteria; black: β-proteobacteria; pink: γ-proteobacteria.

***bamA-skp-lpxD-fabZ-lpxAB-rnhB-dnaE*.** The operon *bamA-skp-lpxD-fabZ-lpxAB-rnhB-dnaE* participates in DNA replication, repair, immune reaction, and signal transduction. It is actually a complex regulon with several promoter sites [22]. Gene *bamA* is highly conserved [23] and is required for Gram-negative outer membrane protein assembly [24, 25]. Gene *dnaE* encodes the alpha-catalytic subunit of the DNA polymerase III holoenzyme [26]. The reconstruction result has shown that those two genes have appeared in all the ancestors. Note that *bamA* is predicted to not be in the same regulatory block as the rest of the operon in γ-proteobacteria. At the same time, gene *dnaE* is not in the same block of the operon in β-proteobacteria. However, these two splits should not affect the overall operon functionality since neither *bamA* nor *dnaE* are found to form a subunit with another gene in the operon. At the same time, the cluster of *lpxD-fabZ-lpxA* is involved in lipid A biosynthesis in many bacteria[27, 28].

rbsDACBKR. The operon *rbsDACBKR* expresses genes associated with the ribose transport complex in *E. coli* [29, 30]. The *rbsABC* genes compose an ATP-dependent ribose transporter that is a member of the ATP-Binding Cassette (ABC) superfamily of transporters [31]. Mutations in each of the components eliminated transport of ribose at an external concentration of 1μM, indicating that the components make up a transport system that is responsible for high-affinity ribose transport [32]. From the reconstruction, we observe that the core gene cluster of the transporter *rbsABC* starts forming in three different inner nodes: (1) the common ancestor of α-proteobacteria; (2) γ-proteobacteria (genus *Pseudomonas*), and (3) γ-proteobacteria (*Enterobacteriaceae, Pasteurellaceae* families). The three other genes, *rbsK*, *rbsD* and *rbsR* are not essential for ribose transport. *rbsR* codes for the repressor protein which regulates the operon [33, 34]. *rbsD*, and *rbsK* are involved in the conversion of D-ribose to D-ribose 5-phosphate [35]. The gene block is most complete in the γ-proteobacteria, but the core transport genes appear also at the common ancestors of the α-proteobacteria.

4.2 Operons from *Bacillus subtilis*

B. subtilis is a Gram-positive, spore forming bacterium commonly found in soil, and is also a normal gut commensal in humans. It is a model organism for Gram-positive spore forming bacteria, and as such its genome of about 4,450 genes is well annotated. Here we used ROAGUE to reconstruct the ancestors of two *B. subtilis* gene blocks across 33 species. We selected species from the order *Bacillales* using PDA. Species from the following families were selected: *Bacillaceae* (including the reference organism *B. subtilis*), *Staphylococcae*: macrococcus and staphylococcus, *Alicyclobacillaceae*, *Listeriaceae* and *Planococcaceae.*

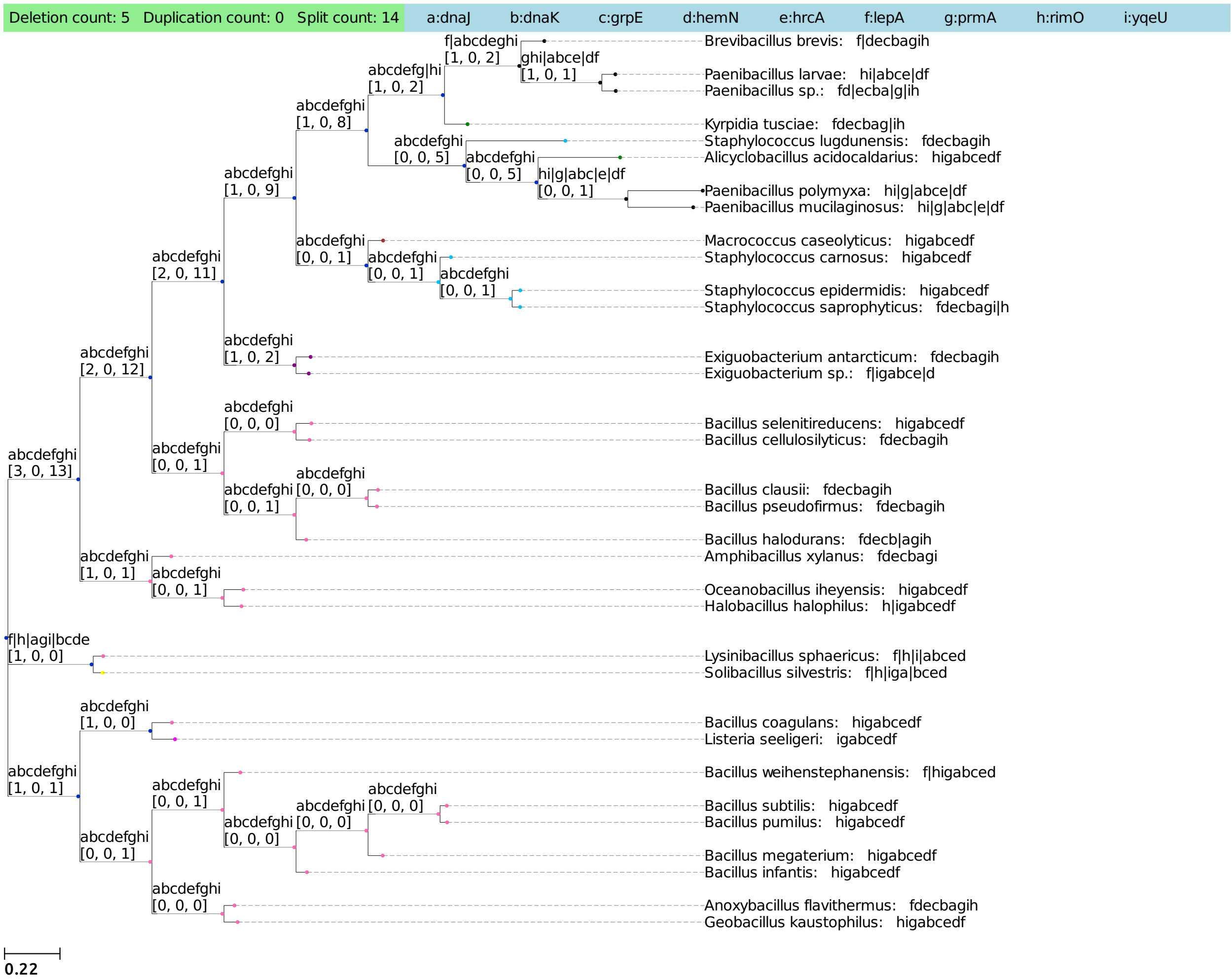
Deletion count: 5
Duplication count: 0
Split count: 14
a:dnaJ
b:dnaK
c:grpE
d:hemN
e:hrcA
f:lepA
g:prmA
h:rimO
i:yqeU
f|abcdeghi [1, 0, 2]
Brevibacillus brevis: f|decbagih
ghi|abce|df [1, 0, 1]
abcdefg|hi [1, 0, 2]
Paenibacillus larvae: hi|abce|df
Paenibacillus sp.: fd|ecba|g|ih
abcdefghi [1, 0, 8]
Kyrpidia tusciae: fdecbag|ih
abcdefghi [0, 0, 5]
Staphylococcus lugdunensis: fdecbagih
abcdefghi [0, 0, 5]
Alicyclobacillus acidocaldarius: higabcedf
abcdefghi [1, 0, 9]
hi|g|abc|e|df [0, 0, 1]
Paenibacillus polymyxa: hi|g|abce|df
Paenibacillus mucilaginosus: hi|g|abc|e|df
abcdefghi [0, 0, 1]
Macrococcus caseolyticus: higabcedf
abcdefghi [2, 0, 11]
abcdefghi [0, 0, 1]
Staphylococcus carnosus: higabcedf
abcdefghi [0, 0, 1]
Staphylococcus epidermidis: higabcedf
Staphylococcus saprophyticus: fdecbagi|h
abcdefghi [1, 0, 2]
abcdefghi [2, 0, 12]
Exiguobacterium antarcticum: fdecbagih
Exiguobacterium sp.: f|igabce|d
abcdefghi [0, 0, 0]
Bacillus selenitireducens: higabcedf
Bacillus cellulosilyticus: fdecbagih
abcdefghi [0, 0, 1]
abcdefghi [3, 0, 13]
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 1]
Bacillus clausii: fdecbagih
Bacillus pseudofirmus: fdecbagih
Bacillus halodurans: fdecb|agih
abcdefghi [1, 0, 1]
Amphibacillus xylanus: fdecbagi
abcdefghi [0, 0, 1]
Oceanobacillus iheyensis: higabcedf
Halobacillus halophilus: h|igabcedf
f|h|agi|bcde [1, 0, 0]
Lysinibacillus sphaericus: f|h|i|abced
Solibacillus silvestris: f|h|iga|bced
abcdefghi [1, 0, 0]
Bacillus coagulans: higabcedf
Listeria seeligeri: igabcedf
abcdefghi [1, 0, 1]
Bacillus weihenstephanensis: f|higabced
abcdefghi [0, 0, 1]
abcdefghi [0, 0, 0]
abcdefghi [0, 0, 0]
Bacillus subtilis: higabcedf
Bacillus pumilus: higabcedf
abcdefghi [0, 0, 1]
Bacillus megaterium: higabcedf
Bacillus infantis: higabcedf
abcdefghi [0, 0, 0]
Anoxybacillus flavithermus: fdecbagih
Geobacillus kaustophilus: higabcedf
0.22

Figure 10: Ancestor reconstruction of *lepA-hemN-hrcA-grpE-dnaKJ-yqeTUV*. Family color codes: brown: *Macrococcus*; black: *Paenibacillaceae*; blue: *Staphylococcus*; green: *Alicyclobacillaceae*; pink: *Bacillaceae*; purple: *Bacillales Family XII*; magenta: *Listeriaceae* bacteria; yellow: *Planococcaceae*.

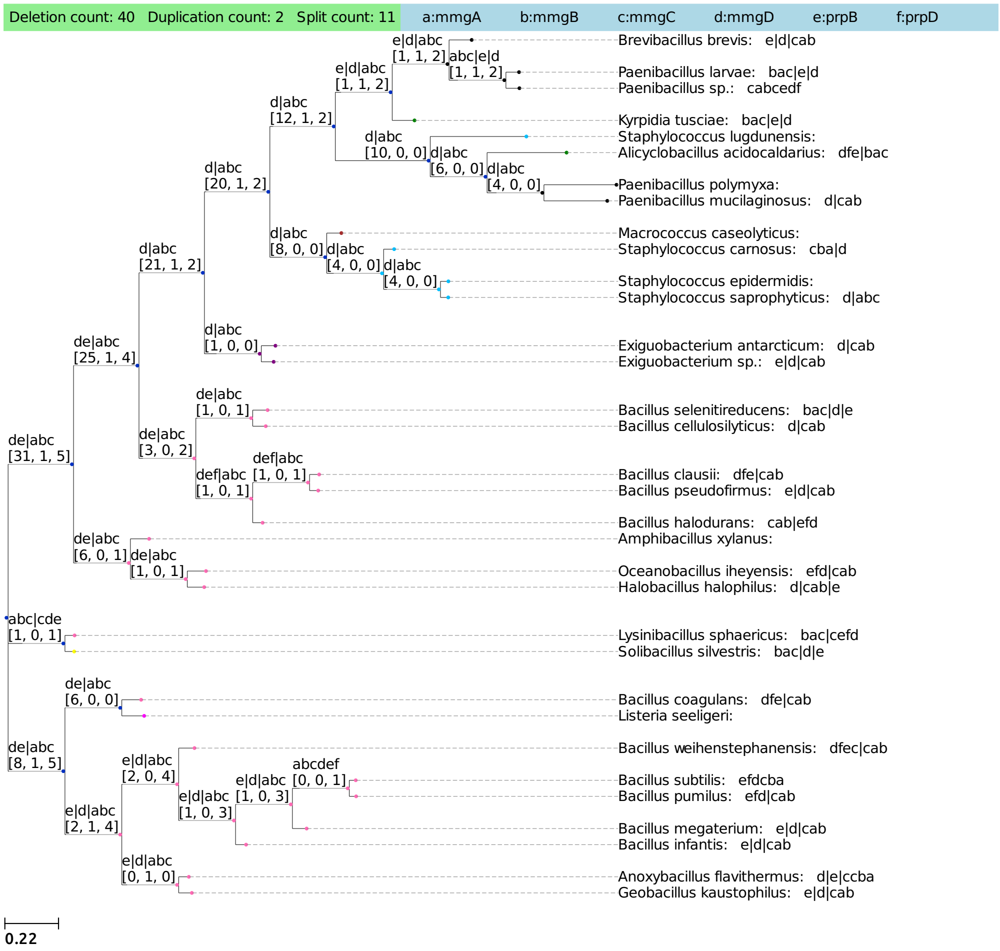
Deletion count: 40
Duplication count: 2
Split count: 11
a:mmgA
b:mmgB
c:mmgC
d:mmgD
e:prpB
f:prpD
e|d|abc [1, 1, 2]
abc|e|d [1, 1, 2]
e|d|abc [1, 1, 2]
d|abc [12, 1, 2]
d|abc [10, 0, 0]
d|abc [6, 0, 0]
d|abc [4, 0, 0]
d|abc [20, 1, 2]
d|abc [21, 1, 2]
d|abc [8, 0, 0]
d|abc [4, 0, 0]
d|abc [4, 0, 0]
d|abc [1, 0, 0]
de|abc [25, 1, 4]
de|abc [1, 0, 1]
de|abc [3, 0, 2]
def|abc [1, 0, 1]
def|abc [1, 0, 1]
de|abc [31, 1, 5]
de|abc [6, 0, 1]
de|abc [1, 0, 1]
abc|cde [1, 0, 1]
de|abc [6, 0, 0]
de|abc [8, 1, 5]
e|d|abc [2, 0, 4]
abcdef [0, 0, 1]
e|d|abc [1, 0, 3]
e|d|abc [1, 0, 3]
e|d|abc [2, 1, 4]
e|d|abc [0, 1, 0]
Brevibacillus brevis: e|d|cab
Paenibacillus larvae: bac|e|d
Paenibacillus sp.: cabcedf
Kyrpidia tusciae: bac|e|d
Staphylococcus lugdunensis:
Alicyclobacillus acidocaldarius: dfe|bac
Paenibacillus polymyxa:
Paenibacillus mucilaginosus: d|cab
Macrococcus caseolyticus:
Staphylococcus carnosus: cba|d
Staphylococcus epidermidis:
Staphylococcus saprophyticus: d|abc
Exiguobacterium antarcticum: d|cab
Exiguobacterium sp.: e|d|cab
Bacillus selenitireducens: bac|d|e
Bacillus cellulosilyticus: d|cab
Bacillus clausii: dfe|cab
Bacillus pseudofirmus: e|d|cab
Bacillus halodurans: cab|efd
Amphibacillus xylanus:
Oceanobacillus iheyensis: efd|cab
Halobacillus halophilus: d|cab|e
Lysinibacillus sphaericus: bac|cefd
Solibacillus silvestris: bac|d|e
Bacillus coagulans: dfe|cab
Listeria seeligeri:
Bacillus weihenstephanensis: dfec|cab
Bacillus subtilis: efdcba
Bacillus pumilus: efd|cab
Bacillus megaterium: e|d|cab
Bacillus infantis: e|d|cab
Anoxybacillus flavithermus: d|e|ccba
Geobacillus kaustophilus: e|d|cab
0.22

Figure 11: Ancestor reconstruction of *mmgABCDE-yqiQ*. Family color codes the same as in Figure 10

lepA-hemN-hrcA-grpE-dnaK-dnaJ-prmA-yqeU-rimO. Gene block *lepA-hemN-hrcA-grpE-dnaK-dnaJ-prmA-yqeU-rimO* facilitates the heat shock response in *B. subtilis* and the gene block *hrcA-grpE-dnaK-dnaJ* was the first identified heat shock operon within *Bacillus spp*[36]. The four genes *hrcA, grpE, dnaK, dnaJ* (e,c,b,a in Figure 10) form a tetracistronic structure, which is essential to the heat shock response role[37]. The four genes are proximal in all the species examined, and form the core of the orthoblock. Overall, this operon is quite conserved, and the ancestral reconstructions are highly similar to the reference operon.

mmgABCDE-prpB. The operon *mmgABCDE-prpB* is expressed during endosporulation [38]. Subunit *mmgABC*'s breakdown of fatty acids is a mean for attaining energy to drive the cell's preparation for dormancy [39]. Hence, it is reasonable to see that the common ancestor has this subunit. In addition, gene *mmgD* and gene *prpB/yqiQ* are predicted to be proximal. Several studies predicted that gene *mmgD*, *prpB*, and *prpD* encode the proteins of the putative methylcitrate shunt [40]. However, they did not specify if deletion mutations might contribute to a defect of the functionality.

5 Conclusions

We developed ROAGUE, a method for the reconstruction of ancestral gene blocks using maximum parsimony. ROAGUE accepts a set of bacterial genomes, a species tree, and a reference gold-standard orthoblock. ROAGUE then identifies the orthoblocks to the gold-standard genome in all the species provided, using the best-orthoblock identification method developed in [4]. ROAGUE then proceeds to reconstruct the ancestral genomes using local or global parsimony. ROAGUE's output contains the species tree with the extant orthoblocks and the reconstructed orthoblocks. We provided several examples of ancestral gene block reconstructions based on reference operons in *E. coli* and *B. subtilis*.

A few interesting observations emerge regarding conservation and ancestry of operons. It appears that essentiality (the trait of being essential to life) and the formation of a protein complex are the main drivers for gene block conservation. This is most apparent in the *atp* operon coding for F_1F_o-atpase in proteobacteria. There are few evolutionary events identified in the *atp* operon ancestry. The ribose transporter block also seems to preserve the core ribose transporter (*rbsABC*), while not the ribose phosphorylation genes *rbsD* and *rbsK*.

ROAGUE does not account for horizontal gene transfer, which is considered to be a major driver in operon evolution[7]. This can ostensibly be dealt with by reconciling a species tree with an operon tree, in the same way that phylogenomic analyses do for gene trees and species trees[41]. In addition, the gene order in a gene block is ignored. While the relationship between gene organization and expression in operons is not well understood, it is clear from several studies that gene order does have an effect on expression and on the functionality of the operon in general (e.g.[42, 43, 44]). Adding the parameters of horizontal gene transfer, gene order preservation, or both to ROAGUE would be highly valuable. We invite the community to contribute to ROAGE, as well as use the tool for phylogenetic analyses of bacterial gene blocks.

Acknowledgements

I.F. was supported, in part, by National Science Foundation award ABI-1551363. O.E. was supported, in part, by National Science Foundation award CCF-1617626.

References

[1] John T. Bonner. *The Evolution of Complexity by Means of Natural Selection.* Princeton University Press, 1988.

[2] Christoph Adami, Charles Ofria, and Travis C. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463–4468, April 2000.

[3] Michael Lynch and John S. Conery. The origins of genome complexity. *Science*, 302(5649):1401–1404, November 2003.

[4] David C Ream, Asma R Bankapur, and Iddo Friedberg. An event-driven approach for studying gene block evolution in bacteria. *Bioinformatics*, 31(13):2075–2083, 2015.

[5] N. H. Horowitz. On the evolution of biochemical syntheses. *Proceedings of the National Academy of Sciences of the United States of America*, 31(6):153–157, June 1945.

[6] F. W. Stahl and N. E. Murray. The evolution of gene clusters and genetic circularity in microorganisms. *Genetics*, 53(3):569–576, March 1966.

[7] Marina Omelchenko, Kira Makarova, Yuri Wolf, Igor Rogozin, and Eugene Koonin. Evolution of mosaic operons by horizontal gene transfer and gene displacement in situ. *Genome Biology*, 4(9):R55+, 2003.

[8] J. G. Lawrence and J. R. Roth. Selfish operons: horizontal transfer may drive the evolution of gene clusters. *Genetics*, 143(4):1843–1860, August 1996.

[9] Heladia Salgado, Socorro Gama-Castro, Martin Peralta-Gil, Edgar Díaz-Peredo, Fabiola Sánchez-Solano, Alberto Santos-Zavaleta, Irma Martínez-Flores, Verónica Jiménez-Jacinto, César Bonavides-Martínez, Juan Segura-Salazar, et al. Regulondb (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic acids research*, 34(suppl 1):D394–D397, 2006.

[10] Olga Chernomor, Bui Quang Minh, Félix Forest, Steffen Klaere, Travis Ingram, Monika Henzinger, and Arndt Haeseler. Split diversity in constrained conservation prioritization using integer linear programming. *Methods in Ecology and Evolution*, 6(1):83–91, 2015.

[11] Daniel P Faith. Conservation evaluation and phylogenetic diversity. *Biological conservation*, 61(1):1–10, 1992.

[12] Elvin Kasimoglu, Soon-Jung Park, Joel Malek, Ching Ping Tseng, and Robert P Gunsalus. Transcriptional regulation of the proton-translocating atpase (atpibefhagdc) operon of escherichia coli: control by cell growth rate. *Journal of bacteriology*, 178(19):5563–5567, 1996.

[13] A E Senior. The proton-translocating atpase of escherichia coli. *Annual Review of Biophysics and Biophysical Chemistry*, 19(1):7–41, 1990. PMID: 2141983.

[14] Gay NJ. Construction and characterization of an escherichia coli strain with a unci mutation. *Journal of Bacteriology*, 158:820–825, 1984.

[15] Vassiliki Lila Koumandou and Sophia Kossida. Evolution of the f 0 f 1 atp synthase complex in light of the patchy distribution of different bioenergetic pathways across prokaryotes. *PLoS Comput Biol*, 10(9):e1003821, 2014.

[16] Fergal J Martin and James O McInerney. Recurring cluster and operon assembly for phenylacetate degradation genes. *BMC evolutionary biology*, 9(1):36, 2009.

[17] Andrey M. Grishin, Eunice Ajamian, Limei Tao, Linhua Zhang, Robert Menard, and Miroslaw Cygler. Structural and functional studies of the escherichia coli phenylacetyl-coa monooxygenase complex. *Journal of Biological Chemistry*, 286(12):10735–10743, 2011.

[18] Jose M Luengo, Jose L Garcia, and Elias R Olivera. The phenylacetyl-coa catabolon: a complex catabolic unit with broad biotechnological applications. *Molecular microbiology*, 39(6):1434–1442, 2001.

[19] Wael Ismail, Magdy El-Said Mohamed, Barry L. Wanner, Kirill A. Datsenko, Wolfgang Eisenreich, Felix Rohdich, Adelbert Bacher, and Georg Fuchs. Functional genomics by nmr spectroscopy. *European Journal of Biochemistry*, 270(14):3047–3054, 2003.

[20] R. Teufel, V. Mascaraque, W. Ismail, M. Voss, J. Perera, W. Eisenreich, W. Haehnel, and G. Fuchs. Bacterial phenylalanine and phenylacetate catabolic pathway revealed. *Proceedings of the National Academy of Sciences*, 107(32):14390–14395, 2010.

[21] Juan Nogales, Raffaella Macchi, Federico Franchi, Dagania Barzaghi, Cristina Fernandez, Jose L Garcia, Giovanni Bertoni, and Eduardo Diaz. Characterization of the last step of the aerobic phenylacetic acid degradation pathway. *Microbiology*, 153(2):357–365, 2007.

[22] Rubing Liang and Jianhua Liu. In-frame deletion of escherichia coli essential genes in complex regulon. *BioTechniques*, 44(2):209–10, 2008.

[23] Ian Gentle, Kipros Gabriel, Peter Beech, Ross Waller, and Trevor Lithgow. The omp85 family of proteins is essential for outer membrane biogenesis in mitochondria and bacteria. *The Journal of cell biology*, 164(1):19–24, 2004.

[24] William T Doerrler and Christian RH Raetz. Loss of outer membrane proteins without inhibition of lipid export in an escherichia coli yaet mutant. *Journal of Biological Chemistry*, 280(30):27679–27687, 2005.

[25] John Werner and Rajeev Misra. Yaet (omp85) affects the assembly of lipid-dependent and lipid-independent outer membrane proteins of escherichia coli. *Molecular microbiology*, 57(5):1450–1459, 2005.

[26] H Maki and Arthur Kornberg. The polymerase subunit of dna polymerase iii of escherichia coli. ii. purification of the alpha subunit, devoid of nuclease activities. *Journal of Biological Chemistry*, 260(24):12987–12992, 1985.

[27] MB Schmid, N Kapur, DR Isaacson, P Lindroos, and C Sharpe. Genetic analysis of temperature-sensitive lethal mutants of salmonella typhimurium. *Genetics*, 123(4):625–633, 1989.

[28] Sandhya Mohan, Theresa M Kelly, Suzanne S Eveland, CR Raetz, and Matt S Anderson. An escherichia coli gene (fabz) encoding (3r)-hydroxymyristoyl acyl carrier protein dehydrase. relation to faba and suppression of mutations in lipid a biosynthesis. *Journal of Biological Chemistry*, 269(52):32896–32903, 1994.

[29] Jelena Zaitseva, Huide Zhang, R Alan Binnie, and Mark Hermodson. The proteins encoded by the rbs operon of escherichia coli: Ii. use of chimeric protein constructs to isolate and characterize rbsc. *Protein science*, 5(6):1100–1107, 1996.

[30] Charlene F Barroga, Huide Zhang, Nadeem Wajih, James H Bouyer, and Mark A Hermodson. The proteins encoded by the rbs operon of escherichia coli: I. overproduction, purification, characterization, and functional analysis of rbsa. *Protein science*, 5(6):1093–1099, 1996.

[31] Yongkyu Park and Chankyu Park. Topology of rbsc, a membrane component of the ribose transporter, belonging to the arah superfamily. *Journal of bacteriology*, 181(3):1039–1042, 1999.

[32] A Iida, S Harayama, T Iino, and GL Hazelbauer. Molecular cloning and characterization of genes required for ribose transport and utilization in escherichia coli k-12. *Journal of bacteriology*, 158(2):674–682, 1984.

[33] Tomohiro Shimada, Ayako Kori, and Akira Ishihama. Involvement of the ribose operon repressor rbsr in regulation of purine nucleotide synthesis in escherichia coli. *FEMS microbiology letters*, 344(2):159–165, July 2013.

[34] C. A. Mauzy and M. A. Hermodson. Structural and functional analyses of the repressor, rbsr, of the ribose operon of escherichia coli. *Protein science : a publication of the Protein Society*, 1(7):831–842, July 1992.

[35] Hyangee Oh, Yongkyu Park, and Chankyu Park. A mutated ptsg, the glucose transporter, allows uptake ofd-ribose. *Journal of Biological Chemistry*, 274(20):14006–14011, 1999.

[36] M Wetzstein, U Völker, J Dedio, S Löbau, U Zuber, M Schiesswohl, C Herget, M Hecker, and W Schumann. Cloning, sequencing, and molecular analysis of the dnak locus from bacillus subtilis. *Journal of Bacteriology*, 174(10):3300–3310, 1992.

[37] Georg Homuth, Shoko Masuda, Axel Mogk, Yasuo Kobayashi, and Wolfgang Schumann. The dnak operon of bacillus subtilis is heptacistronic. *Journal of bacteriology*, 179(4):1153–1164, 1997.

[38] Rejwi Acharya. *Overexpression, purification, and characterization of MmgD from Bacillus subtilis strain 168*. PhD thesis, University of North Carolina at Greensboro, 2009.

[39] Amy L Quattlebaum. *Characterization of Biosynthetic and Catabolic Pathways of Bacillus Subtilis Strain 168*. PhD thesis, University of North Carolina at Greensboro, 2009.

[40] Birgit Voigt, Le Thi Hoi, Britta Jürgen, Dirk Albrecht, Armin Ehrenreich, Birgit Veith, Stefan Evers, Karl-Heinz Maurer, Michael Hecker, and Thomas Schweder. The glucose and nitrogen starvation response of bacillus licheniformis. *Proteomics*, 7(3):413–423, 2007.

[41] J. A. Eisen. A phylogenomic study of the muts family of proteins. *Nucleic acids research*, 26(18):4291–4300, September 1998.

[42] Ayaka Hiroe, Kenji Tsuge, Christopher T. Nomura, Mitsuhiro Itaya, and Takeharu Tsuge. Rearrangement of gene order in the phacab operon leads to effective production of ultrahigh-molecular-weight poly[(r)-3-hydroxybutyrate] in genetically engineered escherichia coli. *Applied and Environmental Microbiology*, 78(9):3177–3184, May 2012.

[43] Jonathan N. Wells, L. Therese Bergendahl, and Joseph A. Marsh. Operon gene order is optimized for ordered protein complex assembly. *Cell reports*, 14(4):679–685, February 2016.

[44] Han N. Lim, Yeong Lee, and Razika Hussein. Fundamental relationship between operon organization and gene expression. *Proceedings of the National Academy of Sciences*, 108(26):10626–10631, June 2011.

6 Supplementary Material

6.1 Local Optimum

6.1.1 Correctness

Let $\hat{\lambda} := Algorithm\ 1(T, G, \Omega, \lambda)$. For each $u \in I(T)$, let u_1, u_2 be its children. Let O, O_1, O_2 respectively be the orthoblock assigned to u, u_1, u_2 by function $\hat{\lambda}$. We will show that our results minimize $d_d(O, O_1) + d_d(O, O_2)$ and $d_u(O, O_1) + d_u(O, O_2)$

Lemma 1: $\forall g \in G$, if $FREQ_g(u) \geq .5$ then either $FREQ_g(u_1) \geq .5$ or $FREQ_g(u_2) \geq .5$ In addition, if $FREQ_g(u) < .5$ then either $FREQ_g(u_1) < .5$ or $FREQ_g(u_2) < .5$
Proof:

1. If $FREQ_g(u) \geq .5$ then either $FREQ_g(u_1) \geq .5$ or $FREQ_g(u_2) \geq .5$
   Assume that $FREQ_g(u_1) < .5$ and $FREQ_g(u_2) < .5$, then

$$\begin{cases} |\{v \in HasLeaf(u_1) | g \in Gene(\lambda(v)\}| < \frac{|HasLeaf(u_1)|}{2} \\ |\{v \in HasLeaf(u_2) | g \in Gene(\lambda(v)\}| < \frac{|HasLeaf(u_2)|}{2} \end{cases}$$

$$\rightarrow \big|\{v \in (HasLeaf(u_1) \cup HasLeaf(u_2)) | g \in Gene(\lambda(v)\}\big| < \frac{|HasLeaf(u_1)|}{2} + \frac{|HasLeaf(u_2)|}{2}$$

   Since u_1, u_2 are the children of u, then

$$\begin{cases} HasLeaf(u_1) \cup HasLeaf(u_2) = HasLeaf(u) \\ HasLeaf(u_1) \cap HasLeaf(u_2) = \emptyset \end{cases}$$

$$\rightarrow \big|\{v \in HasLeaf(u) | g \in Gene(\lambda(v)\}\big| < \frac{|HasLeaf(u)|}{2}$$
$$\rightarrow FREQ_g(u) < .5$$

   By contraposition, if $FREQ_g(u) \geq .5$ then either $FREQ_g(u_1) \geq .5$ or $FREQ_g(u_2) \geq .5$

2. If $FREQ_g(u) < .5$ then either $FREQ_g(u_1) < .5$ or $FREQ_g(u_2) < .5$
   We can prove it using the same logic as above.

Lemma 2: $\forall g \in G$, if $g \in Gene(O)$ *and* $g \notin Gene(O')$, then
$|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| \leq |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$.
Proof:
Since $g \in Gene(O)$, then $FREQ_g(u) \geq .5$. Therefore, $FREQ_g(u_1) \geq .5$ *or* $FREQ_g(u_2) \geq .5$ (by lemma 1). Hence, $g \in Gene(u_1)$ *or* $g \in Gene(u_2)$. Consider 3 cases:

1. If u_1 and u_2 contain g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |1 - 1| + |1 - 1| = 0$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |0 - 1| + |0 - 1| = 2$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| < |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

2. If only u_1 contains g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |1 - 1| + |1 - 0| = 1$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |0 - 1| + |0 - 0| = 1$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

3. If only u_2 contains g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |1 - 0| + |1 - 1| = 1$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |0 - 0| + |0 - 1| = 1$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

From the above cases, we conclude that
$|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| \leq |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

Lemma 3: $\forall g \in G$, if $g \notin Gene(O)$ *and* $g \in Gene(O')$, then
$|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| \leq |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$.
Proof:
Since $g \notin Gene(O)$, then $FREQ_g(u) < .5$. Therefore, $FREQ_g(u_1) < .5$ *or* $FREQ_g(u_2) < .5$ (by lemma 1. Hence, $g \notin Gene(u_1)$ *or* $g \notin Gene(u_2)$. Consider 3 cases:

1. If u_1 and u_2 do not contain g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |0 - 0| + |0 - 0| = 0$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |1 - 0| + |1 - 0| = 2$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| < |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

2. If only u_1 does not contain g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |0 - 0| + |0 - 1| = 1$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |1 - 0| + |1 - 1| = 1$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

3. If only u_2 does not contain g, then
   $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |1 - 1| + |1 - 0| = 1$
   $|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)| = |0 - 1| + |0 - 0| = 1$
   Therefore, $|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| = |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

From the above cases, we conclude that
$|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)| \leq |I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|$

1. Minimal deletions: Given an assignment of orthoblock O' to u, we will show that
   $d_d(O', O_1) + d_d(O', O_2) \geq d_d(O, O_1) + d_d(O, O_2)$

   Proof:

$$\begin{aligned}
d_d(O', O_1) + d_d(O', O_2) &= \sum_g (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) \\
&= \sum_{g \in O'} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) + \\
&\sum_{g \notin O'} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) \\
&= \sum_{g \in O', g \in O} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) + \\
&\sum_{g \in O', g \notin O} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) + \\
&\sum_{g \notin O', g \in O} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) + \\
&\sum_{g \notin O', g \notin O} (|I_g(O') - I_g(O_1)| + |I_g(O') - I_g(O_2)|) \\
&\geq \sum_{g \in O', g \in O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) + \\
&\sum_{g \in O', g \notin O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) + \\
&\sum_{g \notin O', g \in O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) + \\
&\sum_{g \notin O', g \notin O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) \\
&= \sum_{g \in O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) + \\
&\sum_{g \notin O} (|I_g(O) - I_g(O_1)| + |I_g(O) - I_g(O_2)|) \\
&= d_d(O, O_1) + d_d(O, O_2)
\end{aligned} \tag{1}$$

2. Minimal duplication:

   Proof: Applying the same idea as the above proof with $DUP_g(u), Dup(u)$ instead of $FREQ_g(u), Gene(u)$, we will achieve same result.

6.1.2 Runtime

The main challenge is how to store the the data of $FREQ_g(v), HasLeaf(v)$ for each inner node v. This can be done with dynamic programming. Algorithm 1 runs in polynomial time. Together, the algorithm takes $O(m^2) \times O(n) = O(m^2 \times n)$ with n is the number of leaf nodes, and m as the number of genes in the reference orthoblock.

6.2 Global Optimum

6.2.1 Correctness

Let $\hat{\lambda} := Algorithm\ 2(T, G, \Omega, \lambda)$. We will show that $d_d(\hat{\lambda}) := \sum_{(u,v) \in E}(d_d(u, v))$ and $d_u(\hat{\lambda}) := \sum_{(u,v) \in E}(d_u(u, v))$ are minimal.

1. Minimal deletions:

As stated above, $d_d(O, O') := |\sum_g (I_g(O) - I_g(O'))|$. Therefore, we can rewrite out global deletion cost as:

$$d_d(\hat{\lambda}) := \sum_{(u,v)\in E} (d_d(u,v)) = \sum_{(u,v)\in E} (|\sum_g (I_g(\hat{\lambda}(u)) - I_g(\hat{\lambda}(v)))|)$$

Since each gene occurrence within a gene block is independent from each other, we only need to show that our algorithm provide a global minimum deletion for any genes g. Our algorithm is based on Fitch algorithm, and the proof can be followed by the conventional proof of Fitch easily.

2. Minimal duplications:
*Proof:*Applying the same idea as the above proof with $DUP_g(u), Dup(u)$ instead of $FREQ_g(u), Gene(u)$, we will achieve same result.

6.2.2 Run Time

This algorithm is twice as slow as the algorithm 1. The reason is that it has to traverse the tree twice, in post order and level order. However,it still takes $O(m^2 \times n)$ to finish.