# Supplemental Material

## Best reciprocal FASTA

Each CDS (a) from the genome (A) was searched, with FASTA (Pearson 1990), against the CDSs of the other genome (B). If the top hit covered at least 80% of the length of both sequences with at least 30% identity, a reciprocal FASTA search of the top hit sequence (b) was launched against the CDSs of the first genome. If the reciprocal top hit is the same as the original query CDS then (a) and (b) are considered orthologous genes of (A) and (B). In a second step, in order to validate the results, we performed a BLASTN (Altschul et al. 1997) comparison between the strains, visualized using ACT (Carver et al. 2005) to curate ambiguous cases e.g. gene remnants (pseudogenes), IS elements, phage-related CDSs and to check for a syntenic relationship among the putative orthologs.

## Alignments

Whole genome sequence alignments were made using the MAUVE algorithm (Darling et al. 2004). The complete chromosome sequence of the eleven *Salmonella* strains and the four outgroups were aligned. For the *Staphylococcus* dataset, only the thirteen *Staphylococcus* chromosomes were aligned, excluding the four outgroup sequences due to the overall low sequence similarity to the *Staphylococcus* genomes. For the *Streptococcus* dataset the overall low sequence similarity between the different strains did not allow the construction of whole genome sequence alignments. Moreover, for each genus, aminoacid sequence alignments of the core gene (i.e. orthologous genes shared by all the strains of a given genus and the corresponding outgroups) products were also built using CLUSTALW (Thompson et al. 1994); the alignments were manually curated and gapped columns were removed.

## Structural Annotation

*Integrase(-like) protein domains*

Each query genome (six frame translation) was searched against fifteen integrase(-like) Pfam (Sonnhammer et al. 1998) Hidden Markov Models (HMMs), using the HMMER software (http://hmmer.janelia.org/). Throughout this analysis, fifteen protein domains (Supplemental Table 2) that are frequently found in proteins involved in the mobilization of DNA are referred to as integrase-like domains, or simply "integrase".

*Phage-related protein domains*

In order to predict CDSs of putative phage origin, we used the *hmmpfam* search option of the HMMER package and each query genome (six frame translation) was searched against a manually constructed database of 191 phage-related Pfam HMMs (Supplemental Table 2).

*Non-coding RNA*

Each query genome was searched against the non-coding RNA families of Rfam (Griffiths-Jones et al. 2005). This methodology was followed in order that putative associations of GIs with other non-coding RNA families (apart from the tRNA and tmRNA genes) could be captured.

*Compositional analysis*

For all the 668 regions identified in this analysis, their Interpolated Variable Order Motif (IVOM) score (Vernikos and Parkhill 2006) was calculated, using the *alien_hunter* algorithm (http://www.sanger.ac.uk/Software/analysis/alien_hunter/). The IVOM frequency is a weighted sum of compositional biases derived from different order $k$-mers ($0 \leq k \leq 8$) that captures both low and high order compositional deviation from the backbone composition. The IVOM score is expressed as the relative entropy between the query and the genome-backbone (variable order) compositional distribution, i.e. the higher the IVOM score is, the stronger the compositional deviation.

*Repeat analysis*

Repeat analysis at the boundaries of each of the 668 regions was performed, using the REPuter software (Kurtz and Schleiermacher 1999). The REPuter parameters used are as follows: Type of repeats (= *Forward, Complemented*), minimum size of repeats (= *18bp*), number (hamming distance) of mismatches for degenerate repeats (= *3*).

*Other*

All 668 regions were further annotated in terms of size (bp), gene density (number of genes per kb) and their insertion point; in the latter case two distinct states were (binary) evaluated: insertion point within a CDS locus (disrupting the corresponding CDS) or insertion within an intergenic part of the chromosome.

## Machine Learning

Given a set of $N$ examples (training set) along with their corresponding class (i.e. GI, non-GI) we are trying to build a model of how the input vectors $\{\mathrm{x}_i\}_{i=1}^{N}$ affect the corresponding classification $\{c_i\}_{i=1}^{N}$, with the aim of making predictions of the class for unseen input data, based on the model parameters (weights[1]) $\{w_j\}_{j=1}^{K}$ calculated during the training; $K$ denotes the number of basis functions (in our case structural features e.g. repeats, RNA, IVOM, etc) used to describe the data. In order to build structural GI models, we use Generalized Linear Models (GLMs), a form of model suitable for classification and regression analysis. A GI structural model ($S_i$) is the weighted sum of $K$ basis functions of the form:

$$S_i = U + \sum_{j=1}^{K} w_j \cdot x_{ij} \quad (1)$$

[1] Throughout this manuscript, we will refer to the RVM model parameters *w* as “weights” because they quantify the relative contribution of each feature to the model, i.e. the higher the feature weight the higher its contribution to the model. Note that for the model parameters *w* there is a no actual upper or lower bound.

For two-class classification (in our case class 1 corresponds to GI and class 0 to non-GI) the aim is to predict the posterior probability that a given input x is a true GI, given the model. Because we are interested in a binary classification task, we apply to the output of model $S_i$ the logistic function:

$$\sigma(S_i) = \frac{1}{1 + e^{-S_i}} \quad (2)$$

The logistic function (2) normalizes ($0 \leq \sigma(S_i) \leq 1$) the output of model $S_i$ and can be considered as the probability that a given structure is a true GI, given the model. In function (1), $U$ is a constant that controls the output of this function, in such a way that the final score (assuming the logistic function) can take any value between 0 and 1.

The feature weight $w$ is indicative of the actual feature contribution to the given model, (i.e. the higher the weight the higher the feature contribution), however it does not take into account the dispersion of the actual values of a given feature in the training set. A more reliable estimate of the actual feature importance can be calculated through the following function:

$$R_j = w_j \cdot SD_j \quad (3)$$

where $R_j$ is the "importance" of feature $j$ with weight $w_j$ and standard deviation $SD_j$ (the standard deviation of the actual values of a given basis function in the training set). Under this framework, a basis function with significant $SD_j$ will be more important (higher $R$) than a basis function with comparable weight but with lower $SD_j$.

Details about the training and technical aspects of the RVM are discussed in detail in (Tipping 2001). Briefly, the probability that a given dataset is correctly classified given the model is given by the following function:

$$P(c \mid x, w) = \prod_{i=1}^{N} \sigma(S_i)^{ci} (1 - \sigma(S_i))^{1-ci} \quad (4)$$

Note that for binary classification $c \in \{0,1\}$. During the training process, the RVM is estimating appropriate values of the model weights in an iterative fashion, with the aim of maximizing the likelihood function (4). If a given basis function is informative when classifying the training dataset, then by setting its weight to a non-zero value, will increase the number of correctly classified data, which in turn will increase the likelihood function (4), and therefore the probability of the model given the training set. On the other hand, if a basis function in not informative (or has redundant information) for the classification task, there is no actual weight value that would increase the likelihood; however, setting the corresponding weight value to zero, maximizes the posterior probability of the given model. From this point onwards, the given basis function is treated as non informative and is removed ($w_j = 0$) from the model. At the end of the training process only few informative basis functions ('relevance' vectors) with non-zero weights will contribute to the final model. Under this increased sparsity framework, RVM models avoid efficiently overfitting to the training dataset, selecting only a small number of 'relevance' vectors, with good generalization properties on unseen datasets.

## ROC curve

In order to evaluate the performance of the RVM classifier under different GI models, we implemented a receiver operating characteristic (ROC) curve analysis (Supplemental Fig. S4). The ROC curve illustrates graphically the performance of a classifier, under different cut-off values showing the trade-off between sensitivity and specificity. More specifically, in a ROC curve the True Positive rate (Sensitivity) is plotted against the False Positive rate (1-Specificity) for increasing values of the score cut-off of a binary classifier. The area under the (ROC) curve (AUC) is a measure of accuracy: The closer the curve follows the left-hand and the top border of the ROC space, the more accurate the classification model. A perfect classifier (AUC=1) would predict correctly all the True Positives (Sensitivity = 1) giving no False Positives (Specificity = 1). A classifier that makes a random guess would result in an AUC of 0.5.

## Cross-Validation

Cross-validation is a method for estimating generalization error based on resampling. It provides an indication of how well the classifier performs in making new predictions for previously unseen data. Some of the data is removed prior to the training; after the training, the data that was removed is used to test the performance of the learned model on unseen data. In this analysis we pursued a *5*fold cross validation approach dividing each dataset into five subsets; the results of this *5*fold cross validation are shown in Supplemental Figure S4 and summarized in Fig. 3.

## References

Altschul, S.F., T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25:** 3389-3402.

Carver, T.J., K.M. Rutherford, M. Berriman, M.A. Rajandream, B.G. Barrell, and J. Parkhill. 2005. ACT: the Artemis Comparison Tool. *Bioinformatics* **21:** 3422-3423.

Darling, A.C., B. Mau, F.R. Blattner, and N.T. Perna. 2004. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res* **14:** 1394-1403.

Griffiths-Jones, S., S. Moxon, M. Marshall, A. Khanna, S.R. Eddy, and A. Bateman. 2005. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res* **33:** D121-124.

Kurtz, S. and C. Schleiermacher. 1999. REPuter: fast computation of maximal repeats in complete genomes. *Bioinformatics* **15:** 426-427.

Pearson, W.R. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol* **183:** 63-98.

Sonnhammer, E.L., S.R. Eddy, E. Birney, A. Bateman, and R. Durbin. 1998. Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res* **26:** 320-322.

Thompson, J.D., D.G. Higgins, and T.J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* **22:** 4673-4680.

Tipping, M.E. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research* **1:** 211-244.

Vernikos, G.S. and J. Parkhill. 2006. Interpolated variable order motifs for identification of horizontally acquired DNA: revisiting the Salmonella pathogenicity islands. *Bioinformatics* **22:** 2196-2203.