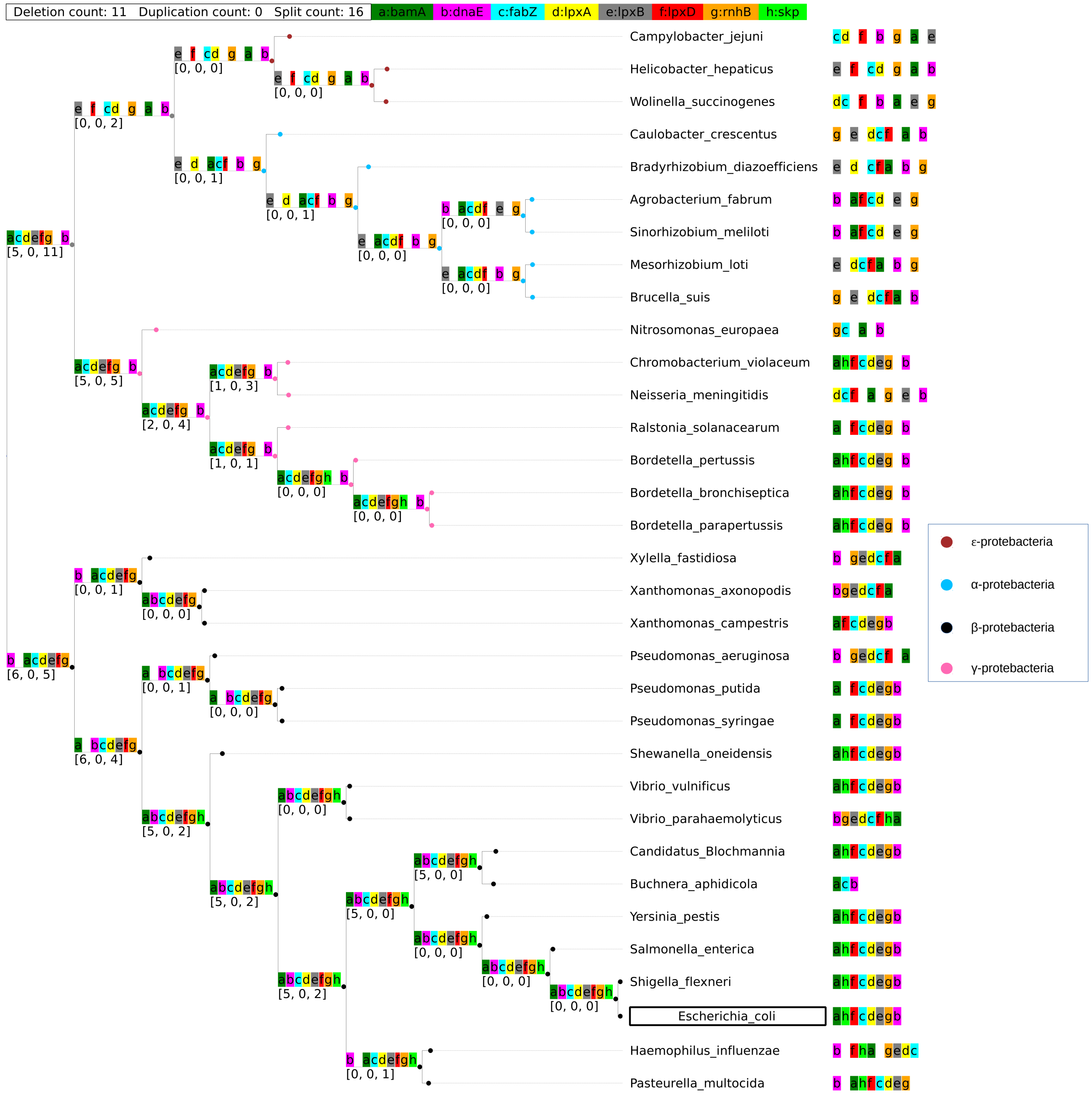
Supplementary Materials

Here we provide an analysis of two additional gene blocks, more detail about the local and global maximum parsimony algorithms, the correctness and proof of the two algorithms discussed in this work.

Additional Gene Blocks

Here are the ancestral reconstructions of two more gene blocks in *E. coli*.



Deletion count: 11 Duplication count: 0 Split count: 16
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]
e acdf b g
[0, 0, 0]
b acdf e g
[0, 0, 0]
e acdf b g
[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]
b acdefg
[6, 0, 5]
b acdefg
[0, 0, 1]
abcdefg
[0, 0, 0]
a bcdefg
[6, 0, 4]
a bcdefg
[0, 0, 1]
a bcdefg
[0, 0, 0]
abcdefgh
[5, 0, 2]
abcdefgh
[5, 0, 2]
abcdefgh
[0, 0, 0]
abcdefgh
[5, 0, 2]
abcdefgh
[5, 0, 0]
abcdefgh
[5, 0, 0]
abcdefgh
[0, 0, 0]
abcdefgh
[0, 0, 0]
abcdefgh
[0, 0, 0]
b acdefgh
[0, 0, 1]
Campylobacter_jejuni cd f b g a e
Helicobacter_hepaticus e f cd g a b
Wolinella_succinogenes dc f b a e g
Caulobacter_crescentus g e dcf a b
Bradyrhizobium_diazoefficiens e d cfa b g
Agrobacterium_fabrum b afcd e g
Sinorhizobium_meliloti 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_bronchiseptica ahfcdeg b
Bordetella_parapertussis 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_vulnificus ahfcdegb
Vibrio_parahaemolyticus bgedcfha
Candidatus_Blochmannia ahfcdegb
Buchnera_aphidicola acb
Yersinia_pestis ahfcdegb
Salmonella_enterica ahfcdegb
Shigella_flexneri ahfcdegb
Escherichia_coli ahfcdegb
Haemophilus_influenzae b fha gedc
Pasteurella_multocida b ahfcdeg
ε-protebacteria
α-protebacteria
β-protebacteria
γ-protebacteria

Fig S1 – Ancestral reconstrucion of gene block *bamA-skp-lpxD-fabZ-lpxAB-rnhB-dnaE*

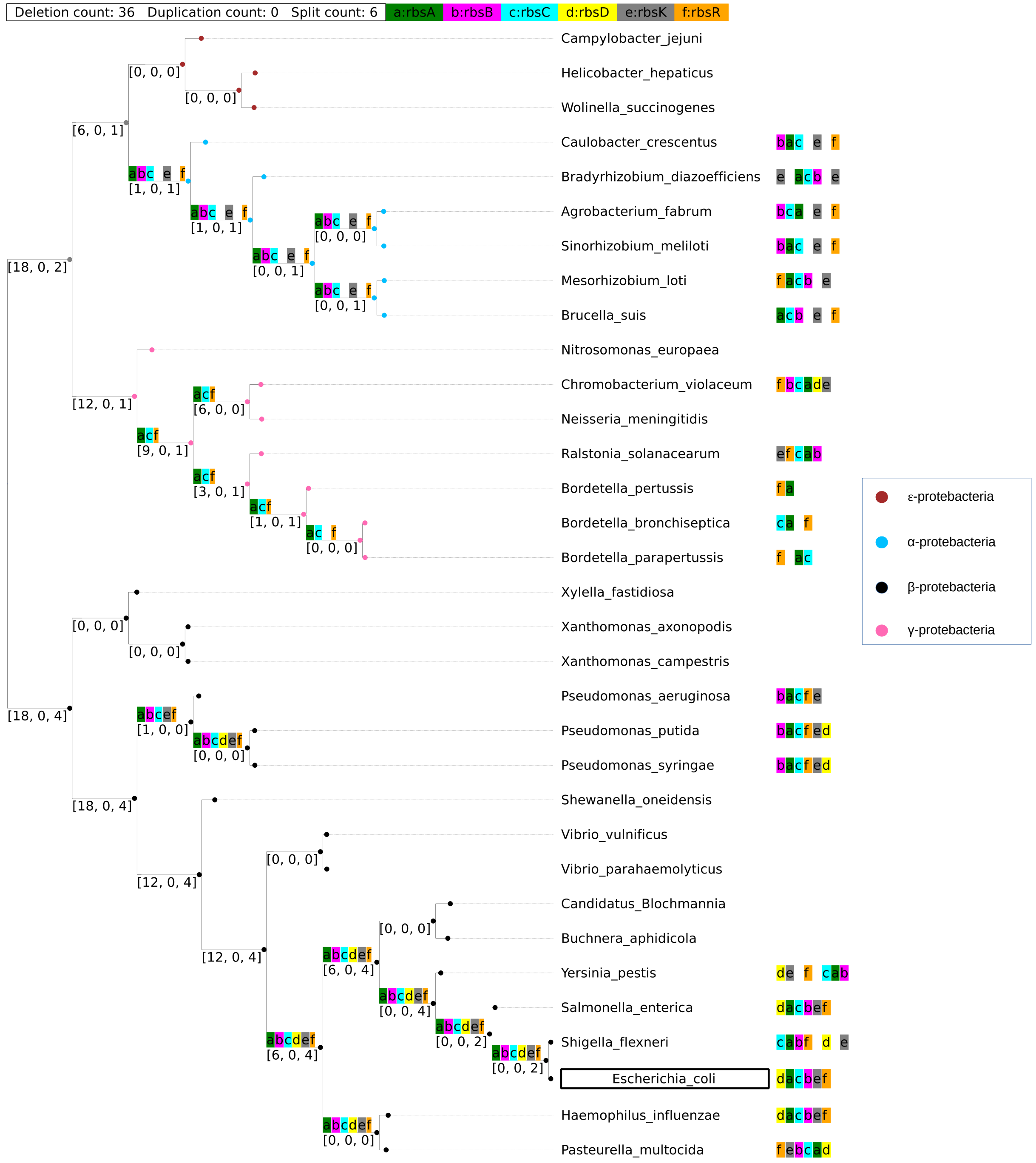


Fig S2 – Ancestral reconstruction of *rbsDACBKR*

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 Liang and Liu (2008). Gene *bamA* is highly conserved Gentle *et al.* (2004) and is required for Gram-negative outer membrane protein assembly Doerrler and Raetz (2005); Werner and Misra (2005). Gene *dnaE* encodes the alpha-catalytic subunit of the DNA polymerase III holoenzyme Maki and Kornberg (1985). 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 bacteriaSchmid *et al.* (1989); Mohan *et al.* (1994).

rbsDACBKR. The operon *rbsDACBKR* expresses genes associated with the ribose transport complex in *E. coli* Zaitseva *et al.* (1996); Barroga *et al.* (1996). The *rbsABC* genes compose an ATP-dependent ribose transporter that is a member of the ATP-Binding Cassette (ABC) superfamily of transporters Park and Park (1999). 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 Iida *et al.* (1984). 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 Shimada *et al.* (2013); Mauzy and Hermodson (1992). *rbsD*, and *rbsK* are involved in the conversion of D-ribose to D-ribose 5-phosphate Oh *et al.* (1999). 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.

Local Maximum Parsimony

Because the three distance measures are interdependent, the local parsimony problem is not trivial. In the following example, we demonstrate why it is difficult to infer a parent from children in the most parsimonious way.

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, including all genes from S in O gives us deletion distance: $d_d(O, O_1) + d_d(O, O_2) = 4$, 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 to include 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 of O_1, O_2 to O respectively become:

$$Rel(O_1,O) : b|cd|ef$$
$$Rel(O_2,O) : bc|de|fb|f|f$$

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 to u become:

$$\mathit{Rel(O_1,O)} : b|cd|e$$
$$\mathit{Rel(O_2,O)} : bc|de|b$$

Hence, by setting our gene block O as either $Rel(O_1, O)$ or $Rel(O_2, O)$, the deletion distance increased by 2 because we excluded gene f which is in S; however, 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 $Gene(O)$, this will not increase the deletion distance. The relevant gene blocks of the two children to u become:

$$\mathit{Rel(O_1,O)} : b|cd|ef|g$$
$$\mathit{Rel(O_2,O)} : bc|de|fb|f|f$$

By setting $O := b|cd|ef|g$, the new split distance is $d_s(O, O_1) + d_s(O, O_2) = 0 + 1 = 1$ 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 set $Gene(O_1) \triangle Gene(O_2)$, which will take exponential time. We therefore provide a heuristic approach that guaranteed minimum deletion and duplication distances, but not split distances. We present a greedy local optimization algorithm as follows. See Figure 3 in the manuscript for a visualization of the process.

```
Input: $T, G, \Omega, \lambda$
Result: $\hat{\lambda}$
for internal node $u$ when traversing $T$ in post-order do
    Let $u_1$, $u_2$ be the children of $u$
    Let $O_1 := \lambda(u_1)$, $O_2 := \lambda(u_2)$
    $initial := GeneBlock(O_1) \cup GeneBlock(O_2)$
    $initial_{gene} := \{g | FREQ_g(u) > .5\}$
    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 $b$
        $U_{1_G} = U_{1_G} \cup b$;
    if $|initial| < |U_{1_G}|$ or $|initial| > |U_{1_G}|$ then
        $initial := U_{1_G}$
    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 $b$
    $\hat{\lambda}(u) := initial$
Return $\hat{\lambda}$
```

Algorithm 1: Local Maximum Parsimony

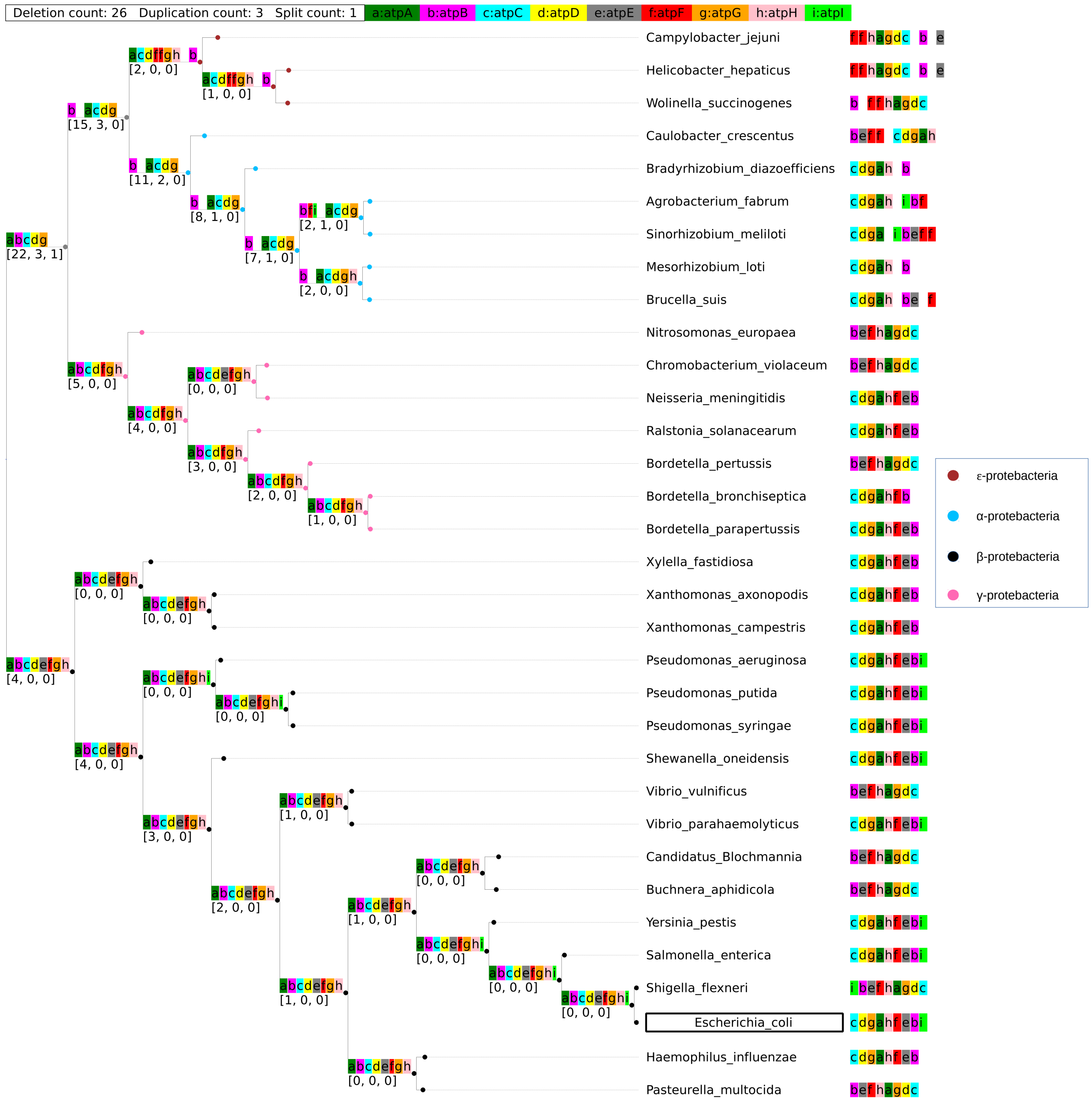


Fig S3 – Ancestral reconstruction of operon *atpIBEFHAGDC* using the local optimization approach.

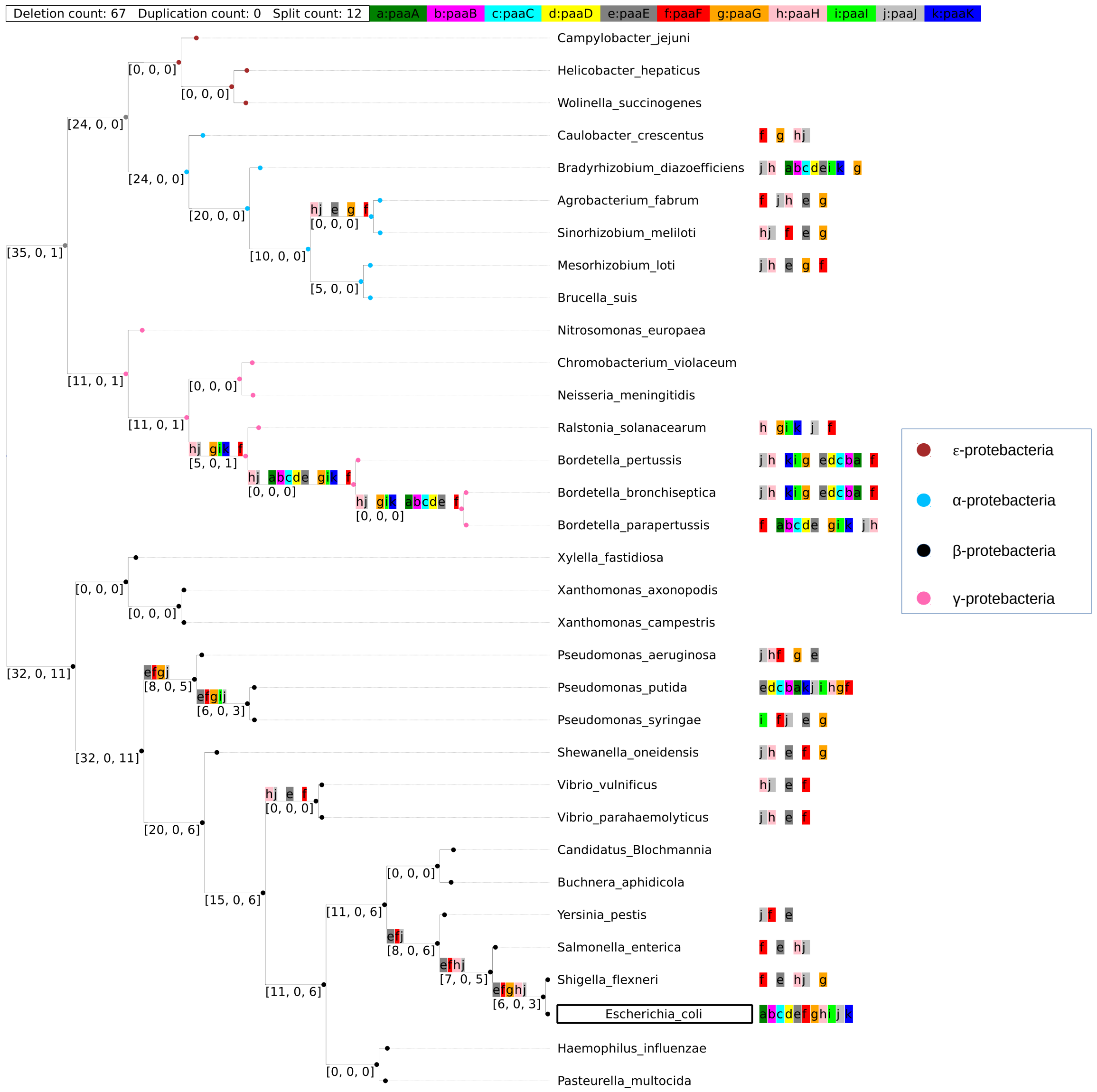


Fig S4 – Ancestral gene block reconstruction of *paaABCDEFGHIJK* using the local reconstruction approach. Asterisks in front of species names indicate that a minimal orthoblock (which should consist of two or more proximal genes that are orthologs to genes in the reference operon) was not found.

Global Approach

Here, we present the global approach algorithm.

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

Algorithm 2: Global approach

Correctness and Proof: Local Optimum

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 $FREQ_g(u) > .5$ then either $FREQ_g(u_1) > .5$ or $FREQ_g(u_2) > .5$ In addition, if $FREQ_g(u) \leq .5$ then either $FREQ_g(u_1) \leq .5$ or $FREQ_g(u_2) \leq .5$
Proof:

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

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

   Define $H := \{v \in (HasLeaf(u_1) \cup HasLeaf(u_2)) | g \in Gene(\lambda(v)\}$, from the two inequalities above, we have:

   $$\big|H\big| \leq \frac{|HasLeaf(u_1)|}{2} + \frac{|HasLeaf(u_2)|}{2}$$

   Since u_1, u_2 are the children of u, then

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

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

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

2. If $FREQ_g(u) \leq .5$ then either $FREQ_g(u_1) \leq .5$ or $FREQ_g(u_2) \leq .5$
   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 $FREQ_g(u) > .5$. Therefore, $FREQ_g(u_1) > .5$ *or* $FREQ_g(u_2) > .5$ (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.

Run Time

The main challenge is storing 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.

Correctness and Proof: Global Optimum

Correctness

Let $\hat{\lambda} := \textit{Algorithm}\ 2(T, G, \Omega, \lambda)$. 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 rationale as in the above proof with $DUP_g(u), Dup(u)$ instead of $FREQ_g(u), Gene(u)$, we will achieve same result.

Run Time

This algorithm is twice as slow as the Local Algorithm. 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.

References

Barroga, C. F., Zhang, H., Wajih, N., Bouyer, J. H., and Hermodson, M. A. (1996). 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.

Doerrler, W. T. and Raetz, C. R. (2005). Loss of outer membrane proteins without inhibition of lipid export in an escherichia coli yaet mutant. *Journal of Biological Chemistry*, **280**(30), 27679–27687.

Gentle, I., Gabriel, K., Beech, P., Waller, R., and Lithgow, T. (2004). The omp85 family of proteins is essential for outer membrane biogenesis in mitochondria and bacteria. *The Journal of cell biology*, **164**(1), 19–24.

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

Liang, R. and Liu, J. (2008). In-frame deletion of escherichia coli essential genes in complex regulon. *BioTechniques*, **44**(2), 209–10.

Maki, H. and Kornberg, A. (1985). 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.

Mauzy, C. A. and Hermodson, M. A. (1992). 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.

Mohan, S., Kelly, T. M., Eveland, S. S., Raetz, C., and Anderson, M. S. (1994). 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.

Oh, H., Park, Y., and Park, C. (1999). A mutated ptsg, the glucose transporter, allows uptake ofd-ribose. *Journal of Biological Chemistry*, **274**(20), 14006–14011.

Park, Y. and Park, C. (1999). Topology of rbsc, a membrane component of the ribose transporter, belonging to the arah superfamily. *Journal of bacteriology*, **181**(3), 1039–1042.

Schmid, M., Kapur, N., Isaacson, D., Lindroos, P., and Sharpe, C. (1989). Genetic analysis of temperature-sensitive lethal mutants of salmonella typhimurium. *Genetics*, **123**(4), 625–633.

Shimada, T., Kori, A., and Ishihama, A. (2013). Involvement of the ribose operon repressor rbsr in regulation of purine nucleotide synthesis in escherichia coli. *FEMS microbiology letters*, **344**(2), 159–165.

Werner, J. and Misra, R. (2005). Yaet (omp85) affects the assembly of lipid-dependent and lipid-independent outer membrane proteins of escherichia coli. *Molecular microbiology*, **57**(5), 1450–1459.

Zaitseva, J., Zhang, H., Binnie, R. A., and Hermodson, M. (1996). 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.