**Supplementary Material**

Start Link and StartLink+: Prediction of Gene Starts in Prokaryotic Genomes

Karl Gemayel, Alexandre Lomsadze and Mark Borodovsky

1 Note 1: Accuracy of combined predictors.

In pattern recognition, adding up an independent and informed method of analysis and prediction leads to significant increase in accuracy. This principle is often applied in order to corroborate an evidence for an inference from data with incomplete information. The task of gene-start prediction is exactly one where such a principle could be applied.

Supplementary Fig. S1: The probability that an identical selection made by two algorithms $A_1$ and $A_2$ is incorrect, as a function of the number of candidates for the selection. The plots are shown for the three dependency conditions between $A_1$ and $A_2$, see text, describing conditions 1,2,3. Here the error rate Err (Ai) = 0.1, except for a random case (1) where the error is the fraction of false candidates.

Suppose we have two algorithms, $A_1$ and $A_2$, whose gene-start predictions $x_1$ and $x_2$, respectively, have average error rates Err($A_1$), Err($A_2$). For a gene with candidate starts $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ and a true start $s \in \mathcal{C}$, we ask the following question: If algorithms $A_1$ and $A_2$ predict the same start $y$ (i.e. $x_1 = x_2 = y$), what is the probability that $y$ indeed is the true start (i.e. $y = s$)? To compute

$$p(y = s|x_1 = y, x_2 = y)$$

we have:
Let consider the cases when the algorithms $A_1$ and $A_2$ predict the same start

1. completely independently
2. with complete dependence (e.g., being the same algorithm),
3. by selecting a start at random among several candidates with equal probability

Then we have for (1) the following. For independence ($P_I$), and full dependence ($P_D$), randomness ($P_R$) conditions (see the proof below) we get

\[
P_I = \frac{1}{1 + (|C| - 1) \frac{\text{Err}(A_2) \text{Err}(A_1)}{\text{Acc}(A_2) \text{Acc}(A_1)}}
\]
\[
P_D = \frac{1}{1 + (|C| - 1) \frac{\text{Err}(A_2)}{\text{Acc}(A_2)}}
\]
\[
P_R = \frac{1}{|C|}
\]

Since $p(x_i = y | y \neq s) = 1 - p(x_i = y | y = s)$, the probability of the identical start prediction being false increases as the number of options to select from increases (Fig. S1). Importantly, we see the effect on the overall error rate when we add a second, independent prediction. This simple study demonstrates that the largest improvement is gained by maximizing independence between the algorithms.

Note that the real error rates of gene-start prediction are difficult to model theoretically, therefore the above consideration makes simplifying assumptions (e.g., on the uniformity of selecting any of the incorrect starts when a wrong prediction is made).

**Proof of Equation 3**

The derivations of Eq 2 under the three conditions of randomness, independence, and full dependence, is as follows. The first expansion (irrespective of condition) is
\[
p(y = s | x_1 = y, x_2 = y) = \frac{p(x_1 = y, x_2 = y, y = s)}{p(x_1 = y, x_2 = y)} = \frac{p(x_1 = y, x_2 = y, y = s)}{p(x_1 = y, x_2 = y, y = s) + p(x_1 = y, x_2 = y, y \neq s)} \]
\[
= \frac{1}{1 + \frac{p(x_1 = y, x_2 = y, y \neq s)}{p(x_1 = y, x_2 = y, y = s)}}
\]
\[
= \frac{1}{1 + \frac{p(y \neq s) \cdot p(x_2 = y | y \neq s) \cdot p(x_1 = y | x_2 = y, y \neq s)}{p(y = s) \cdot p(x_2 = y | y = s) \cdot p(x_1 = y | x_2 = y, y = s)}}
\]
\[
= \frac{1}{1 + \frac{p(y \neq s) \cdot \text{Err}(A_2) \cdot p(x_1 = y | x_2 = y, y \neq s)}{p(y = s) \cdot \text{Acc}(A_2) \cdot p(x_1 = y | x_2 = y, y = s)}}
\]

Note that assuming uniform priors, we have

\[
P(y = s) = \frac{1}{|C|}
\]
\[
P(y \neq s) = \frac{|C| - 1}{|C|}
\]

Specifically, the prior over \( p(y \neq s) \) assumes that when an algorithm makes a wrong prediction, it is equally likely to choose any of the false candidates. This may or may not be true, especially if algorithms tend to choose false starts closer to the true start. This is important especially under conditions where algorithms are somewhat dependent. However, given that we have no knowledge of \( A_1 \) and \( A_2 \), the most straightforward approach is to assume an uninformed prior. Therefore, we get

\[
p(y = s | x_1 = y, x_2 = y) = \frac{1}{1 + (|C| - 1) \cdot \frac{\text{Err}(A_2) \cdot p(x_1 = y | x_2 = y, y \neq s)}{\text{Acc}(A_2) \cdot p(x_1 = y | x_2 = y, y = s)}}
\]

**Case 1:** Algorithms \( A_1 \) and \( A_2 \) are uniform, random selectors. Since \( A_1 \) and \( A_2 \) are completely random, then they are (by definition) independent. This implies that

\[
p(x_1 = y | x_2 = y, y = s) = p(x_1 = y | y = s) = \frac{1}{|C|}
\]
\[
p(x_1 = y | x_2 = y, y \neq s) = p(x_1 = y | y \neq s) = \frac{|C| - 1}{|C|}
\]

Therefore, we have
\[
p(y = s | x_1 = y, x_2 = y) = \frac{1}{|C|}
\]  

(8)

**Case 2:** Algorithms \( A_1 \) and \( A_2 \) are completely independent
Again, since \( A_1 \) and \( A_2 \) are independent, we have

\[
p(x_1 = y | x_2 = y, y = s) = p(x_1 = y | y = s) = \text{Acc}(A_1)
\]

\[
p(x_1 = y | x_2 = y, y \neq s) = p(x_1 = y | y \neq s) = \text{Err}(A_1)
\]

Therefore, we have

\[
p(y = s | x_1 = y, x_2 = y) = \frac{1}{1 + (|C| - 1) \frac{\text{Err}(A_2)\text{Err}(A_1)}{\text{Acc}(A_2)\text{Acc}(A_1)}}
\]  

(10)

As an example, the values for equation 10 for \( C = 2, 3, 4, \) and \( 5 \) are 0.99, 0.98, 0.96, 0.95, 0.94

**Case 3:** Algorithms \( A_1 \) and \( A_2 \) are completely dependent
Here we have

\[
p(x_1 = y | x_2 = y, y = s) = 1
\]

\[
p(x_1 = y | x_2 = y, y \neq s) = 1
\]

Therefore,

\[
p(y = s | x_1 = y, x_2 = y) = \frac{1}{1 + (|C| - 1) \frac{\text{Err}(A_2)}{\text{Acc}(A_2)}}
\]  

(12)

2 Note 2: Effects of the variations in distributions of the Kimura distance on the StartLink performance.

StartLink uses the Kimura distances as a metric to filter out sequences of too close or too distant relatives. We tested several alternative evolutionary distance metrics, including (non-) synonymous substitution rates, amino acid identity, etc. but they either performed equally well or slightly worse at providing StartLink with a good set of orthologous sequences (data not shown). Overall, we found that *those* metrics provide a way to remove sequences of evolutionarily very close/distant relatives, but that
additional filtering (such as analyzing the gaps in the MSA incurred by each query, as described in the main text) might still be needed.

The effect of varying the maximum Kimura threshold from 0.2 to 0.8, while fixing the minimum threshold to 0.1, is shown in Fig. S2. StartLink’s sensitivity is robust with respect to increase of the maximum Kimura distance. On the other hand, the coverage rate fluctuates. This occurs because when more sequences with larger Kimura distance from query are included in the MSA (and thus, more mutations), the gene start becomes more difficult to infer.

**Supplementary Fig. S2:** The effect of changing the maximum Kimura threshold on StartLink’s sensitivity and coverage rates. The minimum Kimura threshold is fixed to 0.1, and \( x \in \{0.2, 0.3, \ldots, 0.8\} \).

Specifically, *N. pharaonis* and *H. salinarum*’s coverage rates suffer when the maximum Kimura threshold is below 0.4. Note, however, that the number of available genomes of Archaea (used as a reference clade) is quite small, especially when compared to numbers in *Enterobacterales* and *Actinobacteria*. This means that, on average, the coverage rates for archaea will be lower than for bacteria.

On the other hand, *M. tuberculosis*’s coverage rate decreases as the maximum threshold increases past 0.4. A direct inspection of the MSAs showed that they appear to have more mutations around the gene-start region than, for example, in *E. coli* for the same range of Kimura distances. It is unclear whether this is due to a sampling bias of genomes in the database or a biological feature. That said, StartLink+’s sensitivity rate remains constant, meaning that for such alignments, it does not make a prediction.

Similarly, Fig. S3 fixes the maximum Kimura threshold to 0.5 and varies the minimum threshold from 0.001 to 0.4. Overall, this has a lower effect on the coverage and sensitivity rates than in the previous
case. We see an initial drop in the case of *E. coli* when the minimum threshold is close to 0.001. In this range, we observed more orthologs with high nucleotide-level identity rate (to the query gene) than in the other three genomes. It is again unclear whether this stems from a genome sampling bias in the database or from some underlying patterns of genome evolution, but the overall reduction in sensitivity is still small.

**Supplementary Fig. S3:** The effect of changing the minimum Kimura threshold on StartLink’s sensitivity and coverage rates. The maximum Kimura threshold is fixed to 0.5, and $x \in \{0.001, 0.1, 0.2, 0.3, 0.4\}$.

It is important to note that for all the above experiments, the sensitivity of StartLink+ remains consistently high and largely unaffected, while the coverage rate mirrors that of StartLink since it is directly dependent on it (data not shown). In other words, our selection of the Kimura distances interval does not need to be fine-tuned beyond what has been done. This is fortunate, given that we do not want to overfit the small size ground-truth data available to us.

Finally, we tested StartLink on small intervals of the Kimura distance, specifically $[0.001, 0.1], [0.1, 0.2], ... [0.7, 0.8]$. This forced StartLink to operate exclusively in regions of evolutionarily very close and very distant orthologs, and allowed us to see a more detailed view of the effect of the Kimura distance on the algorithm. Fig. S4 shows the effect of these ranges on the sensitivity and specificity.
Supplementary Fig. S4: The performance of StartLink on small intervals of the Kimura distance ranges: [0.001,0.1], [0.1,0.2], [0.2,0.3] ... [0.7,0.8]. The x-axis shows the mean Kimura distance; e.g., for range \([a, b]\), the average is \(\frac{b+a}{2}\).

The effect here is more significant, especially at the blocks at the edges of the whole range, i.e. [0.001,0.1] and [0.7,0.8]. The sensitivity is still rather stable, with the biggest impact being in the range [0.001,0.1]. The coverage, however, is changing significantly. First, the coverage behaves differently in \(E. coli\) and \(M. tuberculosis\) compared to the two archaeal species. Specifically, it starts out high in both \(E. coli\) and \(M. tuberculosis\), and then drops down beyond the [0.4,0.5] range. The drop is especially strong in \(E. coli\), where StartLink struggles to make predictions for ranges [0.6, 0.7] and above (although still maintaining a high and stable sensitivity).

On the contrary, the coverage is low for \(H. salinarum\) and (especially) \(N. pharaonis\) for the [0.001, 0.1] and [0.1, 0.2] ranges. It then rises quickly and stabilizes, before dropping a little in the higher ranges. The reasons for this pattern are, in part, due to just over 1,000 genomes of Archaea in the entire clade, compared to over 6,311 and 8,097 genomes for \(Enterobacteriales\) and \(Actinobacteria\), respectively, with much lower nodes in the taxonomy tree. In fact, \(E. coli\) and \(M. tuberculosis\) each have 1,000 + sequenced genomes of the strains within their own species.

Therefore, the observed difference in patterns is likely to only be a result of the undersampling of the microbial species and may disappear once the database is populated more uniformly.
Supplementary Fig. S5: The percentage of gene start differences between PGAP and StartLink+, computed as Diff(PGAP, StartLink+), as a function of the minimum and maximum Kimura distances between a query and its targets. The color-coding key panel shows the correspondence of color and the percentage of gene starts differences. This analysis was done for 443 query genomes.

We observed that deviations of the StartLink+ predictions from the PGAP annotation were in the same range regardless of variations in min and max values of the range of the Kimura between queries and targets Fig. S5.

3 Note 3: Distributions of the MSA scores computed around gene-starts

The StartLink algorithm relies on metrics, such as a measure of sequence conservation, computed in non-coding and coding regions, thus, around candidate starts. These metrics are computed after removing evolutionarily close and distant sequences, as determined by the Kimura distance, from the set of sequences in the initial MSA.

In this section, we analyze the scores computed by StartLink in different sections of the multiple sequence alignment, from 5' downstream. We run StartLink for genes with experimentally verified starts and build multiple sequence alignments. The goal is to show how the two scores defined in Eq 6 and 7 of the main text behave in different regions of the MSA (specifically, upstream of, at, and downstream of the verified start).
First, we emphasize the following: this analysis is done to show how the algorithm behaves. This means that conditions that StartLink uses, such as only extracting LORFs and filtering by Kimura, are inherently part of the analysis. For example, if a query’s verified start is at the LORF 5’ end, then the scores for the non-coding region are not computed, StartLink will not search for a gene-start there.

3.1 Block conservation scores

We start by analyzing the block conservation score $S_{blk}$ defined by Eq 6 of the main text. For each protein MSA, we take blocks (with width 10 aa) in four different regions (Fig. S6): 20 aa upstream of the verified start (Up-far), just upstream of the verified start (Up-close), just downstream of it (Down-close), and 20 aa downstream of the verified start (Down-far).

Supplementary Fig. S6: Blocks selected for the MSA score analysis.

Fig. S7 shows the distribution of scores for each block region. As expected, the upstream (non-coding) regions have a much lower conservation score than the downstream (coding) regions, as mutations happen with higher frequency in non-coding regions than in coding regions. Also, given that sequences were extracted as LORFs, this region of the MSA would have a large number of gaps that reduce overall score. This pattern helps StartLink to skip over such regions. Notice that the rules of the StartLink setup (Kimura filtering, LORF extraction, etc...) facilitate the use a simple metric based on identity conservation to skip over non-coding regions, where conserved blocks are not detected.
Supplementary Material

Supplementary Fig. S7: Distribution of block conservation scores in regions around verified starts.

On the opposite end of the protein MSA, the two blocks downstream of the verified start show relatively high conservation scores. Interestingly, the Down-far block has a higher conservation score (on average) than the Down-close one. This is especially apparent in *M. tuberculosis*. This pattern indicates that non-synonymous mutations just downstream of the gene-start may be more likely to occur than in a randomly selected section deeper in a gene. This may be a reason why computation of evolutionary distance metrics on the entire gene may not select an optimal set of sequences for StartLink’s gene-start search algorithm.

3.2 Identity score for gene start candidates

Here we analyze the effectiveness of the 5’ score (Eq 7 of the main text) for separating true and false starts. Notably, there are two types of false starts.

In an ideal case, the true start will have a high 5’ identity score, due to presence of orthologous genes with the same location of the 5’ end. The upstream (false) candidates tend to have low conservation scores, due to mutations that occur in non-coding region. On the other hand, the (false) candidates downstream of the verified start can show high identity scores, due to conservation across orthologs. Still, the false starts downstream could be discriminated if codons synonymous to GTG and TTG appear in the alignment to betray absence of selection pressure expected for a start codon. Hence, the penalty for the synonymous codons was added in the equation for $S_{5'}$ (see main text).
Fig. S8 shows how the 5’ score separates the annotated (in this case, verified) gene-starts from the false candidates. The gap between the groups is strikingly wide, meaning that the 0.5 threshold is a reasonable, non-overfitting value.

![Fig. S8](image)

**Supplementary Fig. S8:** Distribution of 5’ identity scores for verified starts as well as for upstream and downstream false gene start candidates.

4 Note 4: Distributions of differences in numbers of selected orthologs

The maximum number of allowed targets (currently $N = 50$) is used to limit the number of sequences in MSA. At most $N$ sequences are selected from a large number of BLAST hits and used in the MSA prior to further filtering. The average number of targets per query after a full StartLink run is shown in Fig. 16; it can vary significantly, especially when comparing *Archaea* and *Enterobacterales*.

One reason why *Enterobacterales* has a high average number of targets is the large spread of the Kimura distances compared to other clades (see Fig. 10). On the other side, the spread of Kimura distances within *Archaea* targets has narrow spread, while the average number of targets reaches as low as 10 per query. This is partly due to the small number of genomes in this clade, making it less likely that we find enough sequences within the right Kimura distance range.

Still, we observed on the verified set that the larger final number of targets per query did not translate into a large improvement in performance. For example, for StartLink running with $N = 50$, both archaea (*H. salinarum* and *N. pharaonis*) end up with on average 20 targets per query, compared to *E. coli*’s 40 targets per query. In terms of error rate, however, for *H. salinarum* and *N. pharaonis* we saw the error rate of 3% and 2% respectively, while *E. coli*’s the error rate was 5%.
In another experiment we decreased $N$ to 20 in order to check the performance on Archaea with low target-per-query averages. This change in $N$ resulted in 10 to 15 targets per query for both archaea. However, we saw no real change in error rate; we saw a slight increase in error rate for *H. salinarum* (by 0.6%) and a decrease in error rate for *N. pharaonis* by 0.7%. Moreover, in all verified genomes, when $N$ was shifted we saw a 0.5% (on average per genome) change in error rate, some positive and some negative. This shows that StartLink is stable with respect to $N$, and the minor shifts could be attributed to the underlying randomness of selecting the target sequences.

**5 Note 5: Gene-start prediction in metagenomes**

Standard self-training *ab initio* predictors usually require a sufficiently long sequence (e.g. a sequenced genome) for estimating the models parameters. In metagenomics applications short read assembled sequences normally are lacking full genomic context. Therefore, conventional gene finders such as GeneMarkS-2 cannot run its full cycle with training of the typical model (Lomsadze et al. 2018) due to insufficient training data. Still, GeneMarkS-2 can run a single iteration which is equivalent to running the gene finder with the MetaGeneMark models (Zhu, Lomsadze, and Borodovsky 2010). Such a run relies on pre-built collection of heuristic GC specific models that do not include models for the regulatory motifs in the gene upstream regions.

Fig. S9 shows results of an experiment made with a goal to show what GeneMarkS-2 accuracy in gene start finding would be on short genome fragments. Since StartLink only requires the gene itself, it’s accuracy is stable irrespective of the length of the genomic fragment (as long as the gene itself is contained within the sequenced fragment).

**Supplementary Fig. S9:** Gene start prediction sensitivity of GeneMarkS-2 on short fragments of five genomes are shown by solid lines. The sensitivity of MetaGeneMark and StartLink, are shown by the dotted and dashed lines, respectively, for each genome.
As shown in Fig. S9, the performance of GeneMarkS-2 starts to degrade rapidly when the input fragment size goes below the 200K range. And at 10K, we see a drop all the way down to 75% on *M. tuberculosis*, *R. denitrificans*, and *E. coli*, 96% on *H. salinarum* and 88% on *N. pharaonis*. This is in contrast to StartLink, which doesn’t depend on the genome size at all, and maintains an average accuracy of 95% on all genomes, with the lowest being for *M. tuberculosis* at 93.62%.

This experiment shows the advantage of similarity-based approaches in ability to make predictions of gene starts in short sequence fragments for genes whose orthologs are present in the database.

6 Note 6: StartLink technical details and optimizations

In this section, we discuss some under-the-hood details of the selection process of target sequences in StartLink, focusing on items that can affect the overall performance and runtime. Before we begin, we state the goal of this selection process by describing the desired properties of the final selected set of sequences. We then describe the selection process and discuss optimizations and shortcuts made to reduce the overall runtime.

In StartLink, target sequence selection is a multi-step process. We start with a search of protein sequence targets in a database and then filter out unwanted sequences. Given that our initial databases are large, we use DIAMOND BLASTp, which provides a faster version of the BLASTp search algorithm. This search typically returns thousands of hits per query, and the goal is to select $N$ (in our case, $N = 50$) target sequences (per query) that will be used by StartLink.

Ultimately, we want to choose a set of sequences that, when aligned, would allow us to find the start of the query gene easily. For our purposes, we want this set $S$ (a query and (maximum) $N$ targets) to satisfy the following pairwise constraints:

$$d_{k}(s_1, s_2) \in [0.1, 0.5] \quad \forall s_1, s_2 \in S, s_1 \neq s_2$$

(13)

This means that no pair of sequences in this set should have a Kimura distance smaller than 0.1 or larger than 0.5.

The most straightforward approach to accomplishing this is to compute all pairwise Kimura distances between query and target sequences, and then randomly select $N$ that exist in the desired range. However, the Kimura distance calculation requires a global alignment of two sequences, which in this case would be very costly (given the large initial number of sequences).

6.1 Avoiding global alignment

Therefore, the first order of approximation is to use the already-existing local alignments between a query and its targets (generated by the BLASTp search), and immediately compute approximate Kimura distance values. In our experiments, we found no tangible difference in StartLink’s performance on the set of genes with verified starts based on whether global or local alignment was used in the Kimura distance computation.

Note, however, that these local alignments are only provided between a query and a target, and not between pairs of target sequences. Furthermore, there can still be tens of thousands of targets per query and going through them all would result in many unnecessary computations.
6.2 Avoiding all-pairs target alignments

We can avoid (at this stage) directly comparing target sequences to each other by observing how they compare to the query. This step exploits the fact that two very similar target sequences should have similar Kimura distances to the query. As such, if the Kimura distance between two targets $t_1$ and $t_2$ and the query is roughly the same (i.e. $d_k(q,t_1) \approx d_k(q,t_2)$), then we remove one of them from the list of candidates under the assumption that there is a higher chance that they are very similar to each other. We deem two Kimura distances as approximately the same if they are equal up to the fourth decimal place.

It’s important to note that this heuristic is not optimal, however. In particular, it doesn’t hold strictly in the opposite direction, i.e., in general, two sequences with similar Kimura distances to the query may have differences that did not affect the result of computation of the Kimura distance to the query. That said, we have found that this step slightly improves performance because we are less likely at this stage to import multiple target sequences that are very similar to each other, which would end up in biasing the alignment.

6.3 Avoiding analyzing the large number of target sequences

We first note that hits retrieved from BLASTp are ordered by $e$-value. In practical terms, this means that they are ordered in terms of a similarity score to the query sequence. It also means that they are roughly ordered by the Kimura distance; we say "roughly" because the ordering is not exact, but it does not affect the final performance.

Given this list, we want to find the position of the sequence after which no sequence exists with $d_k \in [a, b]$. If the list was perfectly sorted, this task would boil down to finding the first sequence with $d_k > b$. Given that the ordering is not exact, however, we take some measures to decrease the likelihood that our heuristic removes sequences within the range $[a, b]$.

First, if the total number of target sequences for a query is less than 2,000, we skip this filtering step since the number of computations needed is not large. Second, instead of looking for the first sequence with $d_k > b$, we place an additional buffer that favors keeping in more out-of-range sequences than removing in-range sequences; basically, we update the threshold such that we search for $d_k > b + 0.2$. Third, instead of traversing the list in order and finding the first sequence that satisfies $d_k > b + 0.2$, we perform a binary search through the list, which makes it much less likely to stop at a very low positioned sequence in the list.

As such, if the number of sequences is higher than 2,000, we perform a binary search through the list to find an (approximate) "upper bound" above which all sequences (likely) have a distance value larger than the maximum allowed threshold $b$ (+0.2). This operation is quick, requiring $\log(n)$ Kimura distance computations on average (where $n$ is the number of hits). In other words, for a list of 100,000 hits, we require (on average) 16.6 Kimura distance computations, where each computation is a simple run through an existing local alignment, at the nucleotide level, counting the transition and transversion changes.

We effectively now have a list of sequences whose Kimura distance is in the range $[0, b + 0.2]$. We don’t perform the same operation to find the lower limit because our desired threshold of 0.1 is close to 0, leaving us a small buffer region. Instead, we shuffle the list and go through it one sequence at a
time until we have 50 sequences that satisfy our requirements. In the worst case, we would have to go through all sequences in this range, especially if the target sequences are very similar to the query or each other.

6.4 Optimality in the face of randomness

If we had gone through every sequence in the list, computed $d_k$, sorted it, then filtered out the sequences outside the range $[a, b]$, we would still have to randomly select only $N$ (from the possibly thousands) of the remaining sequences. This reduction, which comes from the requirement of the subsequent multiple sequence alignment step, means that even if we were to act optimally at the initial stages, the act of random selection reduces the likelihood that our final set is actually optimal. In other words, we can act sub-optimally (as suggested in the above heuristics) at the initial steps without necessarily observing a change in the outcome. In practice, we did not see a tangible difference in StartLink’s final performance by acting optimally or sub-optimally at the initial stages.

7 Note 7: Examples of multiple sequence alignments made by StartLink

We show below examples of MSAs where StartLink+ selection of the gene start (#selected) doesn’t match PGAP’s prediction (#ref). The examples were taken from eight genomes, coming from the four clades considered in the main text.

| #selected | M---------------------------------------------------------------|
|-----------|-----------------------------------------------------------------|
| 42864:45542;−;A | Ltsrhtpltqlryrvtnvclll!mislaVcaqsgktiagrvVdetsqpv |
| 1;0.3867  | LsidriysfQyrLiscyvlll-lslavyASAqskerfVgrvdtetvqpv |
| 2;0.387   | LsidriysfQyrLiscyvlll-lslavyASAqskerfVgrvdtetvqpv |
| 3;0.3897  | LsidriysfQyrLiscyvlll-lslavyASAqskerfVgrvdtetvqpv |
| 4;0.3925  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 5;0.3995  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 6;0.4026  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 7;0.4038  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 8;0.405   | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 9;0.4069  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 10;0.4185 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 11;0.4201 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 12;0.4205 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 13;0.4213 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 14;0.4226 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 15;0.423  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 16;0.4241 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 17;0.4259 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 18;0.429  | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 19;0.4292 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |
| 20;0.4296 | LnisshysfQyrLiscyvlll-lsltvfaqskerfSgrVldtetsqpv |

_Bacteroides reticulotermitis, FCB group_
### Supplementary Material

**Haloferax sp., Archaea**

| #selected       | M---------------------------------------------------------------- |
|-----------------|---------------------------------------------------------------|
| #q-3prime       | ---------------------------------------------------------------|
| #ref            | M---------------------------------------------------------------|

`215415;215993;--A MrtrvfaatViallaqaaaflagypfedttta--agdsqgqfptqqtta`

| 1;0.1456       | MrtrlvatViallttagisaafltgypfedttta--agdsqgqfptqqtta |
|----------------|---------------------------------------------------------------|
| 2;0.146        | MrtrlvatViallttagisaafltgypfsddsa--agdsqgqfptqqtta |
| 3;0.164        | MrtrpllaatViallttagysacfltgypfdttta--agdsqgqfptqqtta |
| 4;0.1727       | MrtrlvatViallttagisaafltgypfsddsa--agdsqgqfptqqtta |
| 5;0.1751       | MrtrlvatViallttagisaafltgypfsddsa--agdsqgqfptqqtta |
| 6;0.1773       | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 7;0.1783       | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 8;0.1803       | MrtrlvatViallttagisaafltgypfsddsa--agdsqgqfptqqtta |
| 9;0.1862       | MrtrlvatViallttagisaafltgypfsddsa--agdsqgqfptqqtta |
| 10;0.2057      | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 11;0.227       | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 12;0.2291      | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 13;0.2293      | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 14;0.2316      | MrtrlvatViallttagysaafVtgypfsddsa--agdsqgqfptqqtta |
| 15;0.3906      | Mrtrlvsavvlavaaqgqsvfltgypfadqsdqsdqsdqsdqsfqptqqtta |
| 16;0.3976      | Mrtrlsvvvlavaldqgqfvdsvfltgypfsddsa--elgdsqgqfptqqtta |
| 17;0.4102      | Mrtrlsvvvlavldqgqfvdsvfltgypfsddsa--elgdsqgqfptqqtta |
| 18;0.4142      | Mrtrlsvvvlavldqgqfvdsvfltgypfsddsa--elgdsqgqfptqqtta |
| 19;0.4291      | Mrtrlsvvvlavldqgqfvdsvfltgypfsddsa--elgdsqgqfptqqtta |
| 20;0.4331      | Mrtrlsvvvlavldqgqfvdsvfltgypfsddsa--elgdsqgqfptqqtta |
Tatumella saanichensis, Enterobacterales

Leucobacter chironomi, Actinobacteria

Butyricimonas sp., FCB group
**Abstract**

This study focuses on the comparison between two bacterial species: Halorubrum kocurii and Providencia alcalifaciens. The primary protein sequences of these species are presented below.

### Halorubrum kocurii, Archaea

| Selected | Q3prime |
|----------|---------|
| -M-      |         |

| Ref. | Sequence |
|------|----------|
| 20902;21885;+-B | DMSresq1kndteisaeaqdarsqvlqiqlgdklkstpmlllplwvlqyagq |
| 1;0.124 | VDEsresq1kndteisaeaqdarsqslguigdklkstpmlllplwvlqyagq |
| 2;0.1547 | VDEsresq1kndteisaeaqdarsqvlriqdtecltpmlllplwvlqyagq |
| 3;0.1756 | Massgsqrlgdgileiideaagdarsqvisindteletpmlllplwvlqyagq |
| 4;0.1777 | Massgsqrlgdgileiideaagdarsqvisindteletpmlllplwvlqyagq |
| 5;0.1927 | Massgsqrlgdgileiideaagdarsqvlriddteletpmlllplwvlqyagq |
| 6;0.1964 | Mthsgsrliddgileitseagdarsqvirfddteletpmlllplwvlqyagq |
| 7;0.4616 | Msfpertgkvaevevdstvedaraqtllrikdeletpmlllplwvlqyagq |
| 8;0.4712 | Mfssprtqdiatfdiestaagdaaqtrrtdtsltqmpmlllplwvlqyagq |

### Providencia alcalifaciens, Enterobacterales

| Selected | Q3prime |
|----------|---------|
| -M-      |         |

| Ref. | Sequence |
|------|----------|
| 386604;389732;+A | Mnpsltafsritsgrtransilv1plllvllati1lfwa |
| 1;0.1949 | Mnpsltafsritsgrtransilv1plllvllati1lfwa |
| 2;0.1971 | Mnpsltafsritsgrtransilv1plllvllati1lfwa |
| 3;0.4132 | Mnpsltafsritsgrtransilv1plllvllati1lfwa |
| 4;0.4284 | Loidensamuapstafsritsgrtransilv1plllvllati1lfwa |
8 Supplementary Figures (cont)

Supplementary Fig. S10: The differences between PGAP annotation and StartLink+ predictions. Comparisons were made separately for predictions made at each step of StartLink (A, B, C).
Supplementary Fig. S11: The intergenic distance conservation for $f=0$ and $f=3$ as a function of the most frequent upstream distance per component. The data was computed in experiments with 443 query genomes. The PGAP annotation was used for the positions of gene starts. The origin in X axis is situated at -9.

Supplementary Fig. S12: The distribution of $(\min K(q), \max K(q))$ vectors defined for queries and their homologs. Data on more than 1,000,000 query genes from 443 genomes (Table 1).
Supplementary Fig. S13: The cumulative percentage of queries with at most N BLASTp hits. The shaded bands show the standard deviations (per clade) across query genomes.

Supplementary Fig. S14: The cumulative distribution of BLASTp hits (<40) per query in a genome, shown for the four clades. This is a zoomed in version of Supplementary Figure S13.
9 Supplementary Tables

**Supplementary Table 1:** Number of genomes selected in each prokaryotic clade, and the number of genes with gene starts predicted by StartLink+.

| Clade          | Query | StartLink | StartLink+ |
|----------------|-------|-----------|------------|
| Actinobacteria | 111   | 276,468   | 228,837    |
| Archaea        | 108   | 243,424   | 212,162    |
| Enterobacterales | 119  | 451,255   | 394,271    |
| FCB group      | 105   | 253,257   | 229,212    |
| **Total**      | 443   | 1,224,404 | 1,064,482  |

**Supplementary Table 2:** Percentage of errors in gene start predictions made by the three tools on the commonly predicted set of genes with experimentally verified starts

|          | % of incorrect start predictions |
|----------|----------------------------------|
|          | # of commonly predicted genes in the verified set | StartLink | GeneMarkS2 | Prodigal |
| E. coli  | 762                             | 3.81       | 3.02       | 2.36     |
| H. salinarum | 484                         | 3.10       | 1.65       | 2.89     |
| M. tuberculosis | 547                     | 7.68       | 8.23       | 9.87     |
| N. pharaonis    | 284                        | 1.41       | 1.06       | 1.41     |
| R. denitrificans | 475                     | 5.68       | 2.53       | 4.42     |
| **All %**     | 2,552                        | 4.58       | 3.57       | 4.35     |
| **Average %** | 2,552                       | 4.34       | 3.30       | 4.19     |

**Data and software access**

For additional data supported the findings made in the paper as well as for the software download for academic use see the GitHub page: [https://github.com/gatech-genemark/StartLink-exp](https://github.com/gatech-genemark/StartLink-exp)