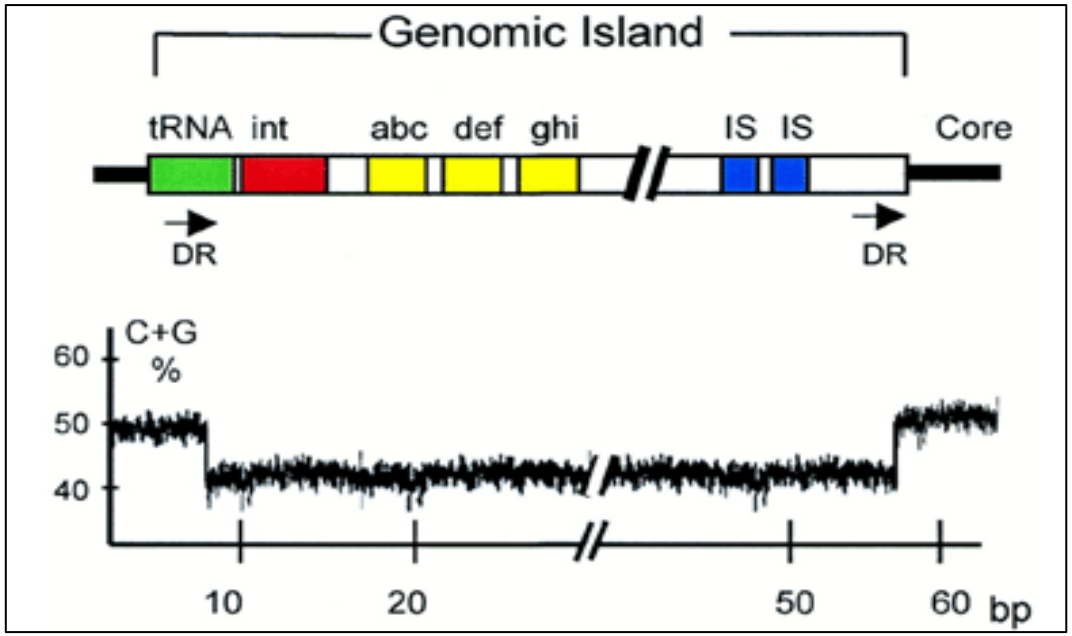
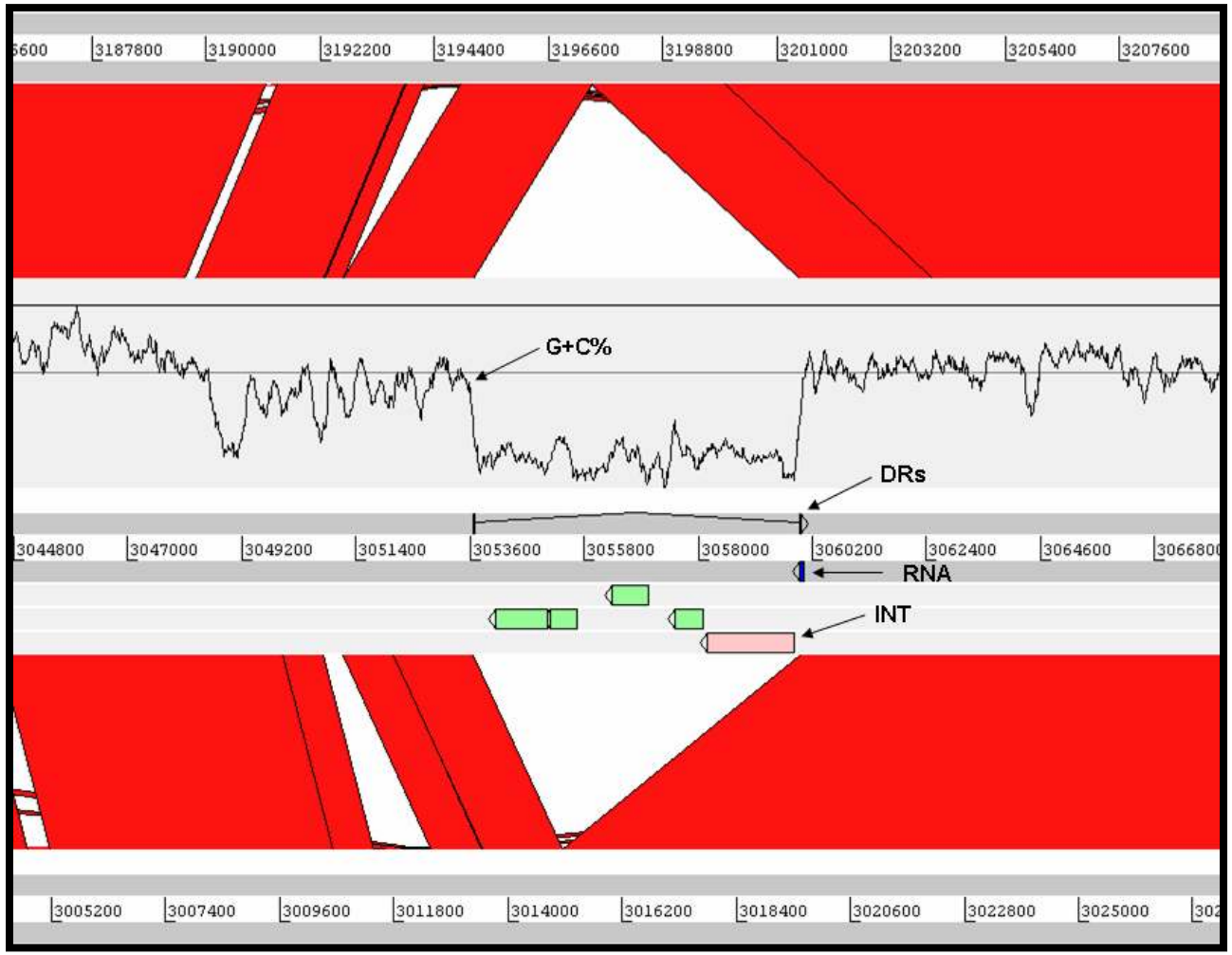
Resolving the Structural Features of Genomic Islands

Genomic Island Structure



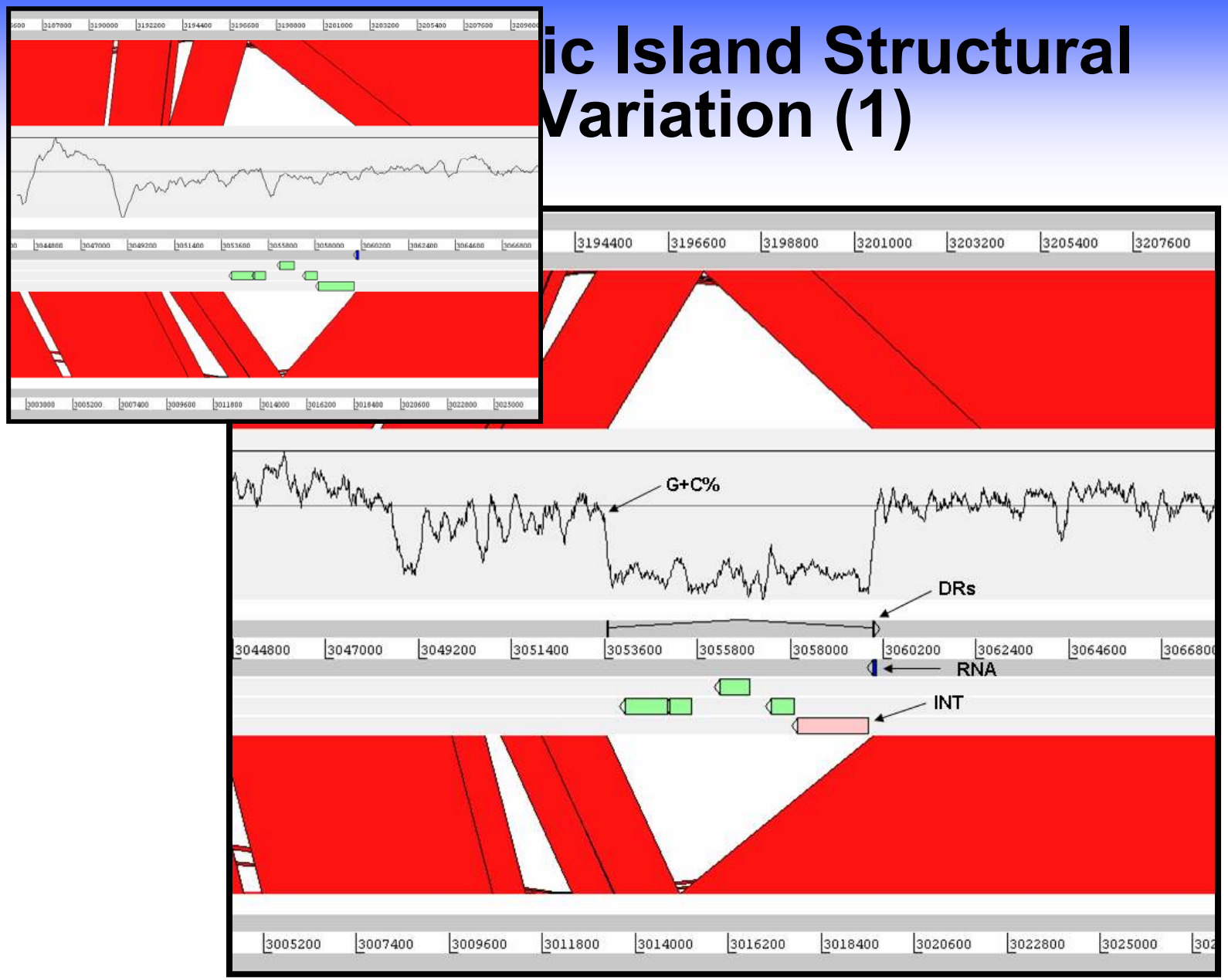
- Large inserts of horizontally acquired DNA (10 to 200kb)
- Sequence composition different from the core backbone composition
- Insertion usually adjacent to RNA genes
- Often flanked by direct repeats or insertion sequence (IS) elements
- Limited phylogenetic distribution i.e. present in some genomes but absent from closely related ones
- Often mosaic structures of several individual acquisitions
- Genetic instability
- Presence of mobility genes (e.g. integrase, transposase)

Genomic Island Structural Variation (1)



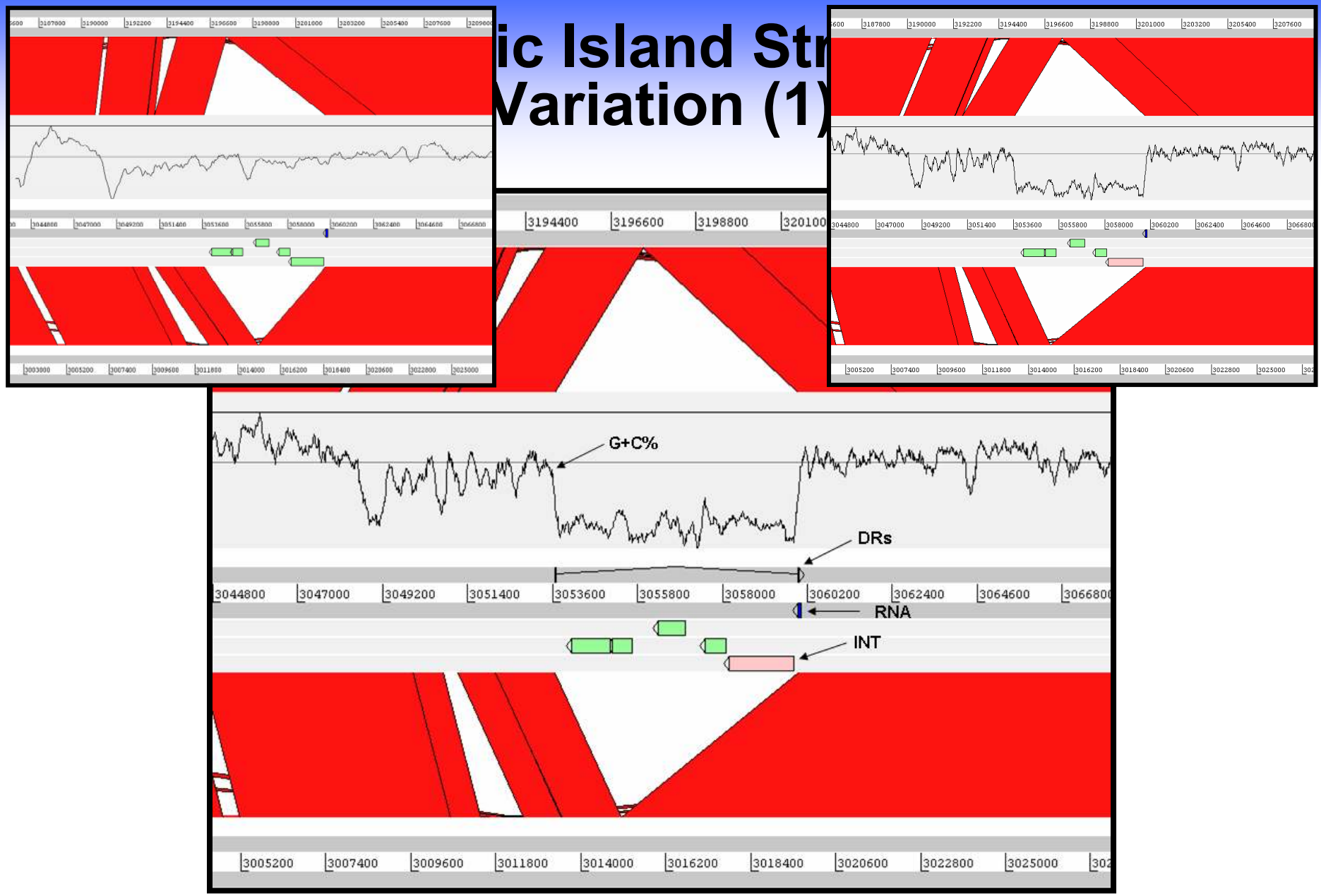
Hacker J et al., Mol Microbiol 1997

ic Island Structural Variation (1)



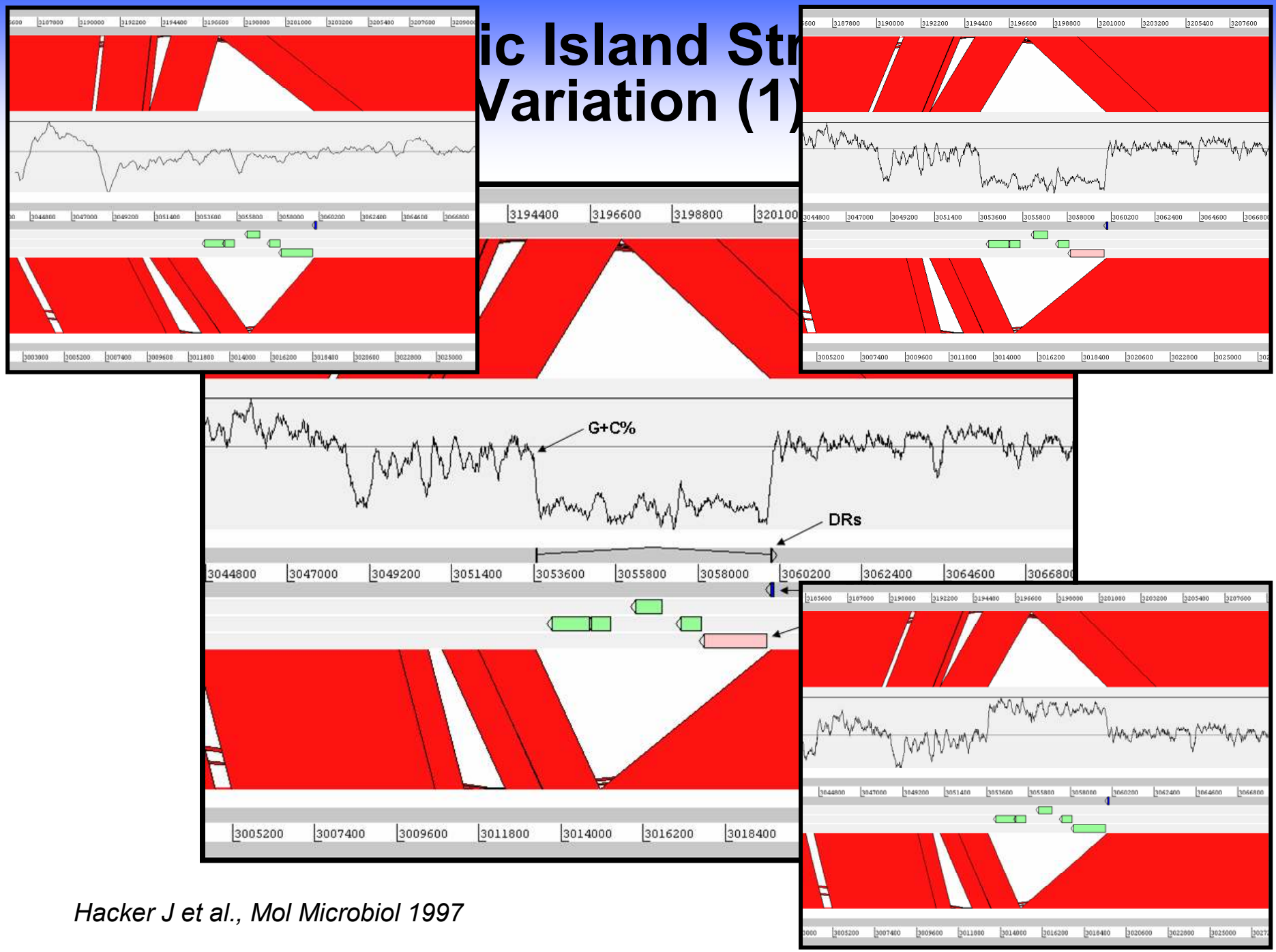
Hacker J et al., Mol Microbiol 1997

ic Island Str
Variation (1)



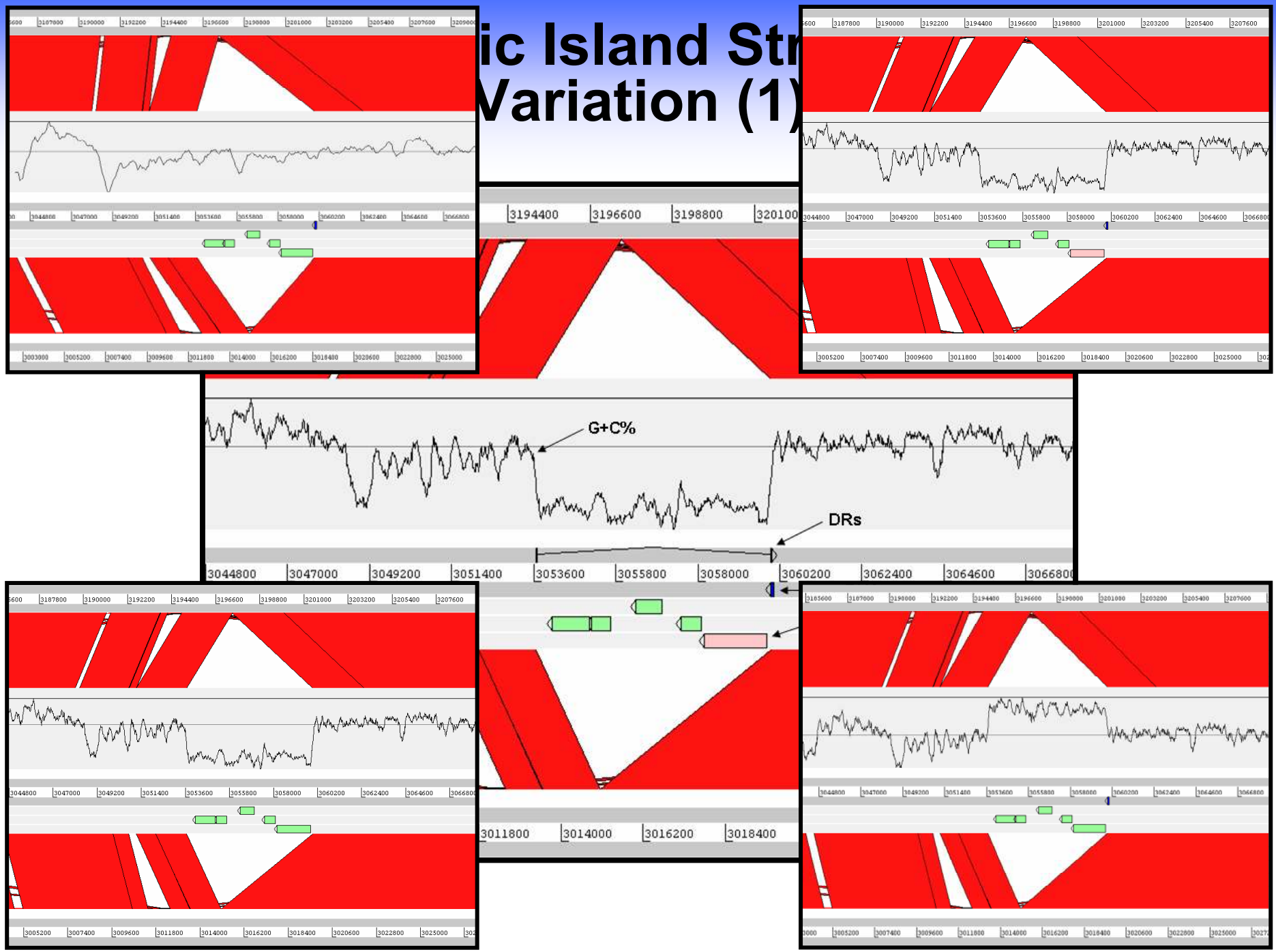
Hacker J et al., Mol Microbiol 1997

ic Island Str
Variation (1)

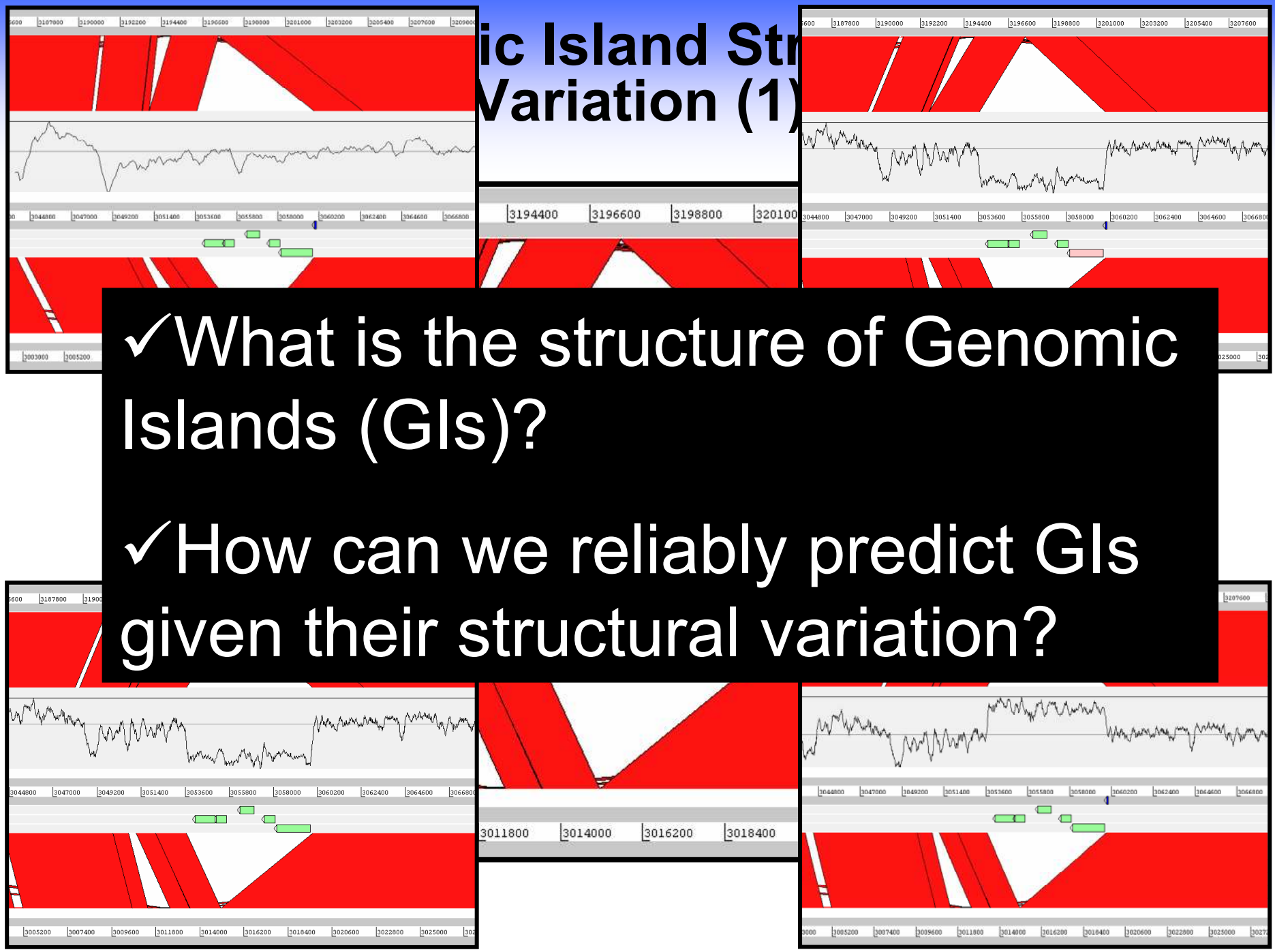


Hacker J et al., Mol Microbiol 1997

ic Island Str
Variation (1)



ic Island Str
Variation (1)



- ✓What is the structure of Genomic Islands (GIs)?
- ✓How can we reliably predict GIs given their structural variation?

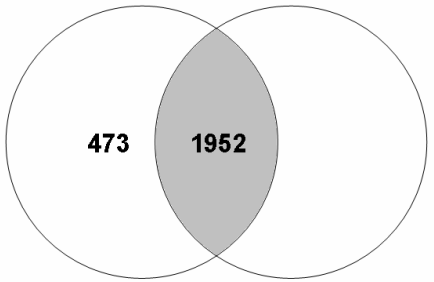
Genomic Island Structural Variation (2)

Coordinates	Host	GI	Size	G+C% deviation	Repeats	Integrase	RNA	Gram
839352..853808	*S. aureus* MW2	vSa3	14457	-4.49	1	1	1	+
1891660..1923796	*S. aureus* MW2	vSaß	32137	-4.24	0	0	1	+
1932974..1959426	*S. aureus* Mu50	vSaß	26453	-4.16	0	1	1	+
2133112..2148791	*S. aureus* Mu50	vSa4	15680	-2.56	1	1	0	+
2251120..2266138	*S. epidermidis* RP62A	vSe1	15019	-1.43	1	0	0	+
1519667..1558081	*S. epidermidis* ATCC15305	vSe2	38415	-6.4	1	1	1	+
1012154..1023023	*S. haemolyticus* JCSC1435	vSh1	10870	-2.87	1	1	0	+
2117669..2133994	*S. haemolyticus* JCSC1435	vSh2	16326	-4.06	1	1	1	+
2578642..2593348	*S. haemolyticus* JCSC1435	vSh3	14707	-1.74	0	1	0	+
385739..432833	*S. agalactiae* NEM316	PAI3	47095	1.64	1	0	0	+
711791..759003	*S. agalactiae* NEM316	PAI7	47213	1.62	1	0	0	+
1013026..1060093	*S. agalactiae* NEM316	PAI8	47068	1.66	0	0	0	+
1163554..1197443	*S. agalactiae* NEM316	PAI10	33890	2.04	0	0	1	+
1255736..126127	S. agalactiae NEM316	PAI11	5544	-6.37	1	1	1	+
302172..361067	*S. typhi* CT18	SPI-6	58896	-0.57	0	0	1	-
605515..609992	*S. typhi* CT18	SPI-16	4478	-9.98	1	1	1	-
1085156..1092735	*S. typhi* CT18	SPI-5	7580	-8.52	0	1	1	-
1625084..1664823	*S. typhi* CT18	SPI-2	39740	-4.91	0	0	1	-
2460780..2465939	*S. typhi* CT18	SPI-17	5122	-13.39	0	0	1	-
2742876..2759156	*S. typhi* CT18	SPI-9	16281	4.62	0	0	1	-
2859262..2899034	*S. typhi* CT18	SPI-1	39773	-6.22	0	0	0	-
3053654..3060017	*S. typhi* CT18	SPI-15	6364	-3.01	1	1	1	-
3132606..3139414	*S. typhi* CT18	SPI-8	6809	-14.03	1	1	1	-
3883111..3900458	*S. typhi* CT18	SPI-3	17348	-5	0	0	1	-
4321943..4346614	*S. typhi* CT18	SPI-4	24672	-7.74	0	0	0	-
4409511..4543072	*S. typhi* CT18	SPI-7	133562	-2.42	1	1	1	-
4683690..4716539	*S. typhi* CT18	SPI-10	32850	-5.51	0	1	1	-

Training Dataset Core Genes

Salmonella

S.bongori 12419
S.arizonae RSK2980
S.typhi CT18
S.typhi TY2
S.paratyphi A SARB42
S.paratyphi A AKU12601
S.typhimurium SL1344
S.typhimurium DT104
S.typhimurium LT2
S.enteritidis PT4
S.gallinarum 287/91

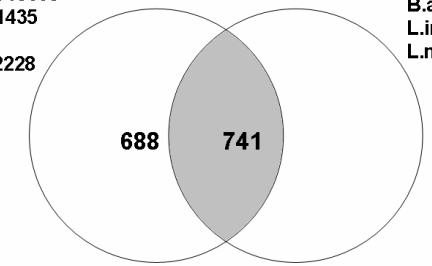


E.coli MG1655
S.flexneri 2a 301
E.coli CFT073
E.coli EDL933

•15 Genomes: 1952 core genes
•641,284 amino acid sites
•Number of constant sites: 553997 (86.4%)

Staphylococcus

S.saprophyticus ATCC15305
S.heamolyticus JCSC1435
S.epidermidis RP62A
S.epidermidis ATCC12228
S.aureus MRSA252
S.aureus RF122
S.aureus Mu50
S.aureus N315
S.aureus MSSA476
S.aureus MW2
S.aureus USA300
S.aureus COL
S.aureus NCTC8325

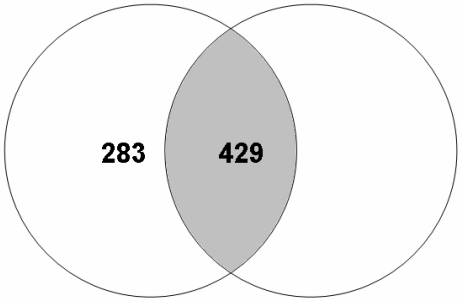


B.subtilis 168
B.anthracis Ames 0581
L.innocua CLIP 11262
L.monocytogenes EDGe

•17 Genomes: 741 core genes
•233,994 amino acid sites
•Number of constant sites: 91571 (39.1%)

Streptococcus

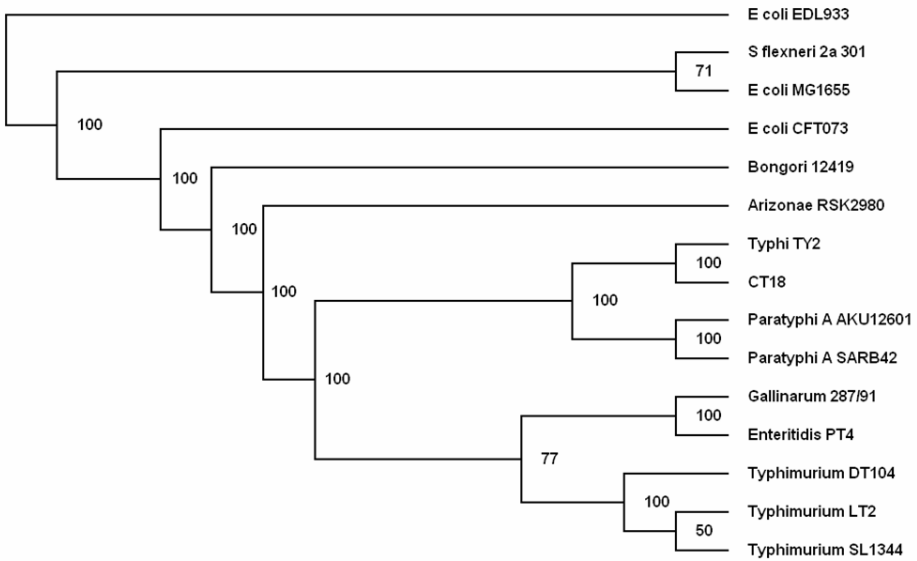
S.pneumoniae R6
S.pneumoniaeTIGR
S.suis P1/7
S.thermophilus CNRZ1066
S.thermophilus LGM18311
S.agalactiae NEM316
S.agalactiae A909
S.uberis 0140J
S.equi 4047
S.pyogenes MGAS10750
S.pyogenes MGAS2096
S.pyogenes MGAS9429
S.pyogenes MANFREDO



L.johnsonii NCC533
L.plantarum
WCFS1
E.faecalis V583
L.lactis IL1403

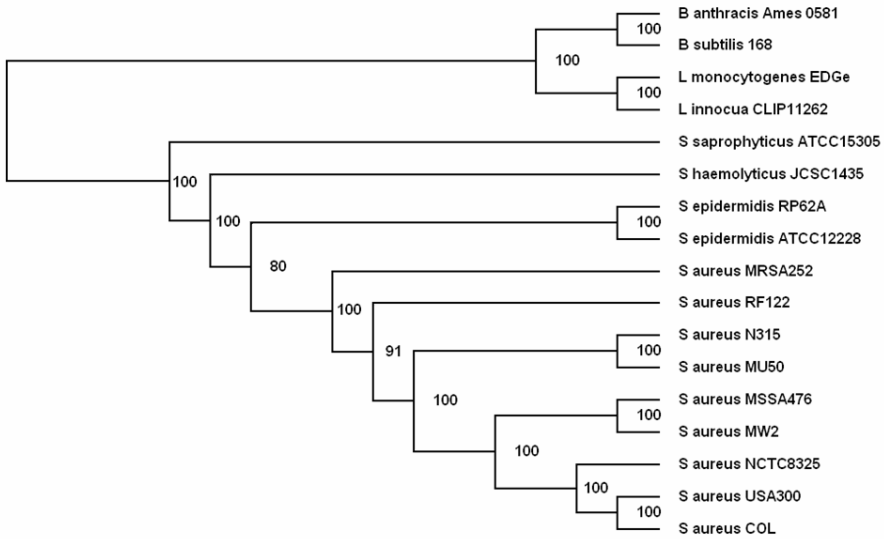
•17 Genomes: 429 core genes
•138,193 amino acid sites
•Number of constant sites: 47595 (34.4%)

Training Dataset (Phylogeny)



1952 core genes: NJ, 100x replicates, Kimura

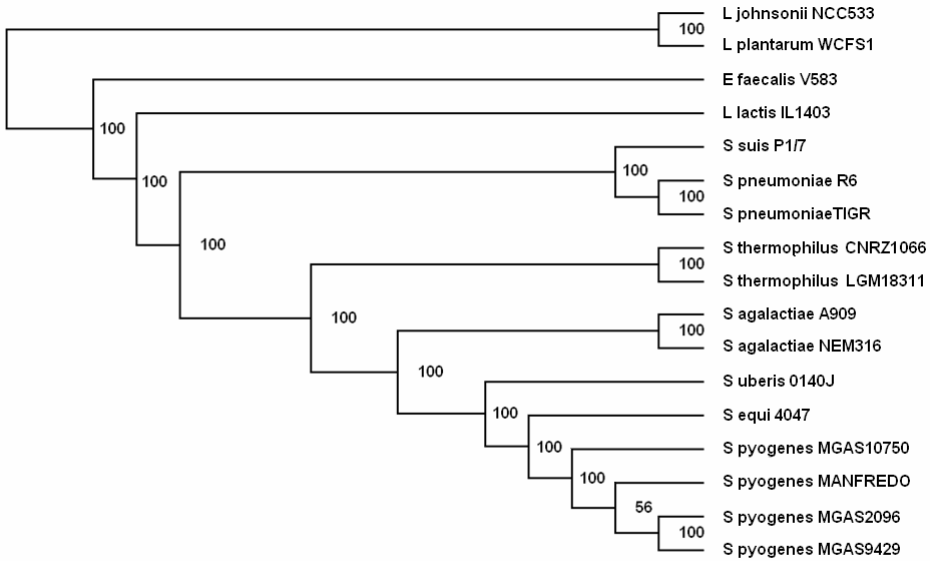
11 Salmonella Strains + 4 outgroups



741 core genes: NJ, 100x replicates, Kimura

13 Staphylococcus Strains + 4 outgroups

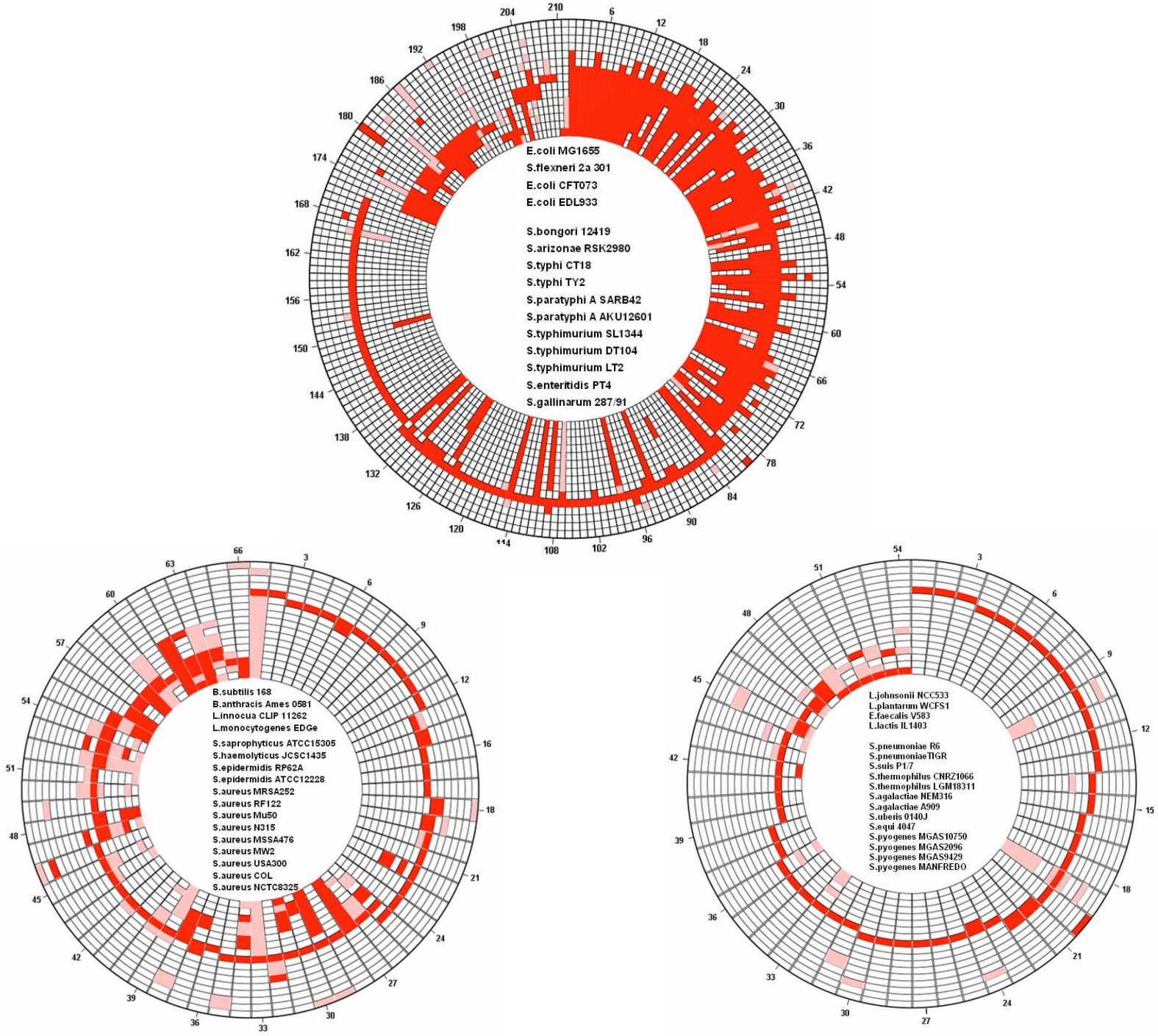
Datasets	Positive examples	Negative examples	Total
Salmonella	211	210	**421**
Streptococcus	54	53	**107**
Staphylococcus	66	74	**140**
Gram -	211	210	421
Gram +	120	127	266
Gram +/-	331	337	**668**



13 Streptococcus Strains + 4 outgroups

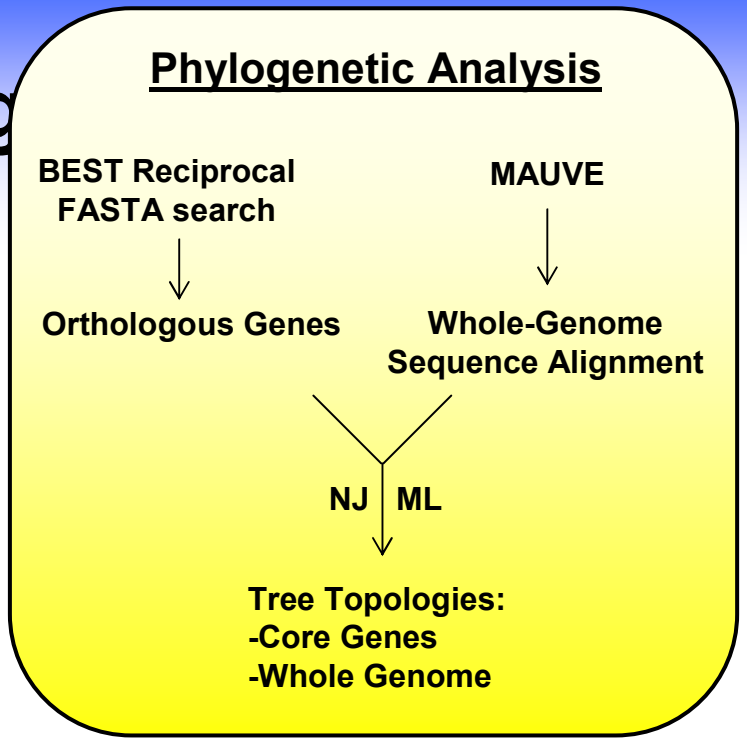
429 core genes: NJ, 100x replicates, Kimura

Training Dataset (Positive Control)

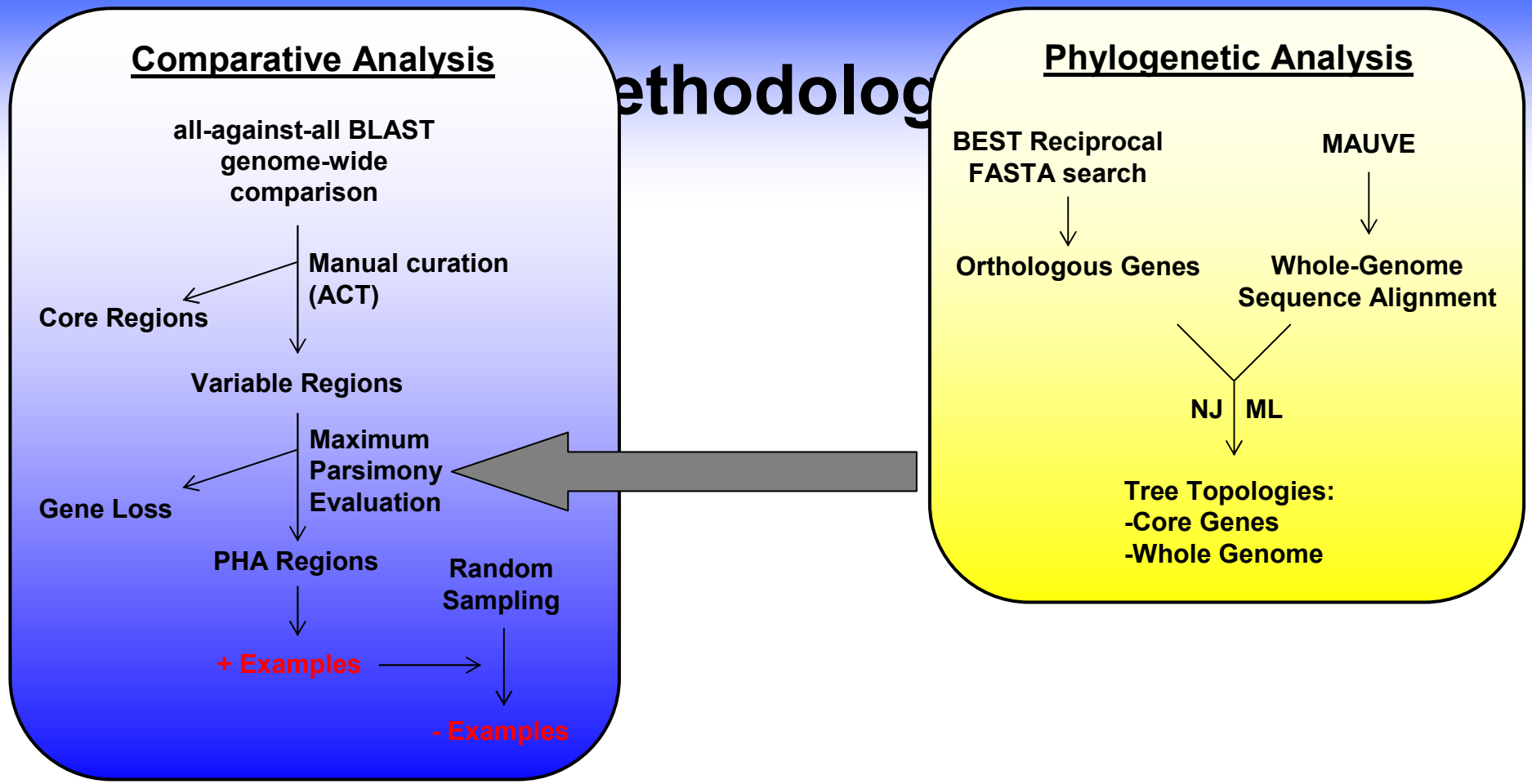


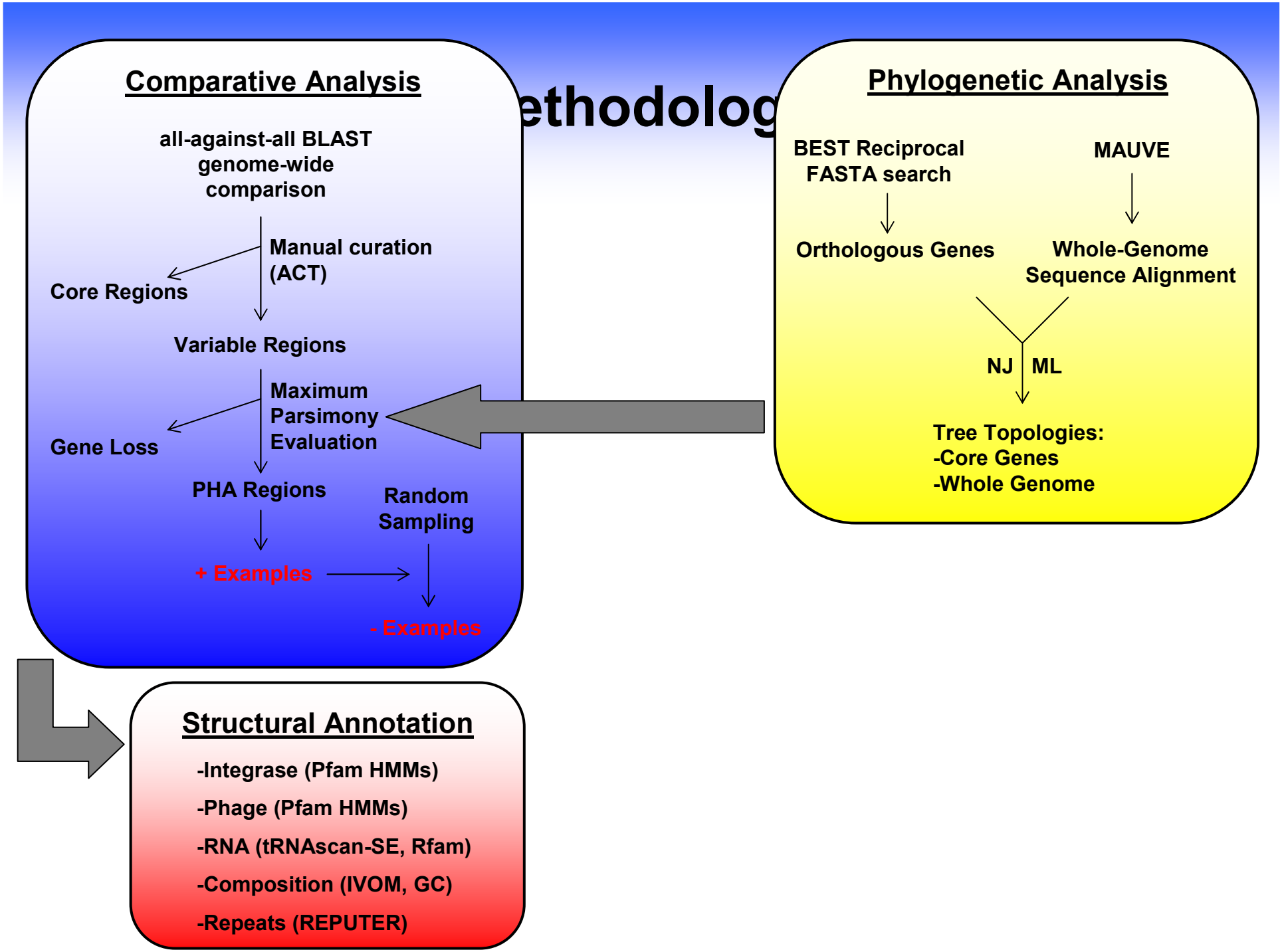
Methodology

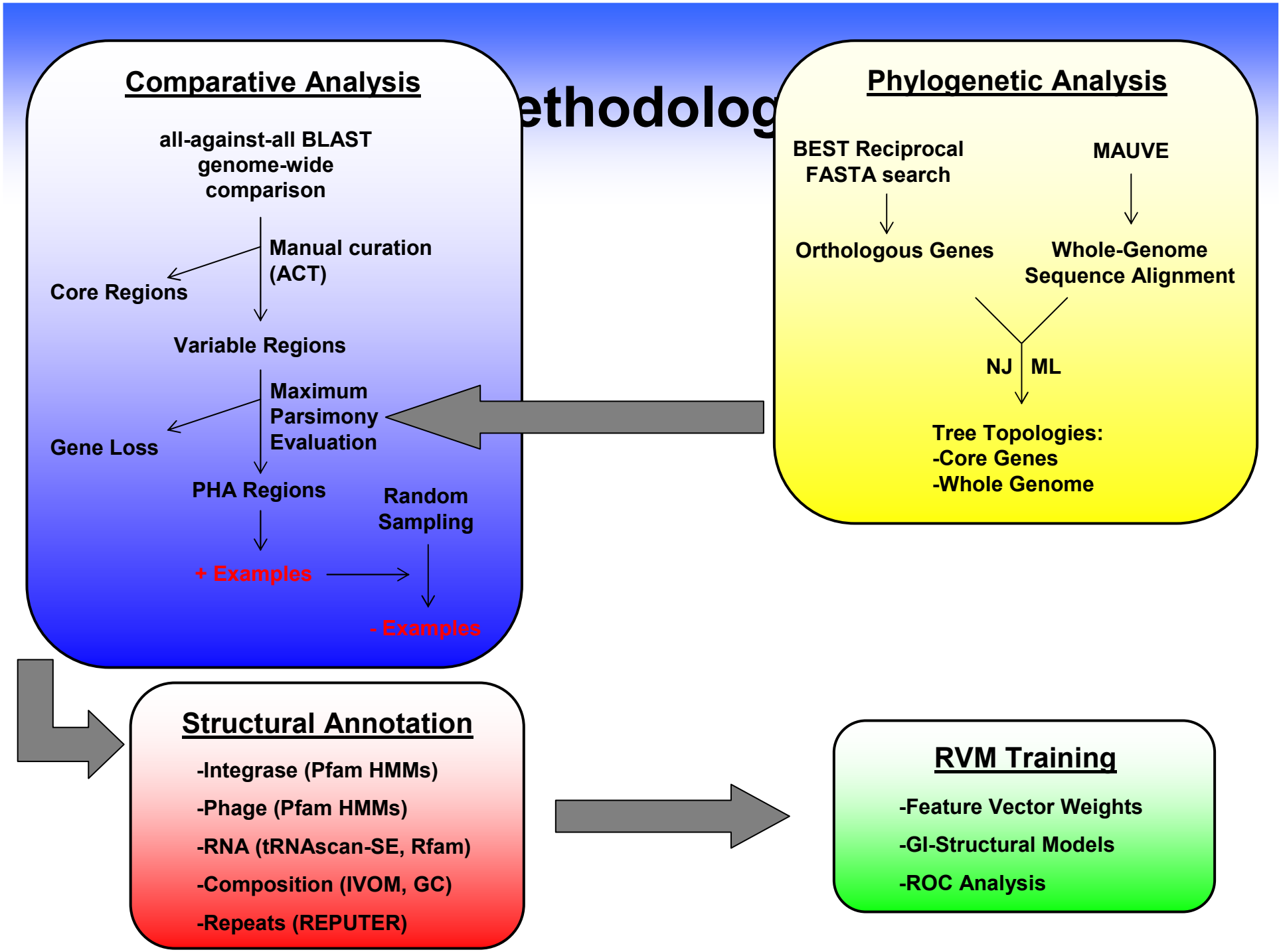
Methodolog


Phylogenetic Analysis
BEST Reciprocal FASTA search
MAUVE
Orthologous Genes
Whole-Genome Sequence Alignment
NJ
ML
Tree Topologies:
-Core Genes
-Whole Genome

ethodolog


Comparative Analysis
all-against-all BLAST genome-wide comparison
Manual curation (ACT)
Core Regions
Variable Regions
Maximum Parsimony Evaluation
Gene Loss
PHA Regions
Random Sampling
+ Examples
- Examples
Phylogenetic Analysis
BEST Reciprocal FASTA search
MAUVE
Orthologous Genes
Whole-Genome Sequence Alignment
NJ
ML
Tree Topologies:
-Core Genes
-Whole Genome


ethodolog
Comparative Analysis
all-against-all BLAST
genome-wide
comparison
Manual curation
(ACT)
Core Regions
Variable Regions
Maximum
Parsimony
Evaluation
Gene Loss
PHA Regions
Random
Sampling
+ Examples
- Examples
Phylogenetic Analysis
BEST Reciprocal
FASTA search
MAUVE
Orthologous Genes
Whole-Genome
Sequence Alignment
NJ
ML
Tree Topologies:
-Core Genes
-Whole Genome
Structural Annotation
-Integrase (Pfam HMMs)
-Phage (Pfam HMMs)
-RNA (tRNAscan-SE, Rfam)
-Composition (IVOM, GC)
-Repeats (REPUTER)


ethodolog
Comparative Analysis
all-against-all BLAST genome-wide comparison
Manual curation (ACT)
Core Regions
Variable Regions
Maximum Parsimony Evaluation
Gene Loss
PHA Regions
Random Sampling
+ Examples
- Examples
Phylogenetic Analysis
BEST Reciprocal FASTA search
MAUVE
Orthologous Genes
Whole-Genome Sequence Alignment
NJ
ML
Tree Topologies:
-Core Genes
-Whole Genome
Structural Annotation
-Integrase (Pfam HMMs)
-Phage (Pfam HMMs)
-RNA (tRNAscan-SE, Rfam)
-Composition (IVOM, GC)
-Repeats (REPUTER)
RVM Training
-Feature Vector Weights
-GI-Structural Models
-ROC Analysis

Test dataset

Genomes	HGTs	Non HGTs	Total
Salmonella	211	211	422
Streptococcus	55	55	110
Staphylococcus	78	78	156
Gram -	211	211	422
Gram +	133	133	266
Gram +/-	344	344	688

Test dataset

Genomes	HGTs	Non HGTs	Total
Salmonella	211	211	422
Streptococcus	55	55	110
Staphylococcus	78	78	156
Gram -	211	211	422
Gram +	133	133	266
Gram +/-	344	344	688

Comparative analysis

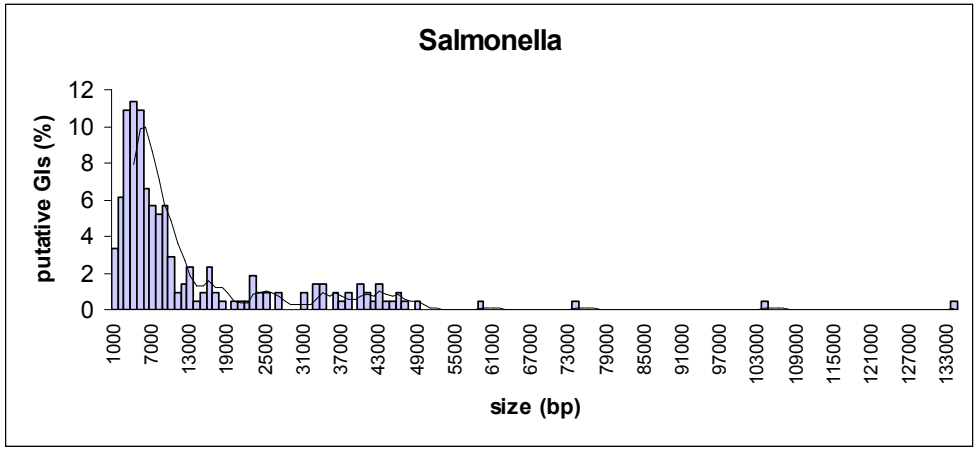
Test dataset

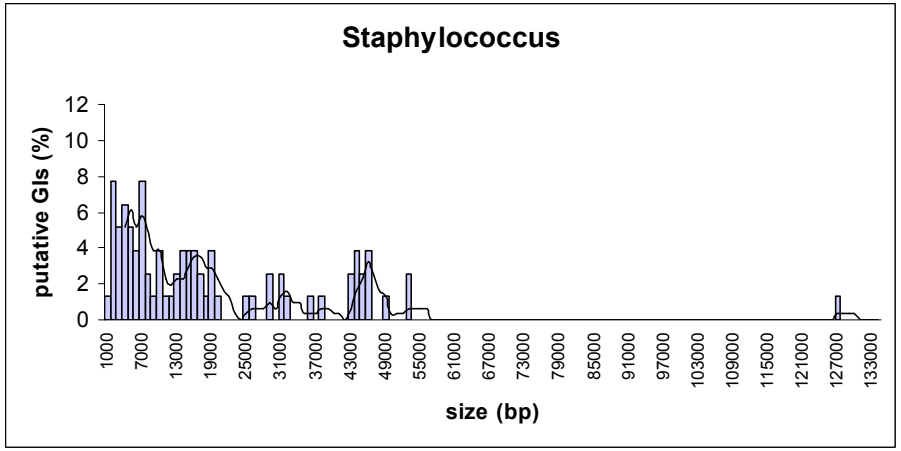
Genomes	HGTs	Non HGTs	Total
Salmonella	211	211	422
Streptococcus	55	55	110
Staphylococcus	78	78	156
Gram -	211	211	422
Gram +	133	133	266
Gram +/-	344	344	688

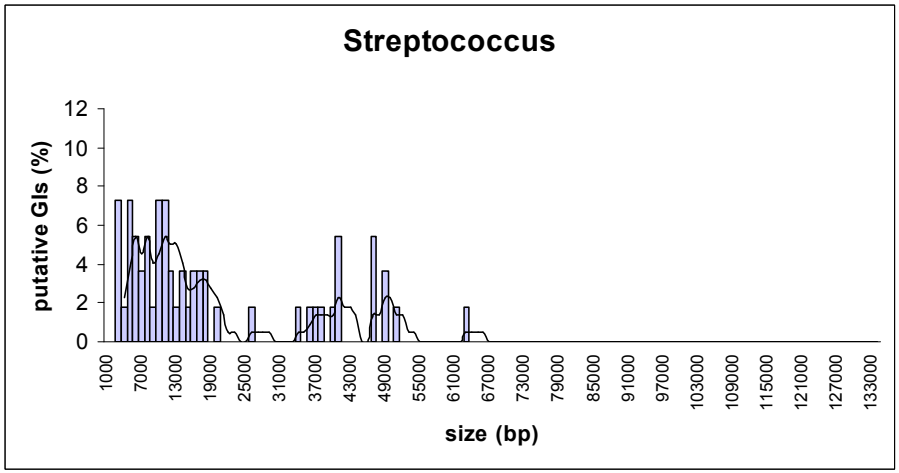
Comparative analysis

Random Sampling following the size distribution of GIs

GI Size Distribution







Generalized Linear Models

Generalized linear models (GLMs) are a commonly used form of model for both classification (separate data into two or more classes) and regression (estimate the value of a continuous function) problems.

Generalized Linear Models

Generalized linear models (GLMs) are a commonly used form of model for both classification (separate data into two or more classes) and regression (estimate the value of a continuous function) problems. GLMs take the form:

$$\eta(x) = \sum_{m=1}^{M} \beta_m \phi_m(x) + K$$

where φ is a set of M basis functions (which can be arbitrary real-valued functions) and β is a vector of weights.

Generalized Linear Models

Generalized linear models (GLMs) are a commonly used form of model for both classification (separate data into two or more classes) and regression (estimate the value of a continuous function) problems. GLMs take the form:

$$\eta(x) = \sum_{m=1}^{M} \beta_m \phi_m(x) + K$$

where φ is a set of M basis functions (which can be arbitrary real-valued functions) and β is a vector of weights.

One way of looking at generalized linear models is that the basis functions define a projection of the data into a high-dimensional space (called feature space) where the data is either linear (for regression problems) or linearly separable (for classification problems). An important step in GLM learning is to find a feature space which allows a linear model to fit the training data, while not being of such high dimensionality that overfitting becomes a problem.

Relevance Vector Machine

The **Relevance Vector Machine** (RVM) is a sparse method for training generalized linear models. This means that it will generally select only a subset (often a small subset) of the provided basis functions to use in the final model.

RVM download:

http://www.miketipping.com/index.php?page=rvm

http://www.vectoranomaly.com/downloads/downloads.htm

Relevance Vector Machine

N examples (training dataset)

$$\{\mathbf{x}_i\}_{i=1}^{N} \qquad (1)$$

Relevance Vector Machine

N examples (training dataset)

$$\{\mathbf{x}_i\}_{i=1}^{N} \qquad (1)$$

$$\{c_i\}_{i=1}^{N} \qquad (2)$$

Relevance Vector Machine

N examples (training dataset)

$$\{\mathbf{x}_i\}_{i=1}^{N} \qquad (1)$$

$$\{c_i\}_{i=1}^{N} \qquad (2)$$

$$\{w_j\}_{j=1}^{K} \qquad (3)$$

Relevance Vector Machine

N examples (training dataset)

$$\{\mathrm{x}_i\}_{i=1}^{N} \qquad (1)$$

$$\{c_i\}_{i=1}^{N} \qquad (2)$$

$$\{w_j\}_{j=1}^{K} \qquad (3)$$

$$S_i = U + \sum_{j=1}^{K} w_j \cdot x_{ij} \qquad (4)$$

Relevance Vector Machine

N examples (training dataset)

$$\{\mathrm{x}_i\}_{i=1}^{N} \qquad (1)$$

$$\{c_i\}_{i=1}^{N} \qquad (2)$$

$$\{w_j\}_{j=1}^{K} \qquad (3)$$

$$S_i = U + \sum_{j=1}^{K} w_j \cdot x_{ij} \qquad (4)$$

$$\sigma(S_i) = \frac{1}{1+e^{-S_i}} \qquad (5)$$

Relevance Vector Machine

N examples (training dataset)

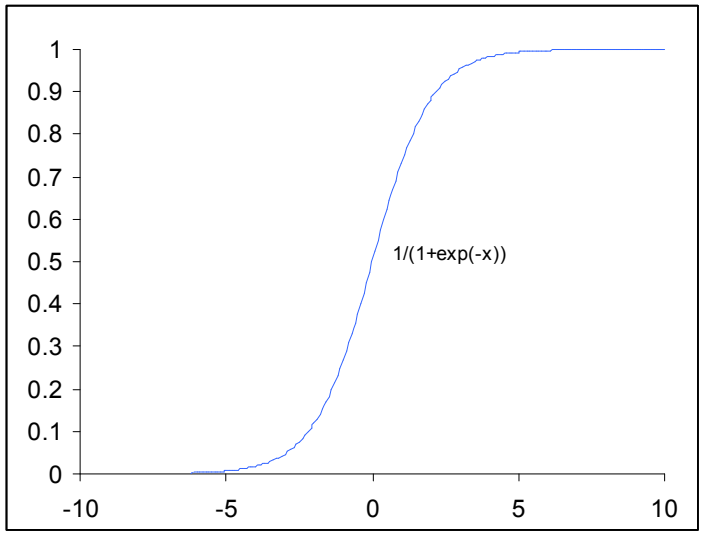
$$\{\mathbf{x}_i\}_{i=1}^{N} \qquad (1)$$

$$\{c_i\}_{i=1}^{N} \qquad (2)$$

$$\{w_j\}_{j=1}^{K} \qquad (3)$$

$$S_i = U + \sum_{j=1}^{K} w_j \cdot x_{ij} \qquad (4)$$

$$\sigma(S_i) = \frac{1}{1+e^{-S_i}} \qquad (5)$$



Relevance Vector Machine

N examples (training dataset)

$$\{\mathrm{x}_i\}_{i=1}^{N} \qquad (1)$$

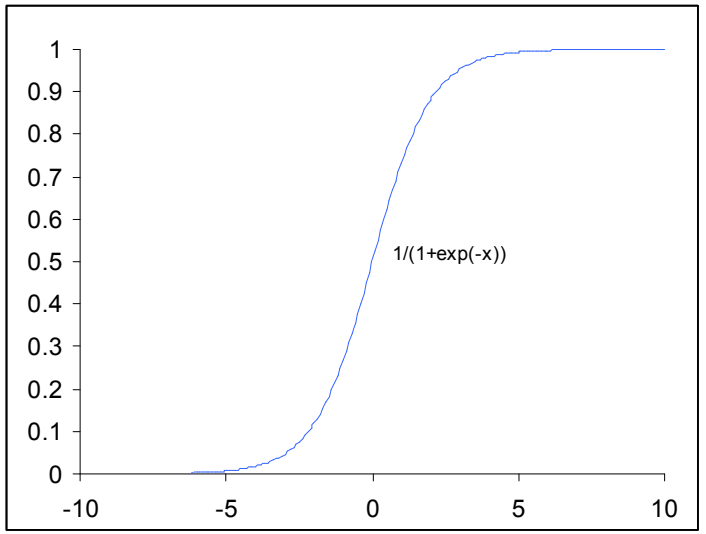
$$\{c_i\}_{i=1}^{N} \qquad (2)$$

$$\{w_j\}_{j=1}^{K} \qquad (3)$$

$$S_i = U + \sum_{j=1}^{K} w_j \cdot x_{ij} \qquad (4)$$

$$\sigma(S_i) = \frac{1}{1+e^{-S_i}} \qquad (5)$$

$$R_j = w_j \cdot SD_j \qquad (6)$$



Relevance Vector Machine

The probability that the data set is correctly labelled given some classifier model:

$$P(c \mid x, w) = \prod_{i=1}^{N} \sigma(S_i)^{c_i} (1 - \sigma(S_i))^{1-c_i} \qquad (7)$$

Relevance Vector Machine

The probability that the data set is correctly labelled given some classifier model:

$$P(c \mid x, w) = \prod_{i=1}^{N} \sigma(S_i)^{c_i} (1 - \sigma(S_i))^{1-c_i} \qquad (7)$$

Assuming that the training data is correctly labeled, Bayes' theorem allows us to turn this expression around and infer likely values of the weights given some labeled data.

$$P(w \mid x, c) \propto P(w) P(c \mid x, w) \qquad (8)$$

Relevance Vector Machine

The probability that the data set is correctly labelled given some classifier model:

$$P(c \mid x, w) = \prod_{i=1}^{N} \sigma(S_i)^{c_i} (1 - \sigma(S_i))^{1-c_i} \qquad (7)$$

Assuming that the training data is correctly labeled, Bayes' theorem allows us to turn this expression around and infer likely values of the weights given some labeled data.

$$P(w \mid x, c) \propto P(w) P(c \mid x, w) \qquad (8)$$

In this expression, we have introduced an extra probability distribution, *P(w)*, which is our prior belief in the values of the weights. If we merely wished to perform classical GLM training, we would have no preference for any particular value of the weights, and would encode this by providing a very broad ("non-informative") prior. However, as discussed previously, we believe that simple models are more likely to make useful generalizations. A preference for simplicity can be encoded using an Automatic Relevance Determination prior. In this case, we introduce an additional vector of parameters, α. Each element of the vector controls the width of the prior over the corresponding weight:

Relevance Vector Machine

The probability that the data set is correctly labelled given some classifier model:

$$P(c \mid x, w) = \prod_{i=1}^{N} \sigma(S_i)^{c_i} (1 - \sigma(S_i))^{1-c_i} \qquad (7)$$

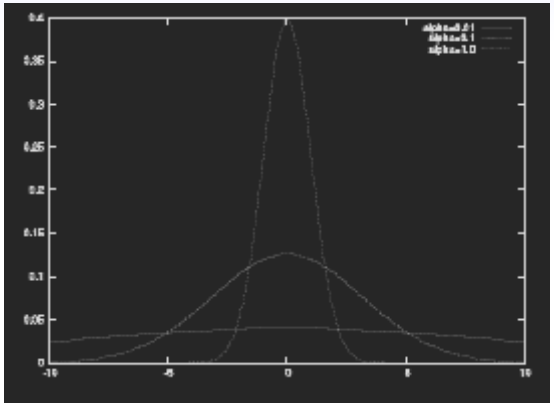
Assuming that the training data is correctly labeled, Bayes' theorem allows us to turn this expression around and infer likely values of the weights given some labeled data.

$$P(w \mid x, c) \propto P(w) P(c \mid x, w) \qquad (8)$$

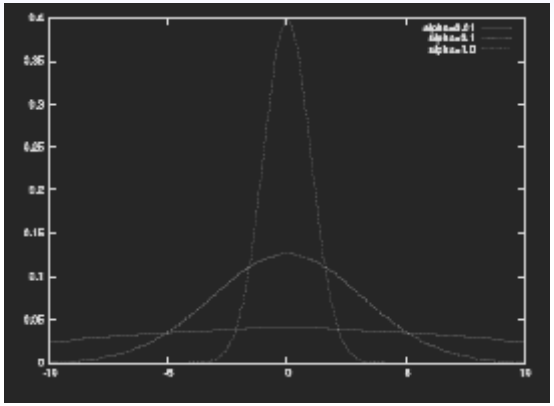
In this expression, we have introduced an extra probability distribution, *P(w)*, which is our prior belief in the values of the weights. If we merely wished to perform classical GLM training, we would have no preference for any particular value of the weights, and would encode this by providing a very broad ("non-informative") prior. However, as discussed previously, we believe that simple models are more likely to make useful generalizations. A preference for simplicity can be encoded using an Automatic Relevance Determination prior. In this case, we introduce an additional vector of parameters, *α*. Each element of the vector controls the width of the prior over the corresponding weight:

$$P(w) = \prod_{i} \mathcal{G}(w_i \mid 0, a_i^{-1}) \qquad (9)$$

Relevance Vector Machine

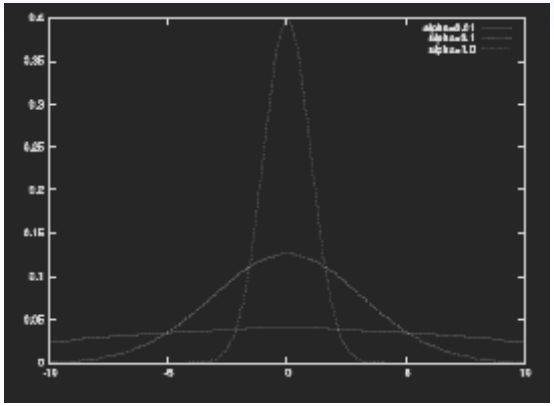


Relevance Vector Machine



The RVM "trick" is to define the inverse variances of these Gaussian distributions, α, as variables, and to infer their values as well. This form of prior is known as an automatic relevance determination (ARD) in Mackay 1994. The inclusion of an ARD prior rewards simplicity.

Relevance Vector Machine



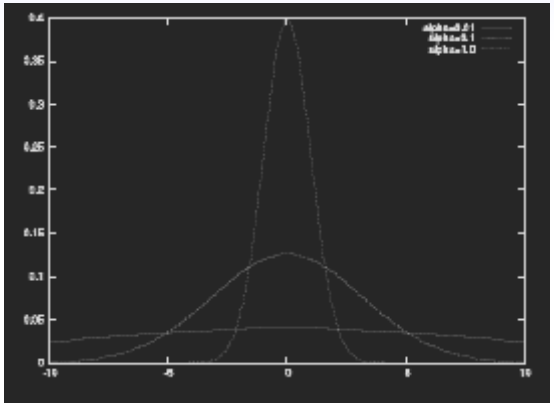
The RVM "trick" is to define the inverse variances of these Gaussian distributions, α, as variables, and to infer their values as well. This form of prior is known as an automatic relevance determination (ARD) in Mackay 1994. The inclusion of an ARD prior rewards simplicity.

To include these new parameter in the inference process, we also need to specify a hyperprior over values of α. For the RVM, a very broad gamma distribution is used.

Considering just a single basis function, there are two possibilities:

• The basis function provides additional information about the specified classification problem. When its weight is set to some non-zero value, the amount of misclassified training data is reduced. This increases the value of equation 7, and therefore the probability of that model given the data.

Relevance Vector Machine



The RVM "trick" is to define the inverse variances of these Gaussian distributions, α, as variables, and to infer their values as well. This form of prior is known as an automatic relevance determination (ARD) in Mackay 1994. The inclusion of an ARD prior rewards simplicity.

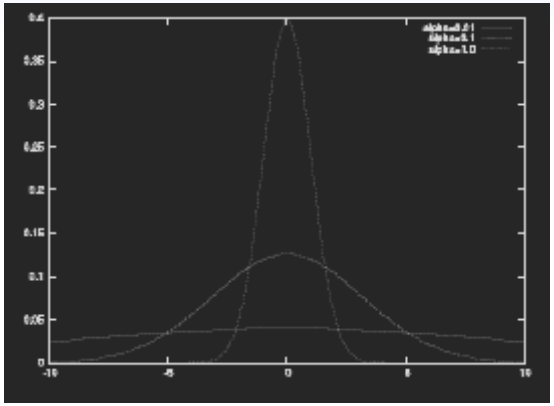
To include these new parameter in the inference process, we also need to specify a hyperprior over values of α. For the RVM, a very broad gamma distribution is used.

Considering just a single basis function, there are two possibilities:

• The basis function provides additional information about the specified classification problem. When its weight is set to some non-zero value, the amount of misclassified training data is reduced. This increases the value of equation 7, and therefore the probability of that model given the data.

• If the basis function provides no information because it is irrelevant to the problem, there is no value of the weight that will lead to a significant increase in the likelihood. At this point, the prior term in the model comes into play: by setting the α_i parameter to a large value, the prior distribution $P(W_i)$ becomes sharply peaked around zero. By then setting Wi to zero, the posterior probability of the model is maximized.

Relevance Vector Machine



The RVM "trick" is to define the inverse variances of these Gaussian distributions, α, as variables, and to infer their values as well. This form of prior is known as an automatic relevance determination (ARD) in Mackay 1994. The inclusion of an ARD prior rewards simplicity.

To include these new parameter in the inference process, we also need to specify a hyperprior over values of α. For the RVM, a very broad gamma distribution is used.

Considering just a single basis function, there are two possibilities:

• The basis function provides additional information about the specified classification problem. When its weight is set to some non-zero value, the amount of misclassified training data is reduced. This increases the value of equation 7, and therefore the probability of that model given the data.

• If the basis function provides no information because it is irrelevant to the problem, there is no value of the weight that will lead to a significant increase in the likelihood. At this point, the prior term in the model comes into play: by setting the α_i parameter to a large value, the prior distribution $P(W_i)$ becomes sharply peaked around zero. By then setting Wi to zero, the posterior probability of the model is maximized.

•Similarly, when two basis functions offer redundant information, the posterior is maximized by using only one of them in the model. When a basis function has a sufficiently high α, it can be marked as irrelevant, and removed from the model. As a result, the RVM will learn simple models even when presented with a large starting set of basis functions. In addition, the computational cost if each iteration falls with the number of dimensions under consideration.

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

Tipping 2001

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

This is a highly effective mechanism for avoiding overfitting, which leads to good generalisation, and which furthermore results in a sparse model dependent only on a subset of kernel functions: those associated with training examples that lie either on the margin or on the 'wrong' side of it.

Tipping 2001

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

This is a highly effective mechanism for avoiding overfitting, which leads to good generalisation, and which furthermore results in a sparse model dependent only on a subset of kernel functions: those associated with training examples that lie either on the margin or on the 'wrong' side of it.

However, despite its success, we can identify a number of significant and practical disadvantages:

•Although relatively sparse, SVMs make unnecessarily liberal use of basis functions since the number of support vectors required typically grows linearly with the size of the training set.

Tipping 2001

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

This is a highly effective mechanism for avoiding overfitting, which leads to good generalisation, and which furthermore results in a sparse model dependent only on a subset of kernel functions: those associated with training examples that lie either on the margin or on the 'wrong' side of it.

However, despite its success, we can identify a number of significant and practical disadvantages:

•Although relatively sparse, SVMs make unnecessarily liberal use of basis functions since the number of support vectors required typically grows linearly with the size of the training set.

•Predictions are not probabilistic. In regression the SVM outputs a point estimate, and in classification, a 'hard' binary decision. Ideally, we desire to estimate the conditional distribution in order to capture uncertainty in our prediction. Posterior probability estimates have been coerced from SVMs via post-processing (Platt, 2000), although we argue that these estimates are unreliable.

Tipping 2001

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

This is a highly effective mechanism for avoiding overfitting, which leads to good generalisation, and which furthermore results in a sparse model dependent only on a subset of kernel functions: those associated with training examples that lie either on the margin or on the 'wrong' side of it.

However, despite its success, we can identify a number of significant and practical disadvantages:

•Although relatively sparse, SVMs make unnecessarily liberal use of basis functions since the number of support vectors required typically grows linearly with the size of the training set.

•Predictions are not probabilistic. In regression the SVM outputs a point estimate, and in classification, a 'hard' binary decision. Ideally, we desire to estimate the conditional distribution in order to capture uncertainty in our prediction. Posterior probability estimates have been coerced from SVMs via post-processing (Platt, 2000), although we argue that these estimates are unreliable.

•It is necessary to estimate the error/margin trade-off parameter 'C'. This generally entails a cross-validation.

Tipping 2001

[Relevance or Support] Vector Machine?

SVM key feature, in the classification case: its target function attempts to minimise a measure of error on the training set while simultaneously maximising the 'margin' between the two classes (in the feature space implicitly defined by the kernel).

This is a highly effective mechanism for avoiding overfitting, which leads to good generalisation, and which furthermore results in a sparse model dependent only on a subset of kernel functions: those associated with training examples that lie either on the margin or on the 'wrong' side of it.

However, despite its success, we can identify a number of significant and practical disadvantages:

•Although relatively sparse, SVMs make unnecessarily liberal use of basis functions since the number of support vectors required typically grows linearly with the size of the training set.

•Predictions are not probabilistic. In regression the SVM outputs a point estimate, and in classification, a 'hard' binary decision. Ideally, we desire to estimate the conditional distribution in order to capture uncertainty in our prediction. Posterior probability estimates have been coerced from SVMs via post-processing (Platt, 2000), although we argue that these estimates are unreliable.

•It is necessary to estimate the error/margin trade-off parameter 'C'. This generally entails a cross-validation.

RVM - none of the above limitations:
•Exploits overall fewer basis functions
•Increased sparsity
•Avoid overfitting
•Simpler models
•Probabilistic Bayesian Learning with posterior probability estimates

Tipping 2001

SVM

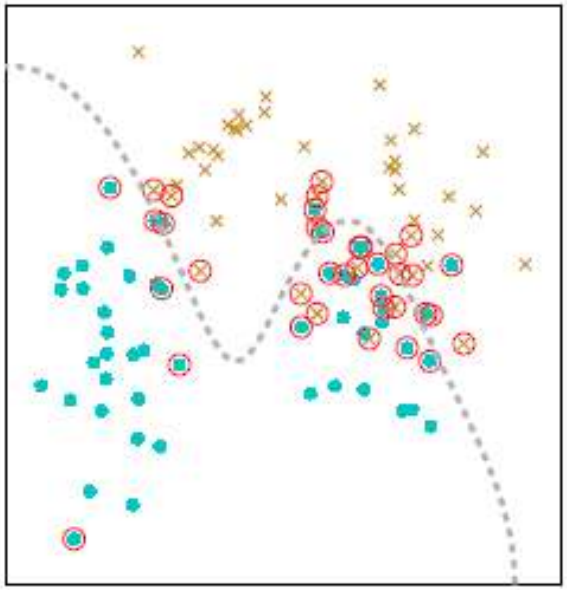
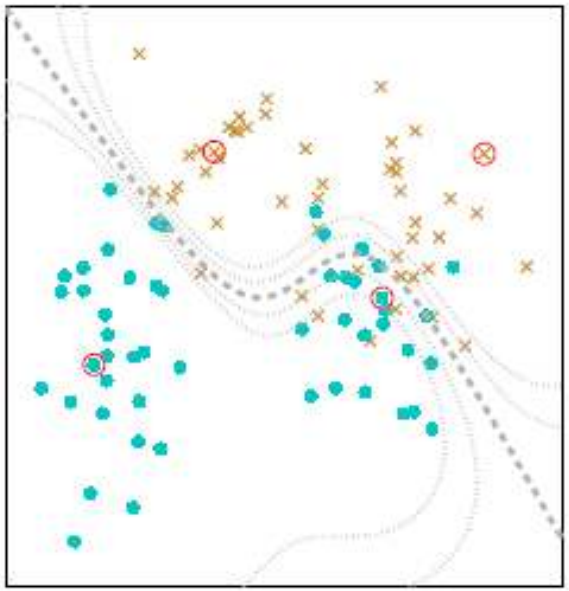


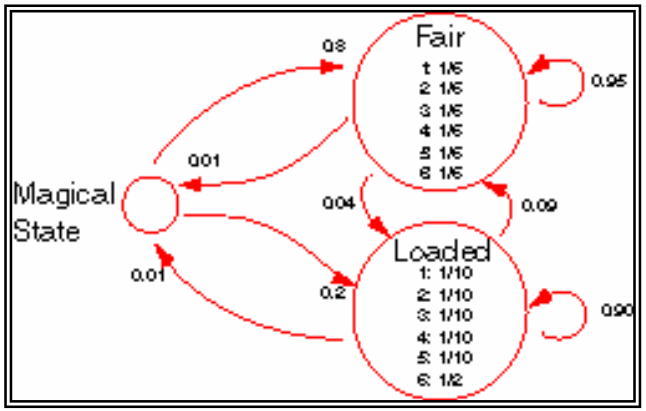
Figure 5: Support vector classifier of the Ripley for which there are 38 kernel functions.



RVM

Figure 6: Variational relevance vector classifier of the Ripley dataset for which there are 4 kernel functions.

Biojava Project



http://biojava.org/wiki/Main_Page

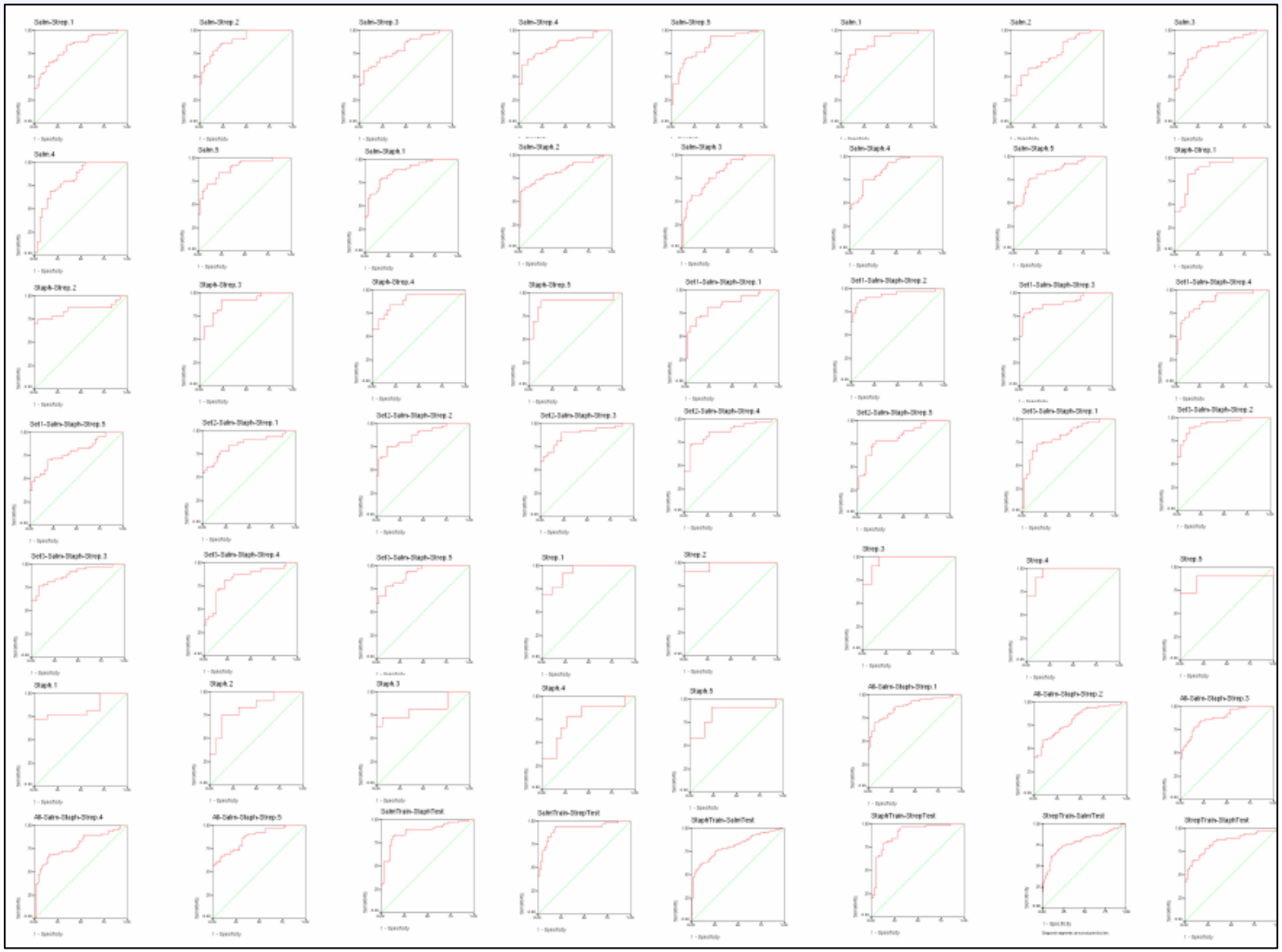
Specificity and Sensitivity

$$Sp = \frac{TN}{TN + FP} \quad (1)$$

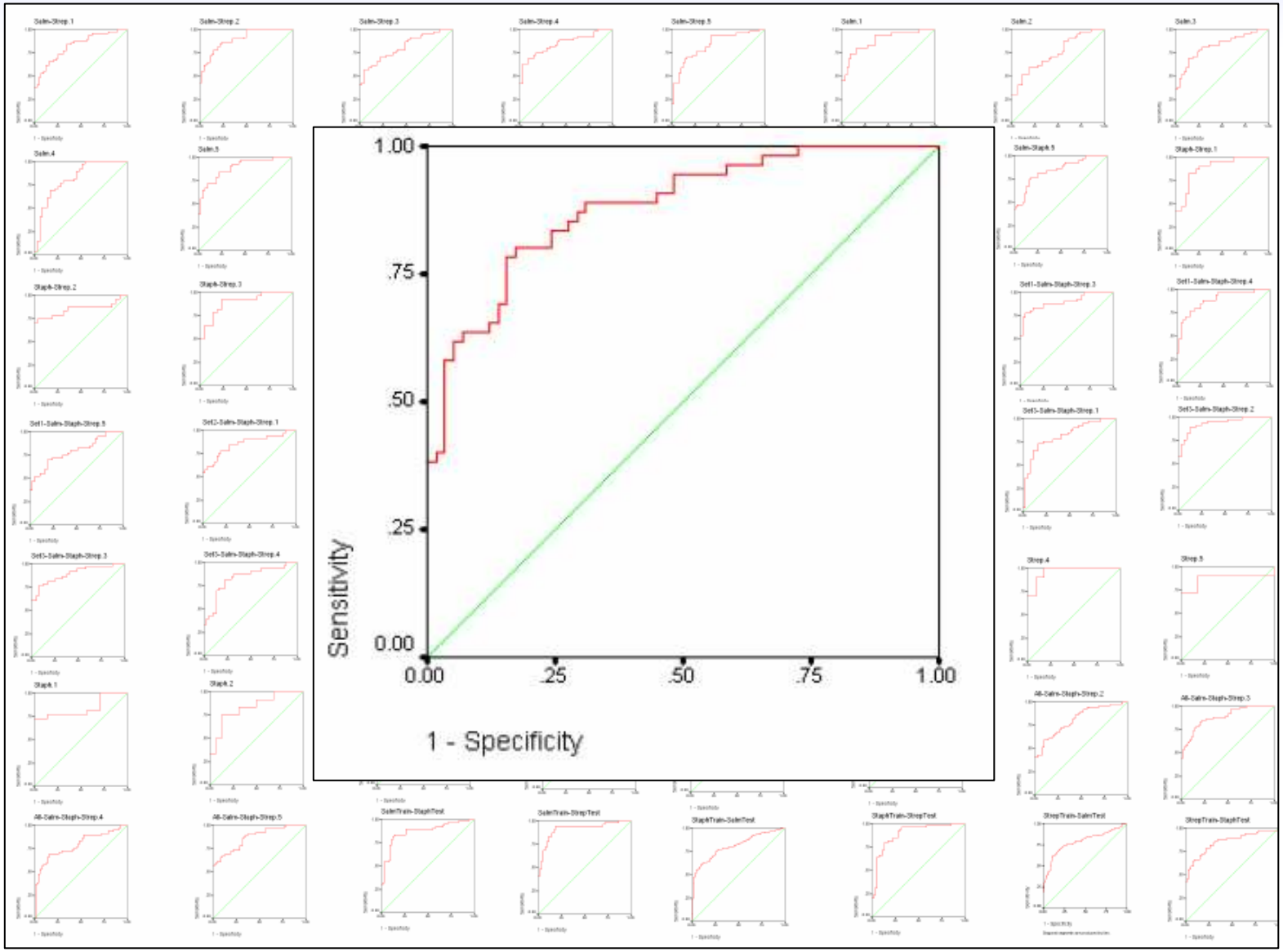
$$Sn = \frac{TP}{TP + FN} \quad (2)$$

$$Ac = \frac{TP + TN}{(TP + FP) + (TN + FN)} \quad (3)$$

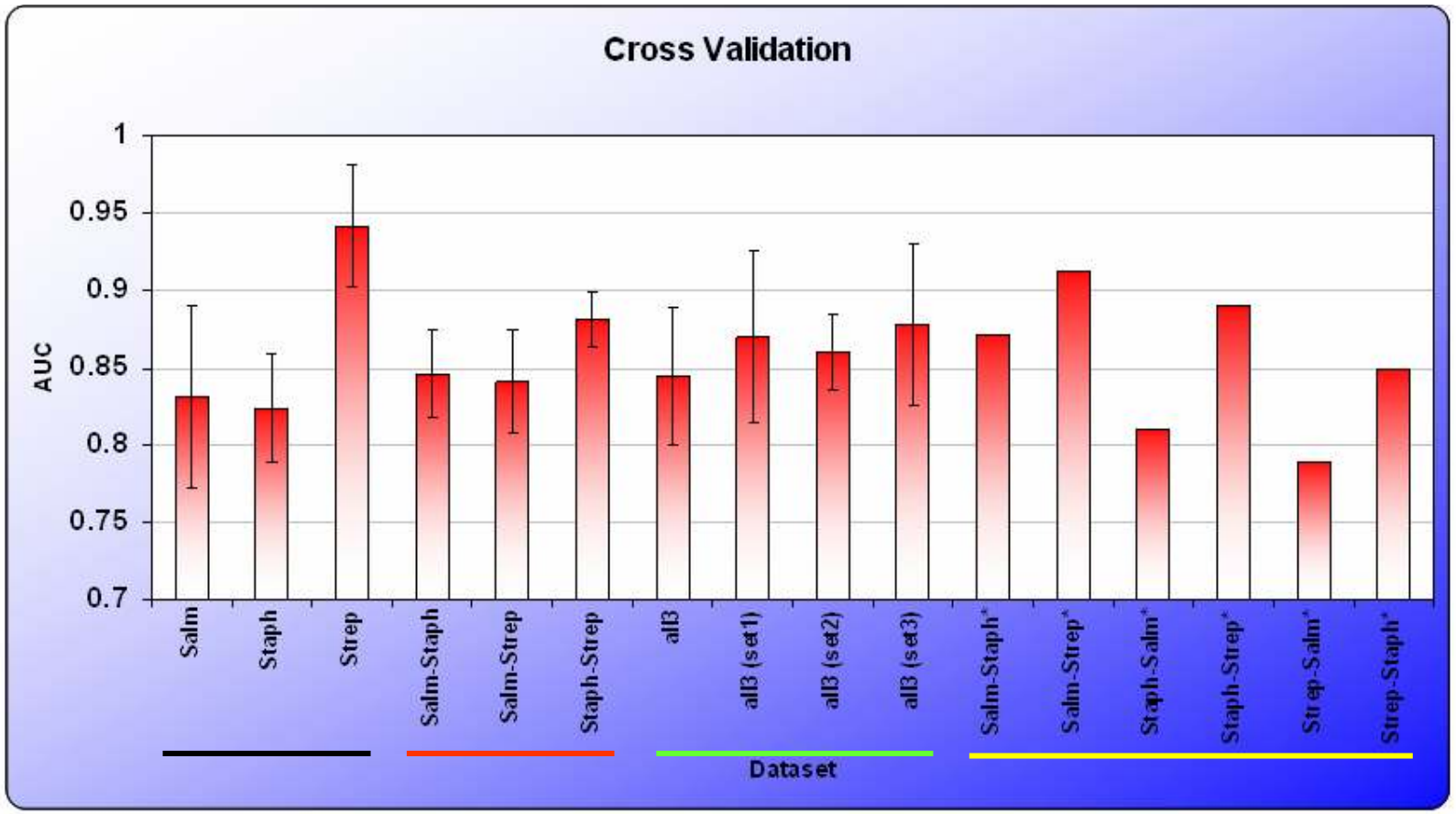
Receiver Operating Characteristic Curve



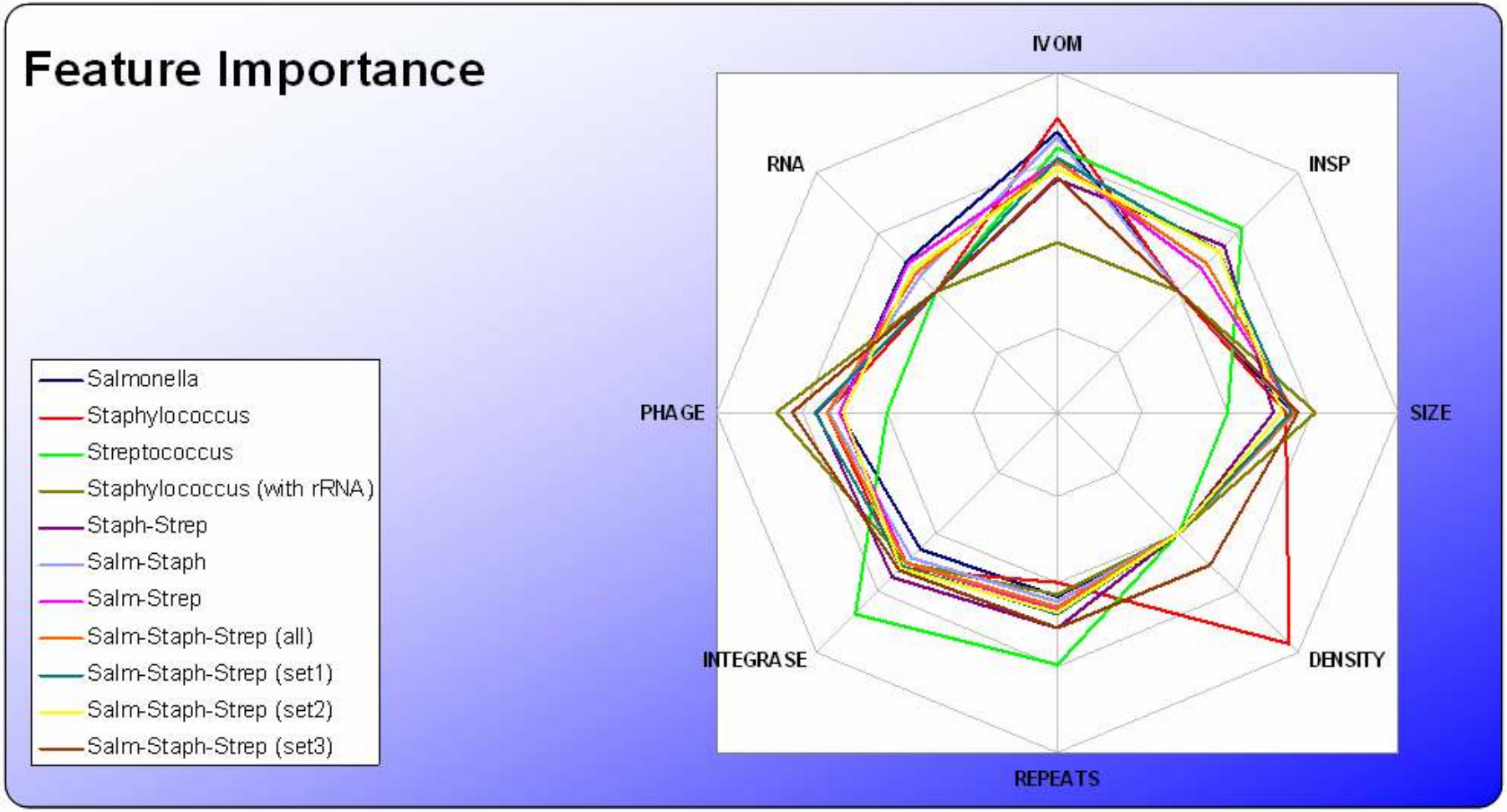
Receiver Operating Characteristic Curve



Cross Validation

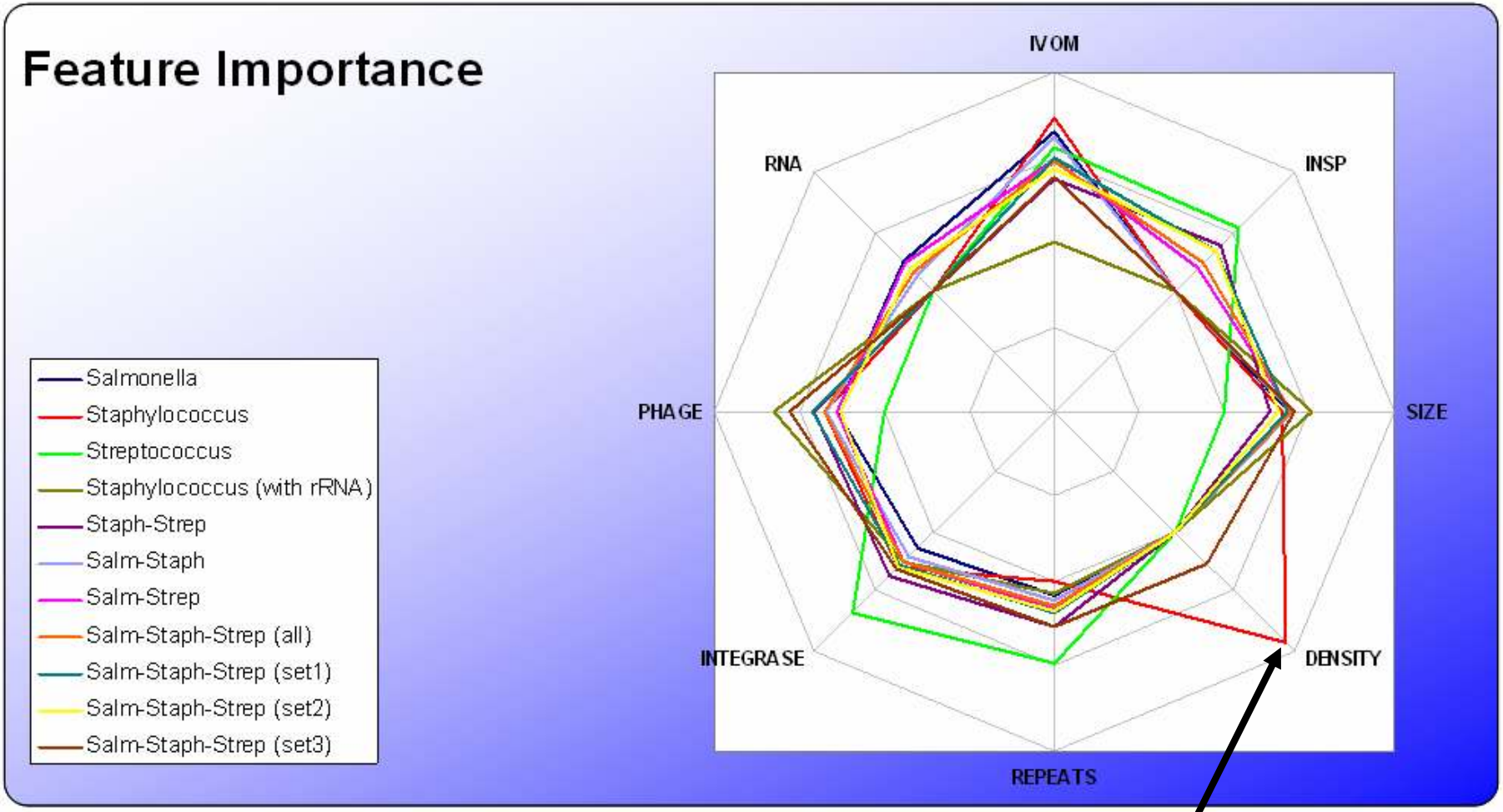


GI structural models



Vernikos GS et al., Genome Res 2008

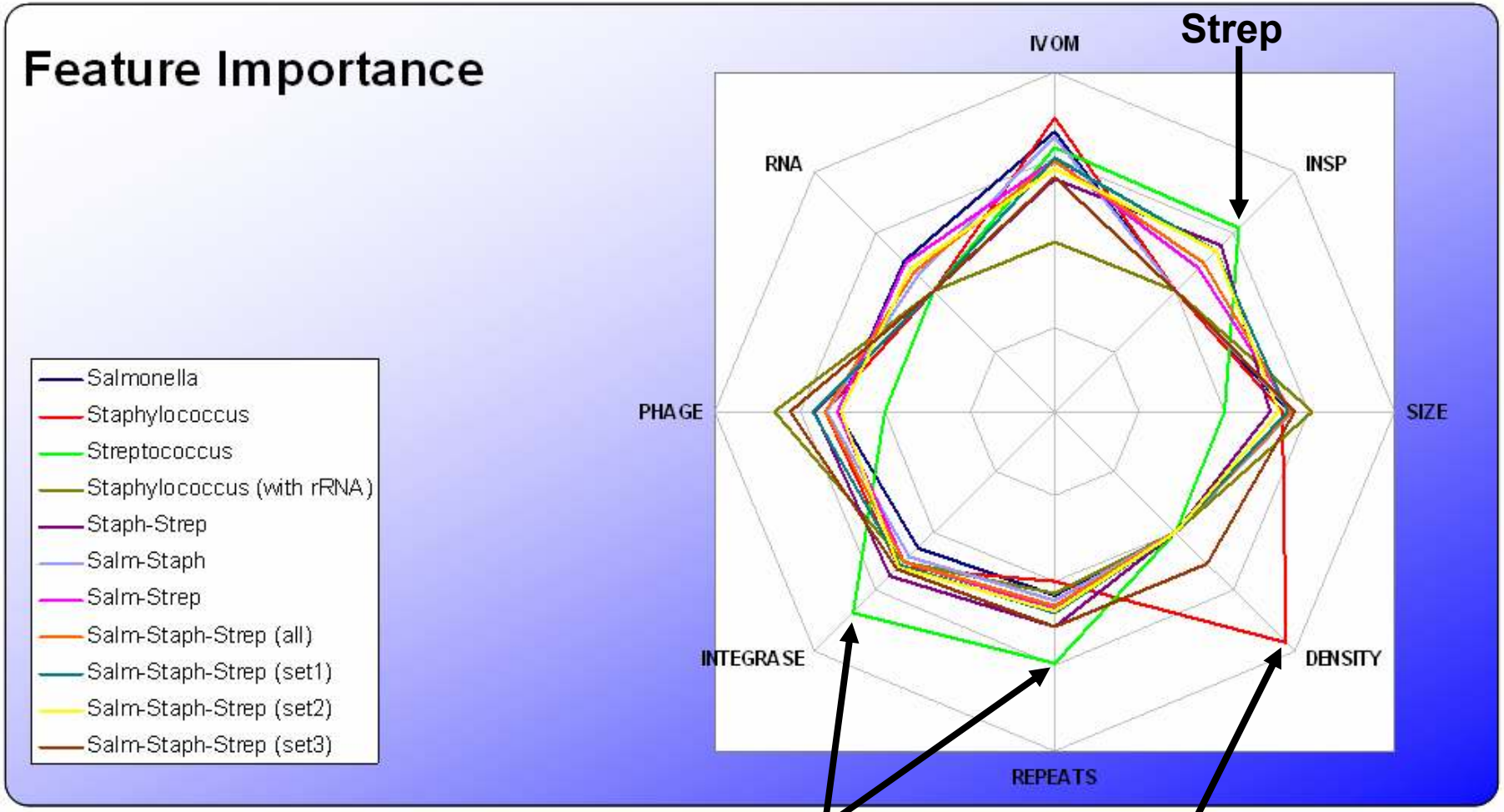
GI structural models



Staph

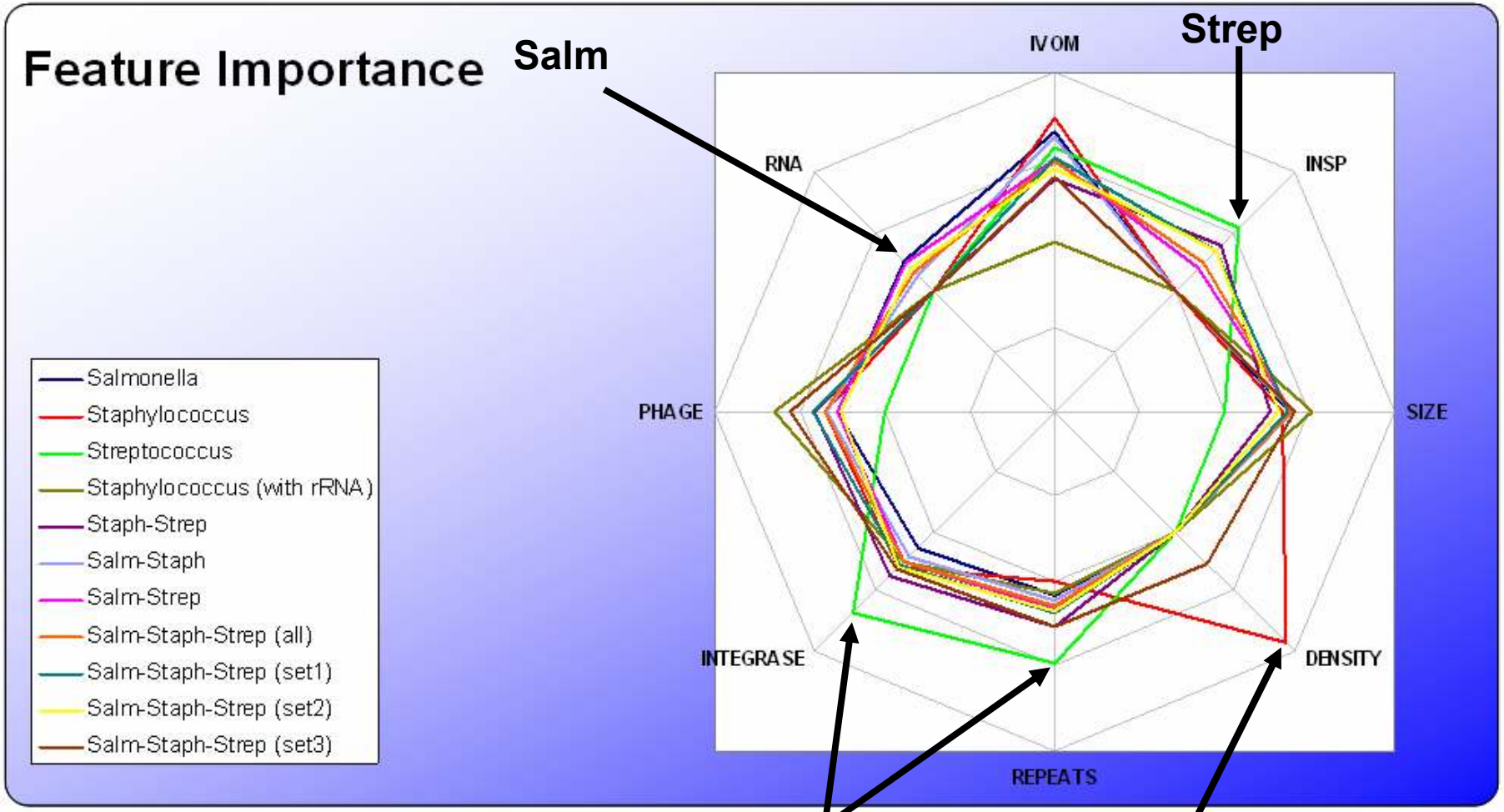
Vernikos GS et al., Genome Res 2008

GI structural models



Vernikos GS et al., Genome Res 2008

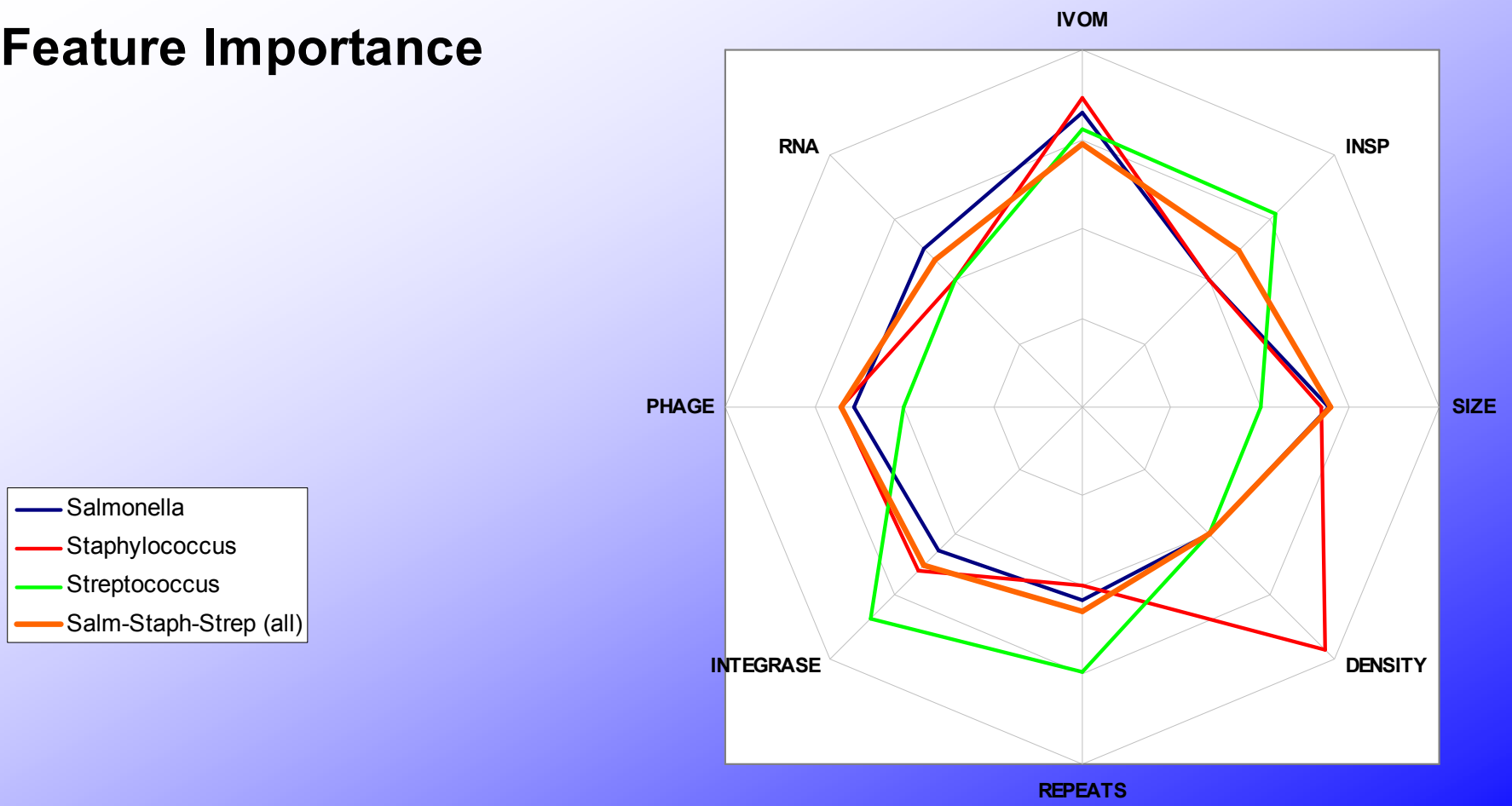
GI structural models



Vernikos GS et al., Genome Res 2008

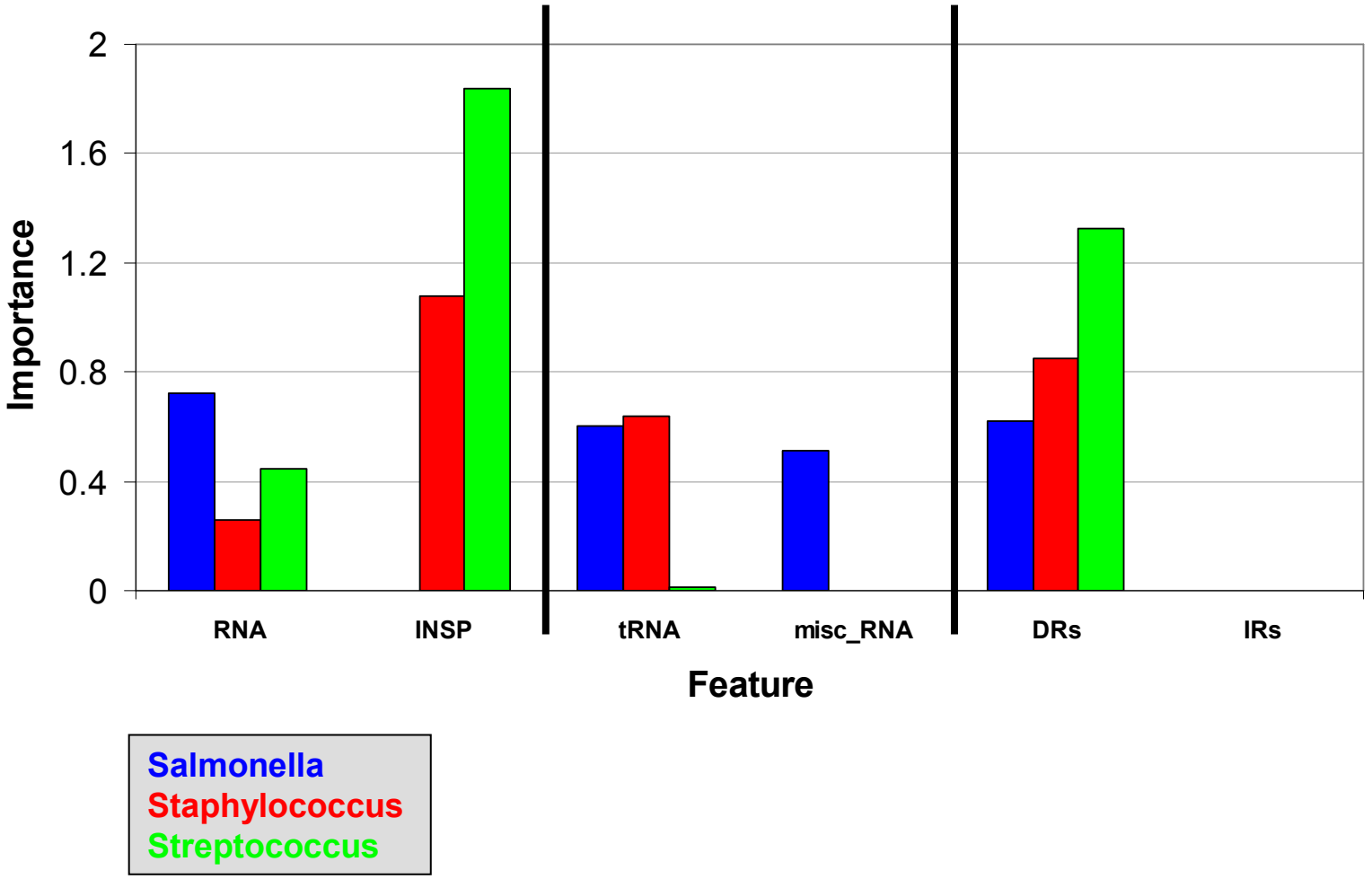
GI structural models

Feature Importance

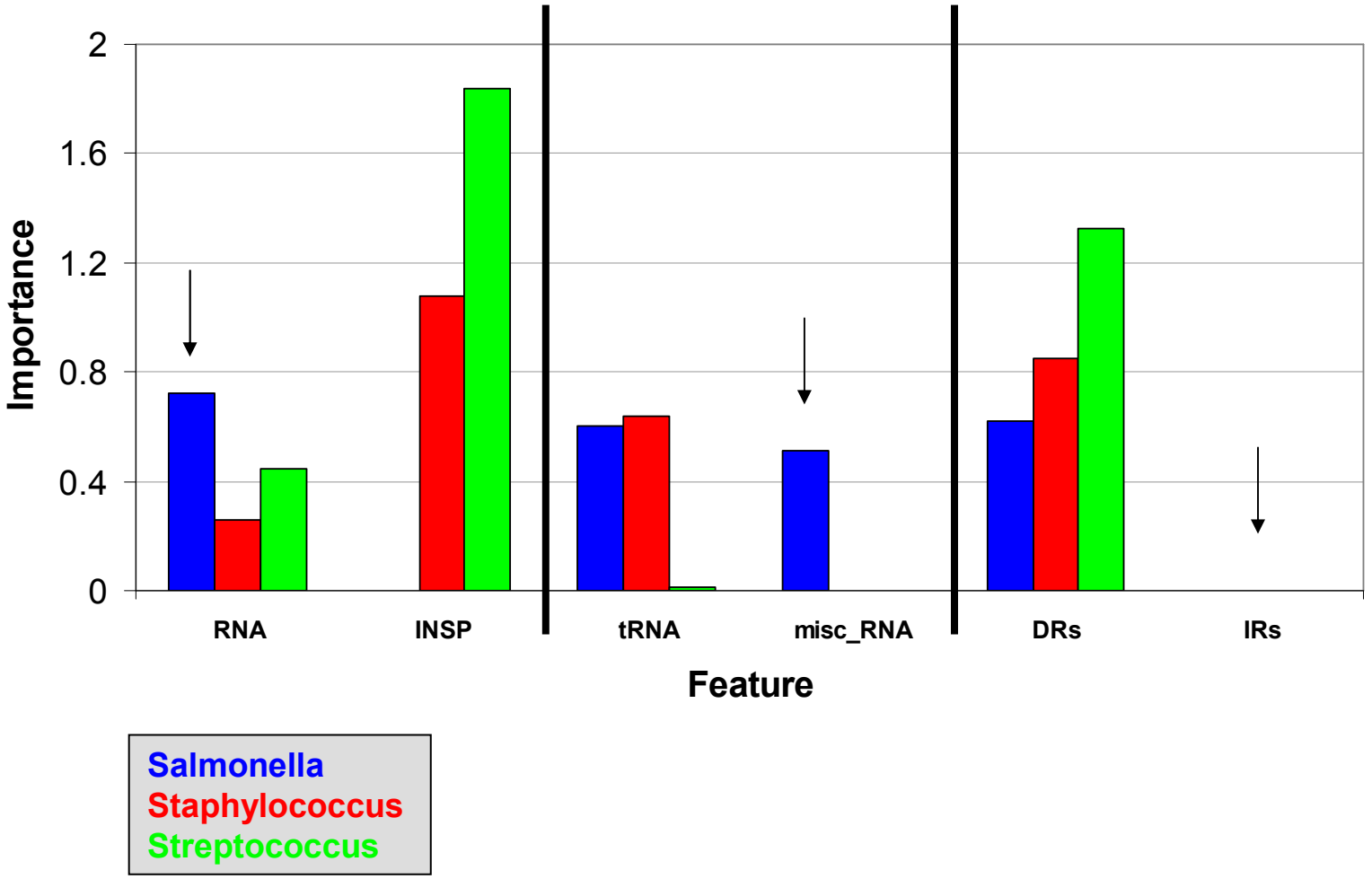


Vernikos GS et al., Genome Res 2008

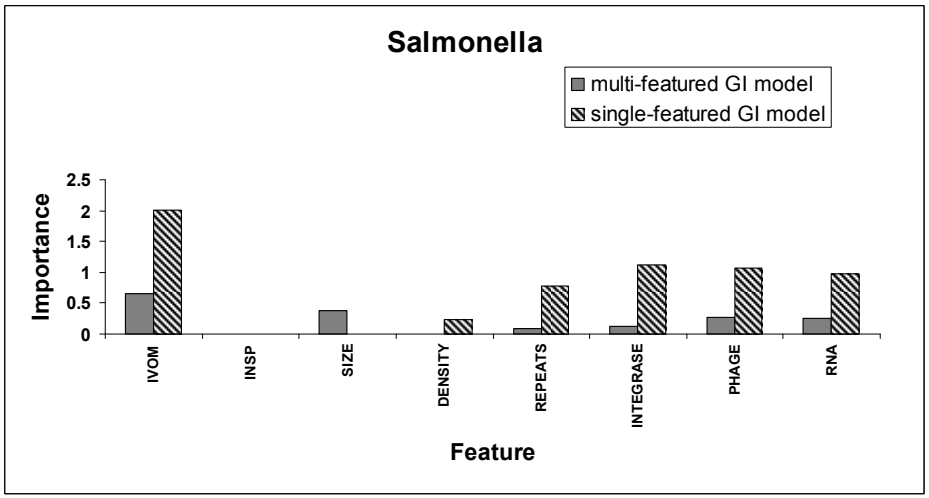
GI structural models (higher resolution)

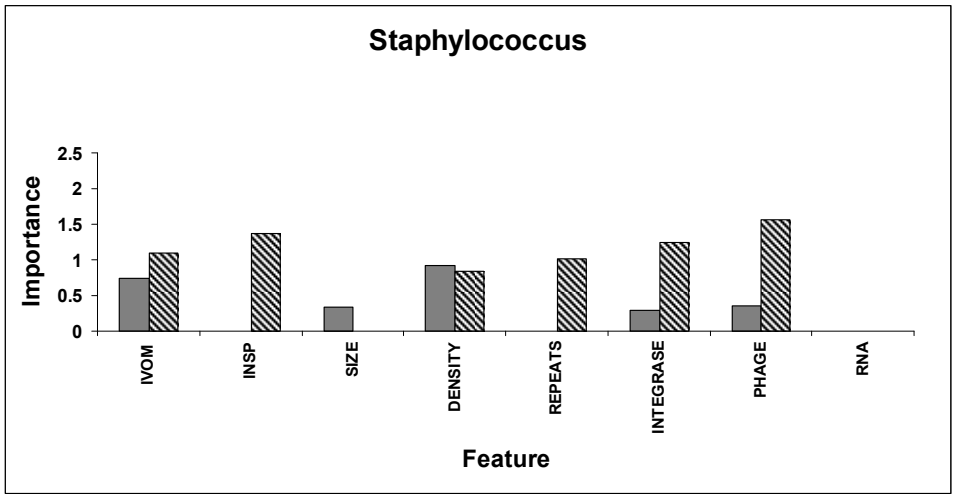


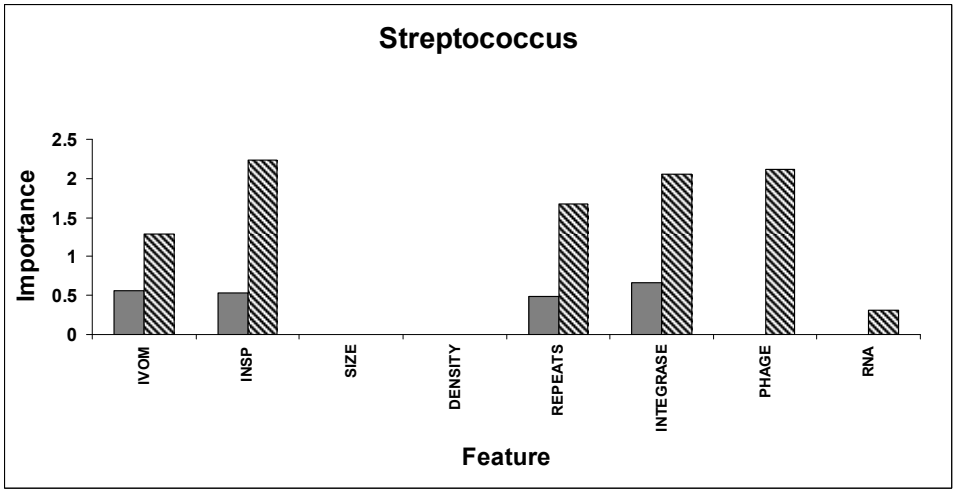
GI structural models (higher resolution)



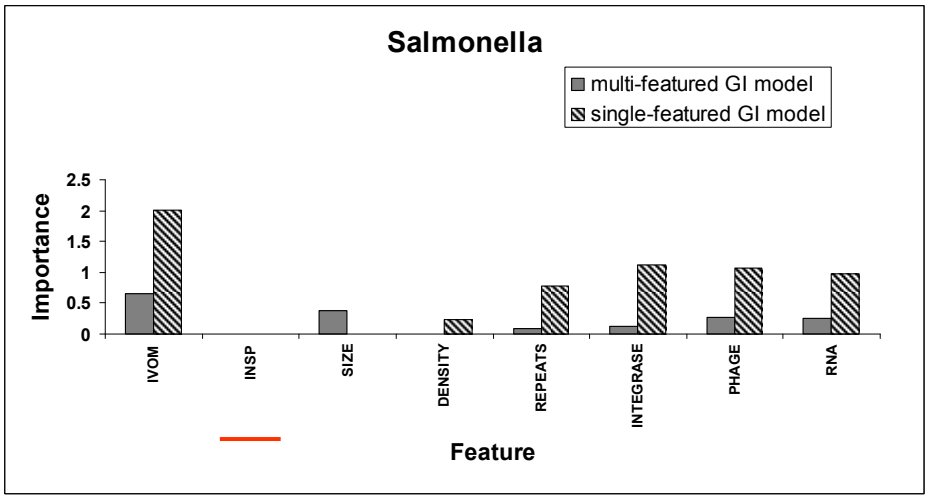
Multi vs single-featured GI structural models

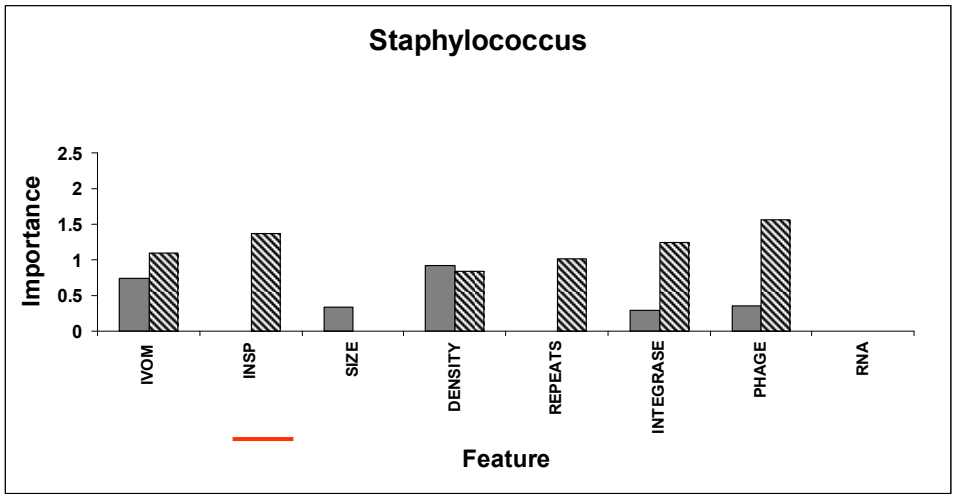


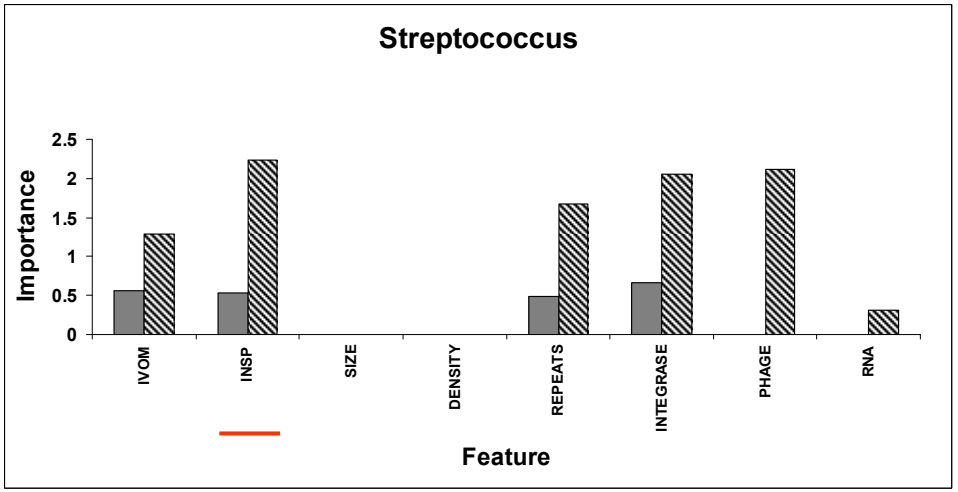




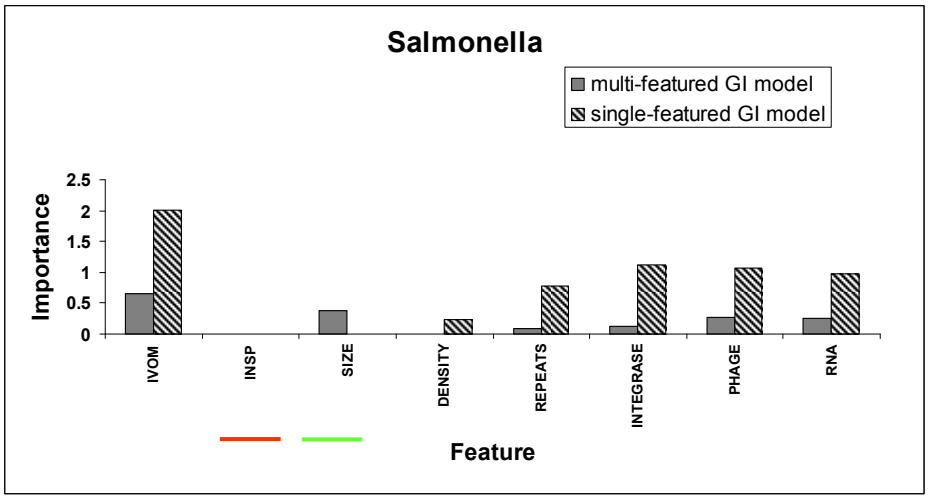
Multi vs single-featured GI structural models

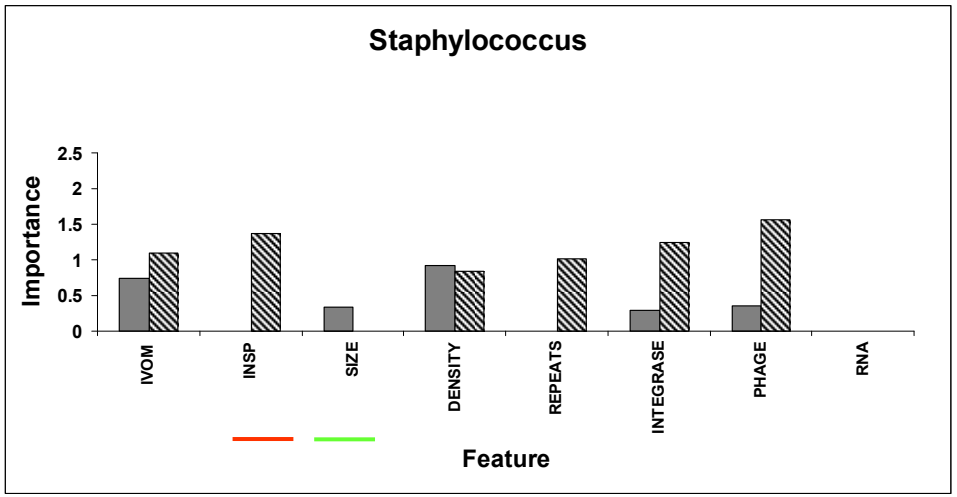


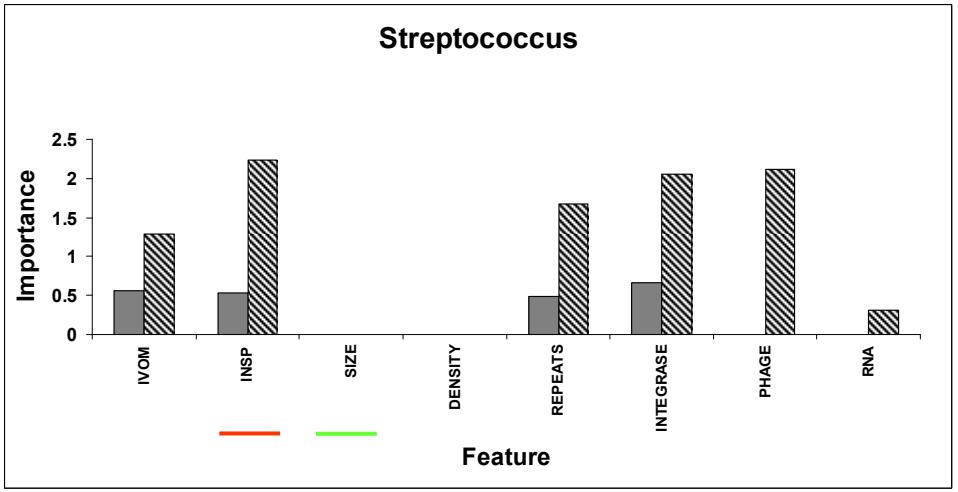




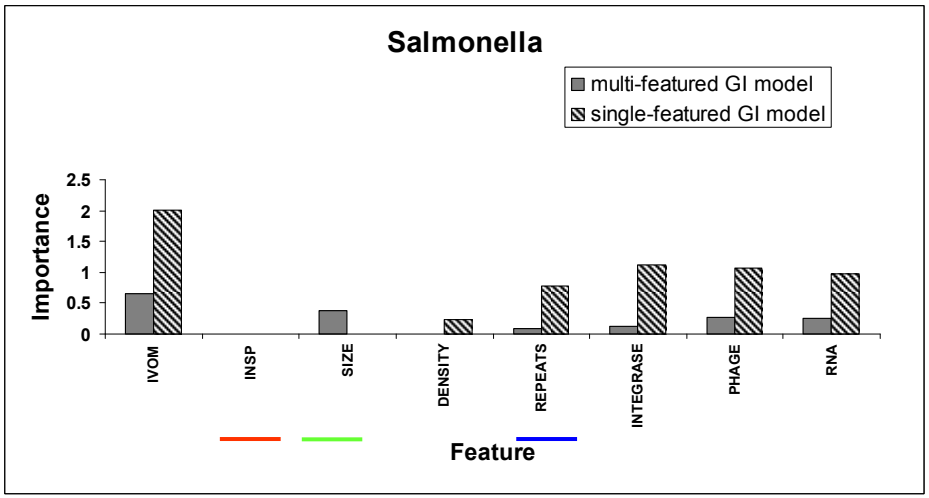
Multi vs single-featured GI structural models

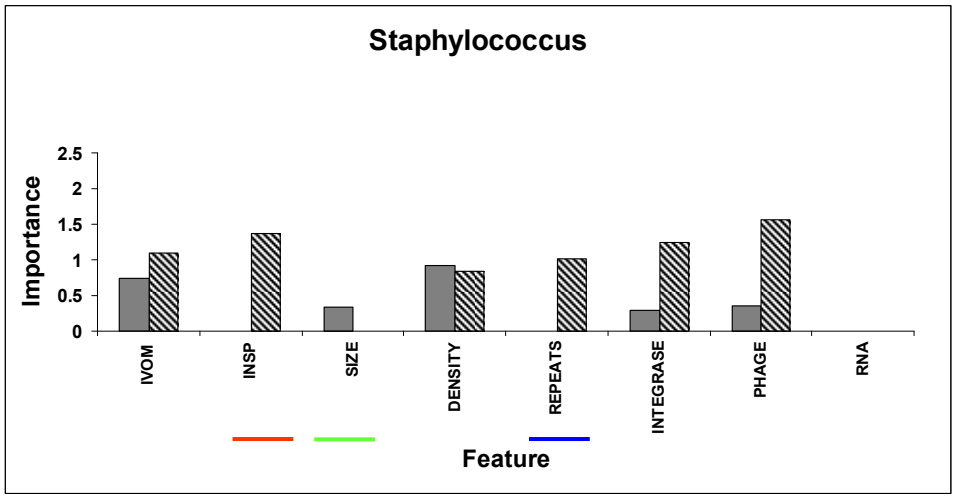


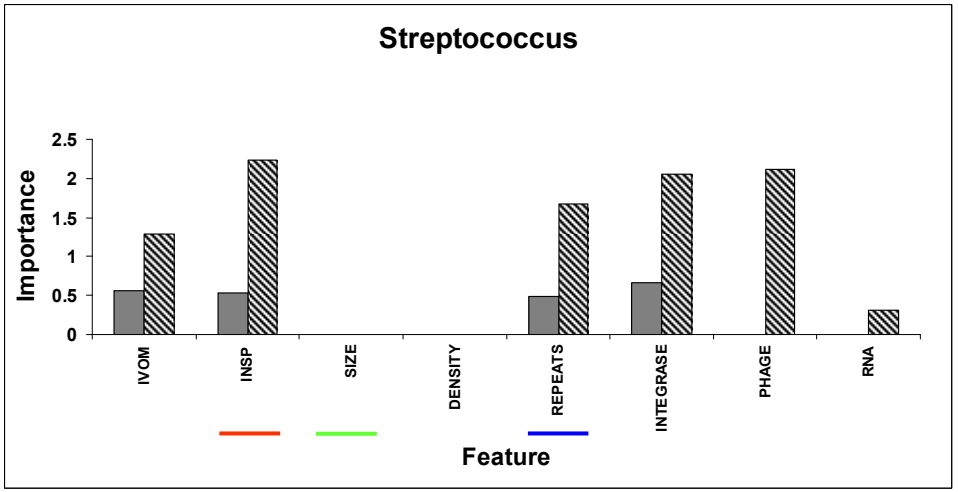




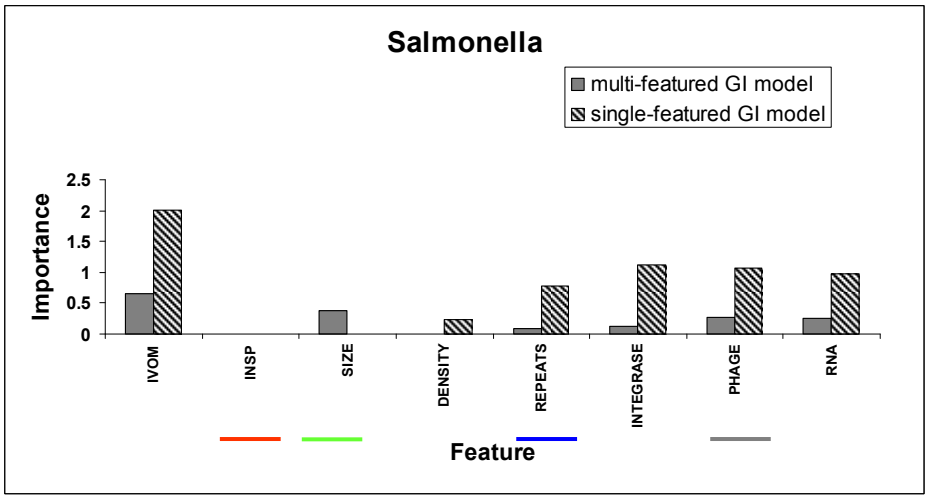
Multi vs single-featured GI structural models

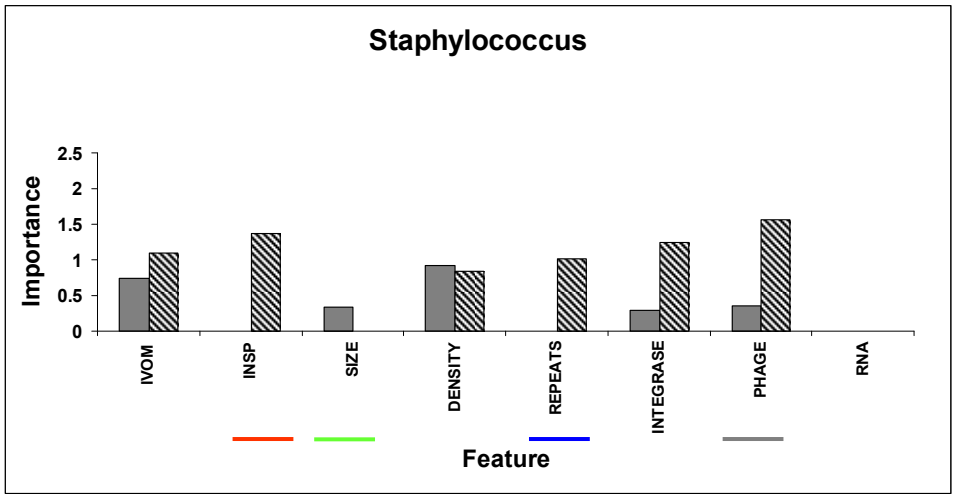


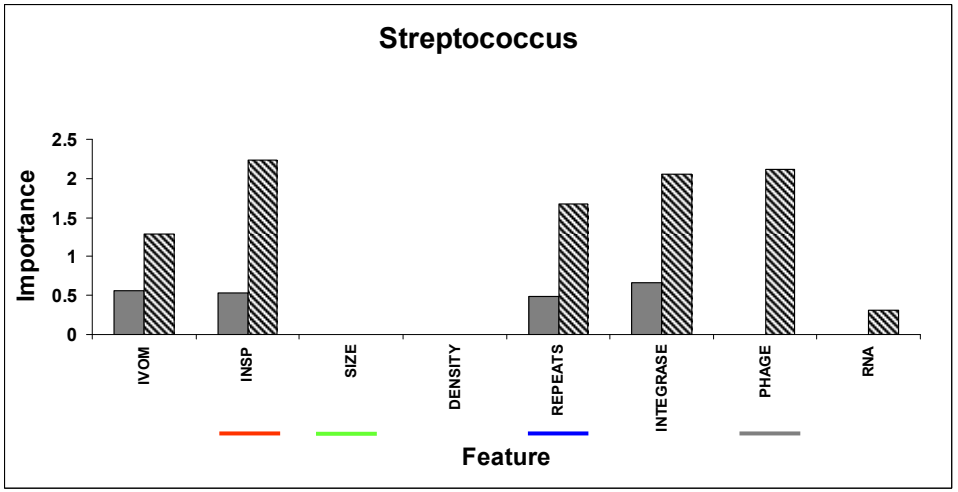




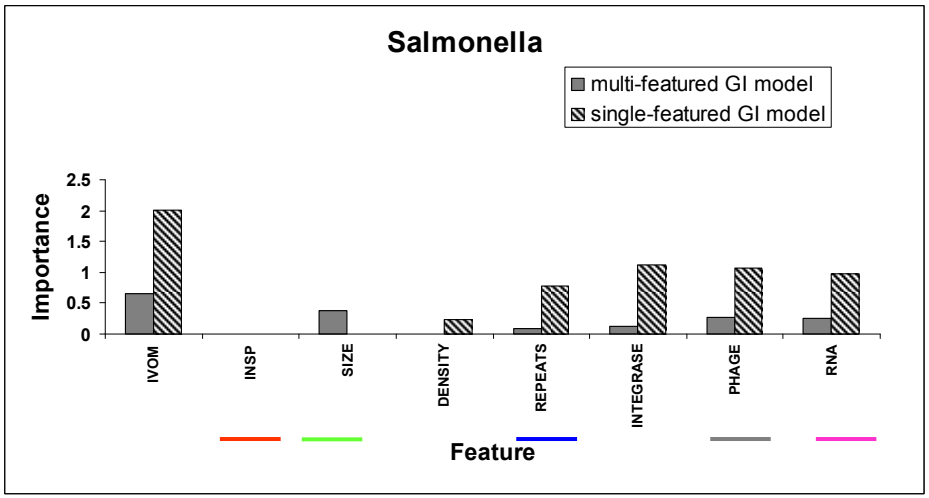
Multi vs single-featured GI structural models

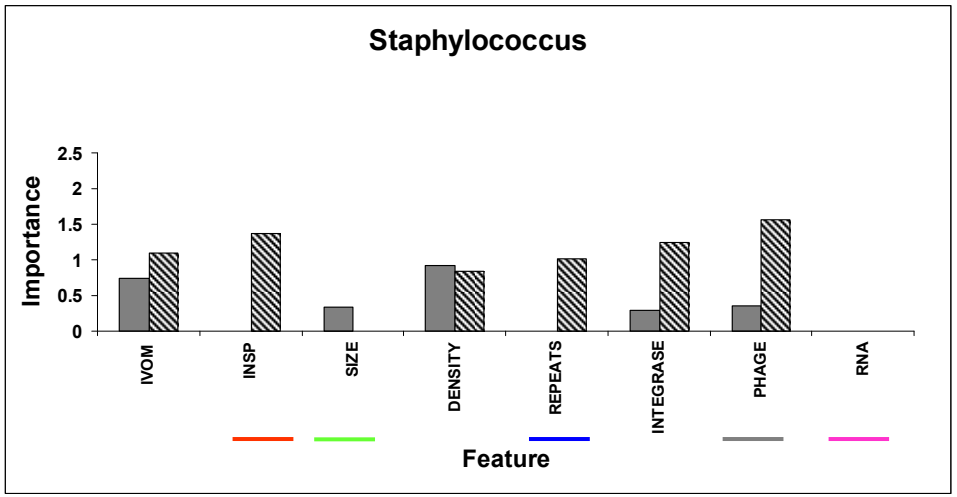


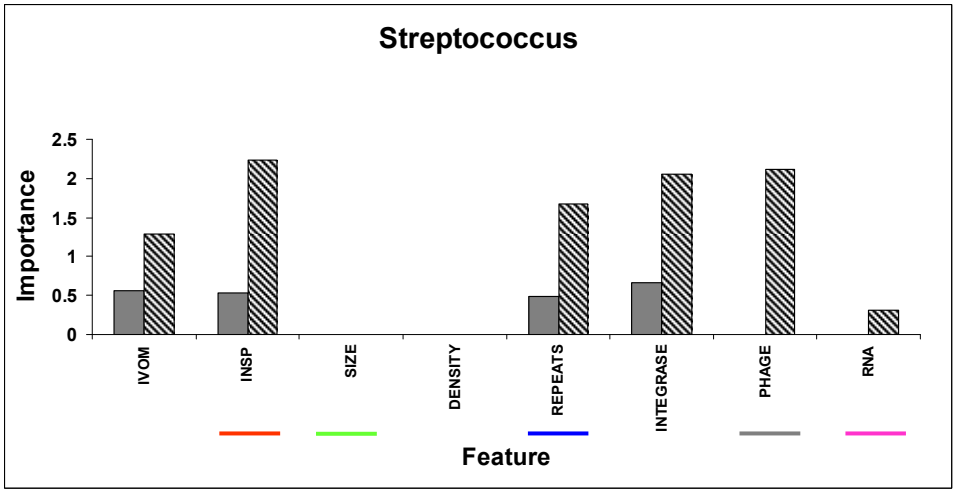




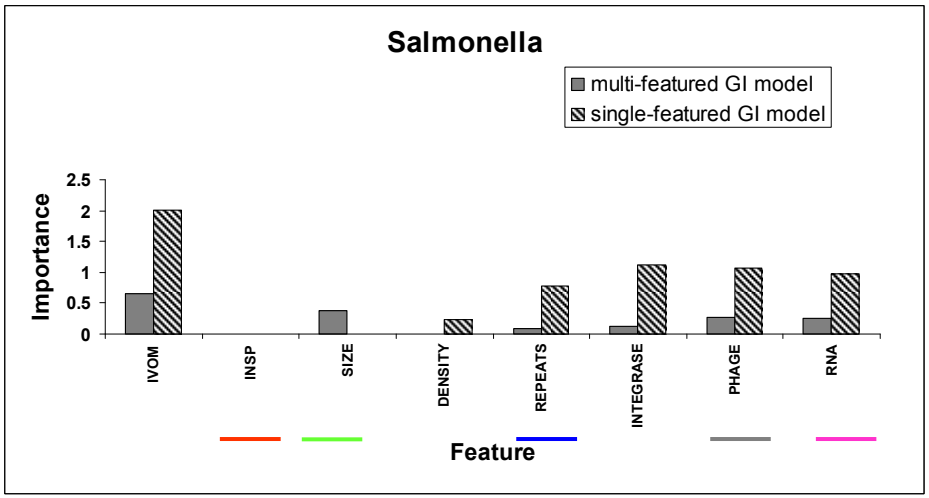
Multi vs single-featured GI structural models

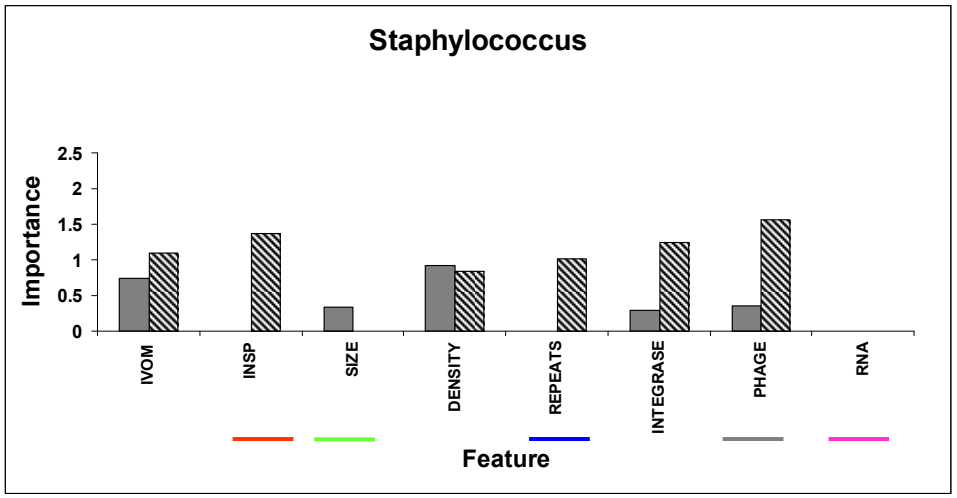


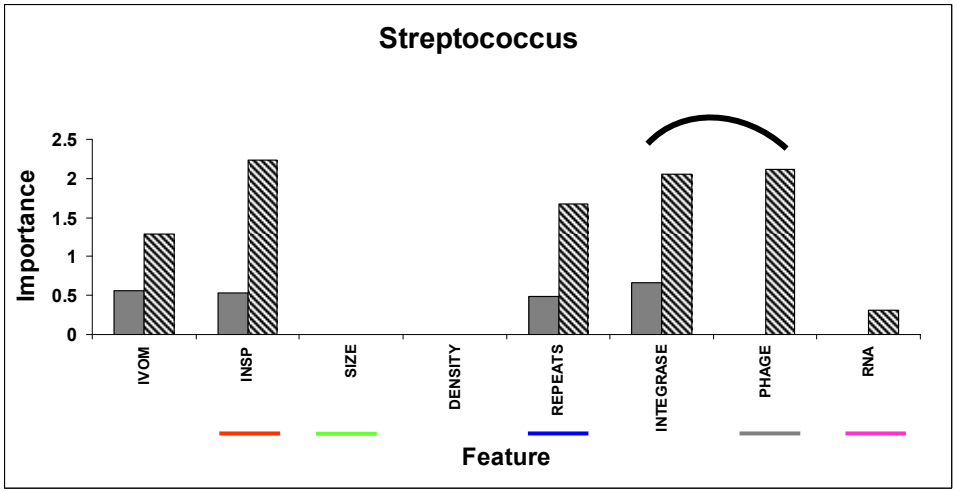




Multi vs single-featured GI structural models







Generalized Linear Models

1) Si = -0.764 + 6.203 (x)IVOM + 0.000(x)INSP + -4.956(x)SIZE + 0.000(x)DENS + 0.635(x)REPEATS + 0.995(x)INT + 2.086(x)PHAGE + 1.968(x)RNA
2) Si = -2.978 + 4.151 (x)IVOM + 3.219(x)INSP + 0.000(x)SIZE + 0.000(x)DENS + 2.185(x)REPEATS + 3.351(x)INT + 0.000(x)PHAGE + 0.000(x)RNA
3) Si = -0.005 + 0.000 (x)IVOM + 0.000(x)INSP + -4.324(x)SIZE + 0.000(x)DENS + 0.360(x)REPEATS + 1.303(x)INT + 3.995(x)PHAGE + 0.000(x)RNA
4) Si = -4.583 +12.752 (x)IVOM + 0.000(x)INSP + -2.843(x)SIZE + 2.486(x)DENS + 0.000(x)REPEATS + 1.552(x)INT + 2.157(x)PHAGE + 0.000(x)RNA
5) Si = -1.544 + 3.756 (x)IVOM + 2.842(x)INSP + -2.583(x)SIZE + 0.000(x)DENS + 1.297(x)REPEATS + 1.892(x)INT + 2.554(x)PHAGE + 0.000(x)RNA
6) Si = -0.923 + 6.528 (x)IVOM + 0.000(x)INSP + -4.462(x)SIZE + 0.000(x)DENS + 0.771(x)REPEATS + 1.404(x)INT + 2.441(x)PHAGE + 1.159(x)RNA
7) Si = -0.763 + 4.330 (x)IVOM + 2.516(x)INSP + -4.941(x)SIZE + 0.000(x)DENS + 1.030(x)REPEATS + 1.630(x)INT + 2.027(x)PHAGE + 1.842(x)RNA
8) Si = -0.879 + 4.659 (x)IVOM + 2.795(x)INSP + -4.434(x)SIZE + 0.000(x)DENS + 0.897(x)REPEATS + 1.553(x)INT + 2.433(x)PHAGE + 1.319(x)RNA
9) Si = -1.293 + 5.285 (x)IVOM + 3.072(x)INSP + -3.914(x)SIZE + 0.000(x)DENS + 1.007(x)REPEATS + 1.668(x)INT + 2.847(x)PHAGE + 0.000(x)RNA
10) Si = -1.057 + 4.234 (x)IVOM + 3.003(x)INSP + -3.396(x)SIZE + 0.000(x)DENS + 0.927(x)REPEATS + 1.722(x)INT + 1.664(x)PHAGE + 1.539(x)RNA
11) Si = -1.627 + 3.552 (x)IVOM + 0.000(x)INSP + -4.138(x)SIZE + 0.727(x)DENS + 1.449(x)REPEATS + 1.728(x)INT + 3.685(x)PHAGE + 0.000(x)RNA

1) Salmonella
2) Streptococcus
3) Staphylococcus (with rRNA)
4) Staphylococcus
5) Staph-Strep
6) Salm-Staph
7) Salm-Strep
8) Salm-Staph-Strep (all3)
9) Salm-Staph-Strep (set1)
10) Salm-Staph-Strep (set2)
11) Salm-Staph-Strep (set3)

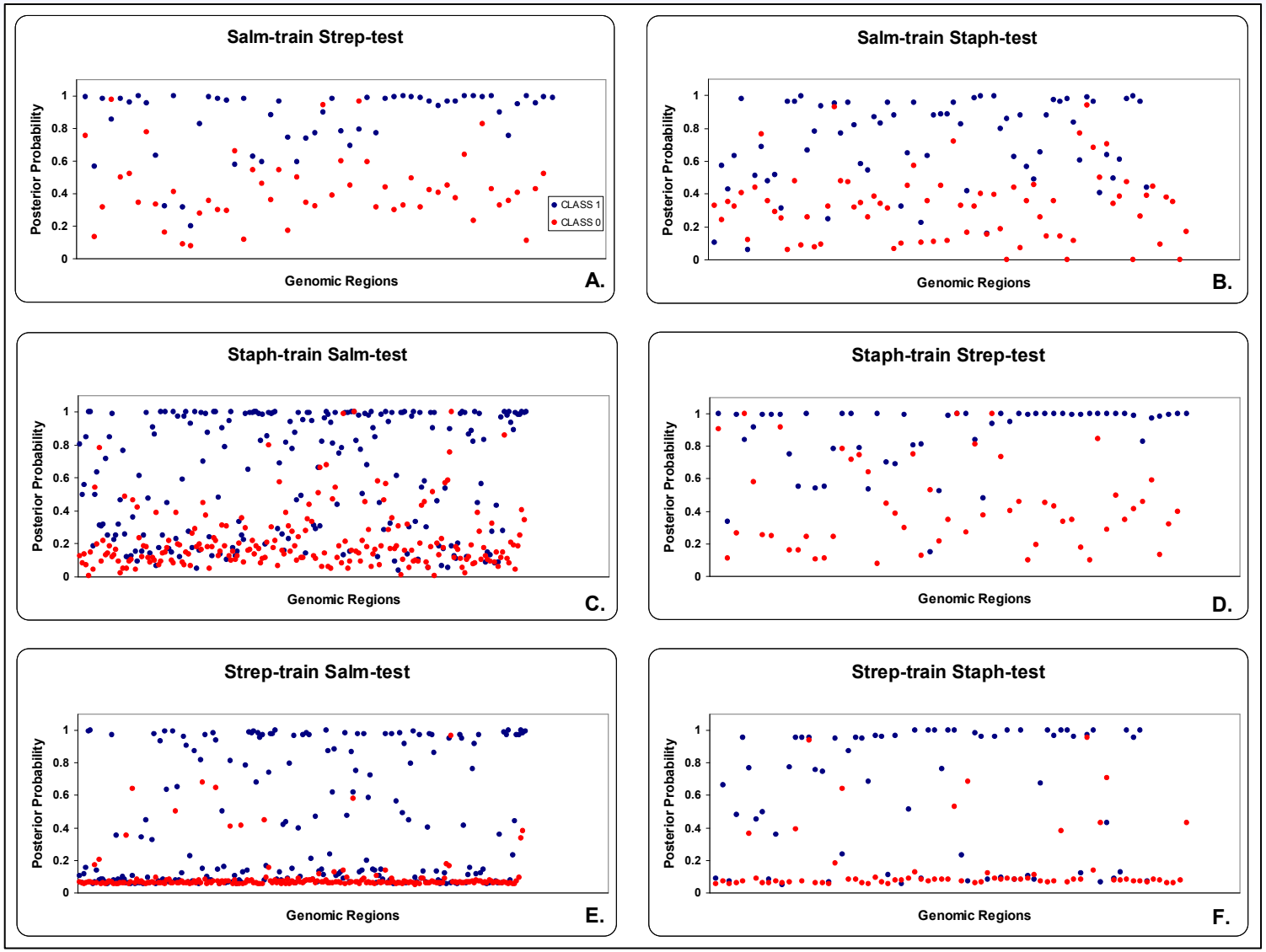
Generalized Linear Models

```
1)  Si = -0.764 + 6.203 (x)IVOM + 0.000(x)INSP + -4.956(x)SIZE + 0.000(x)DENS..
2)  Si = -2.978 + 4.151 (x)IVOM + 3.219(x)INSP +  0.000(x)SIZE + 0.000(x)DENS..
3)  Si = -0.005 + 0.000 (x)IVOM + 0.000(x)INSP + -4.324(x)SIZE + 0.000(x)DENS..
4)  Si = -4.583 +12.752 (x)IVOM + 0.000(x)INSP + -2.843(x)SIZE + 2.486(x)DENS..
5)  Si = -1.544 + 3.756 (x)IVOM + 2.842(x)INSP + -2.583(x)SIZE + 0.000(x)DENS..
6)  Si = -0.923 + 6.528 (x)IVOM + 0.000(x)INSP + -4.462(x)SIZE + 0.000(x)DENS..
7)  Si = -0.763 + 4.330 (x)IVOM + 2.516(x)INSP + -4.941(x)SIZE + 0.000(x)DENS..
8)  Si = -0.879 + 4.659 (x)IVOM + 2.795(x)INSP + -4.434(x)SIZE + 0.000(x)DENS..
9)  Si = -1.293 + 5.285 (x)IVOM + 3.072(x)INSP + -3.914(x)SIZE + 0.000(x)DENS..
10) Si = -1.057 + 4.234 (x)IVOM + 3.003(x)INSP + -3.396(x)SIZE + 0.000(x)DENS..
11) Si = -1.627 + 3.552 (x)IVOM + 0.000(x)INSP + -4.138(x)SIZE + 0.727(x)DENS..
```

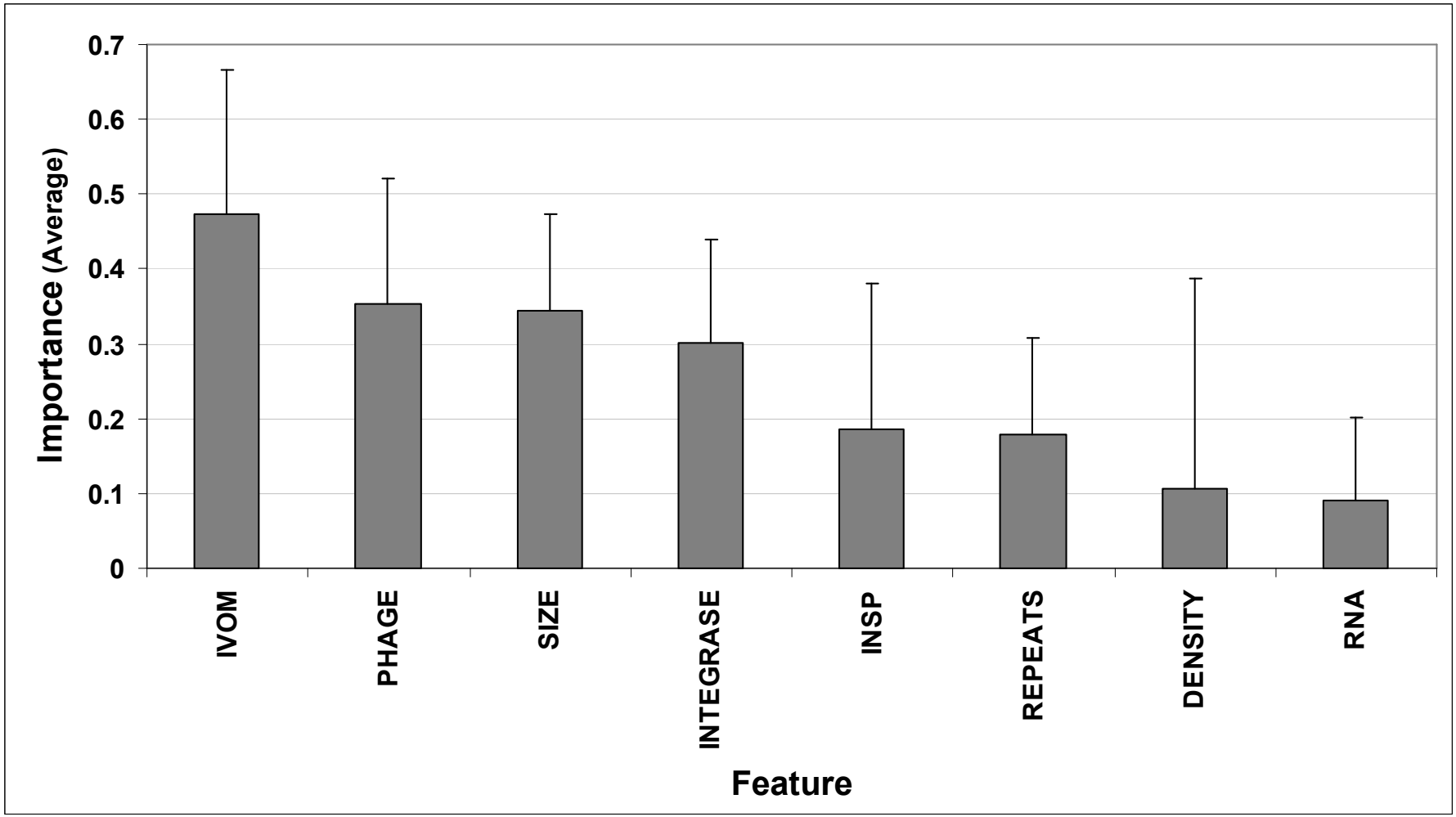
“Genus-blind” cross validation

Class 1

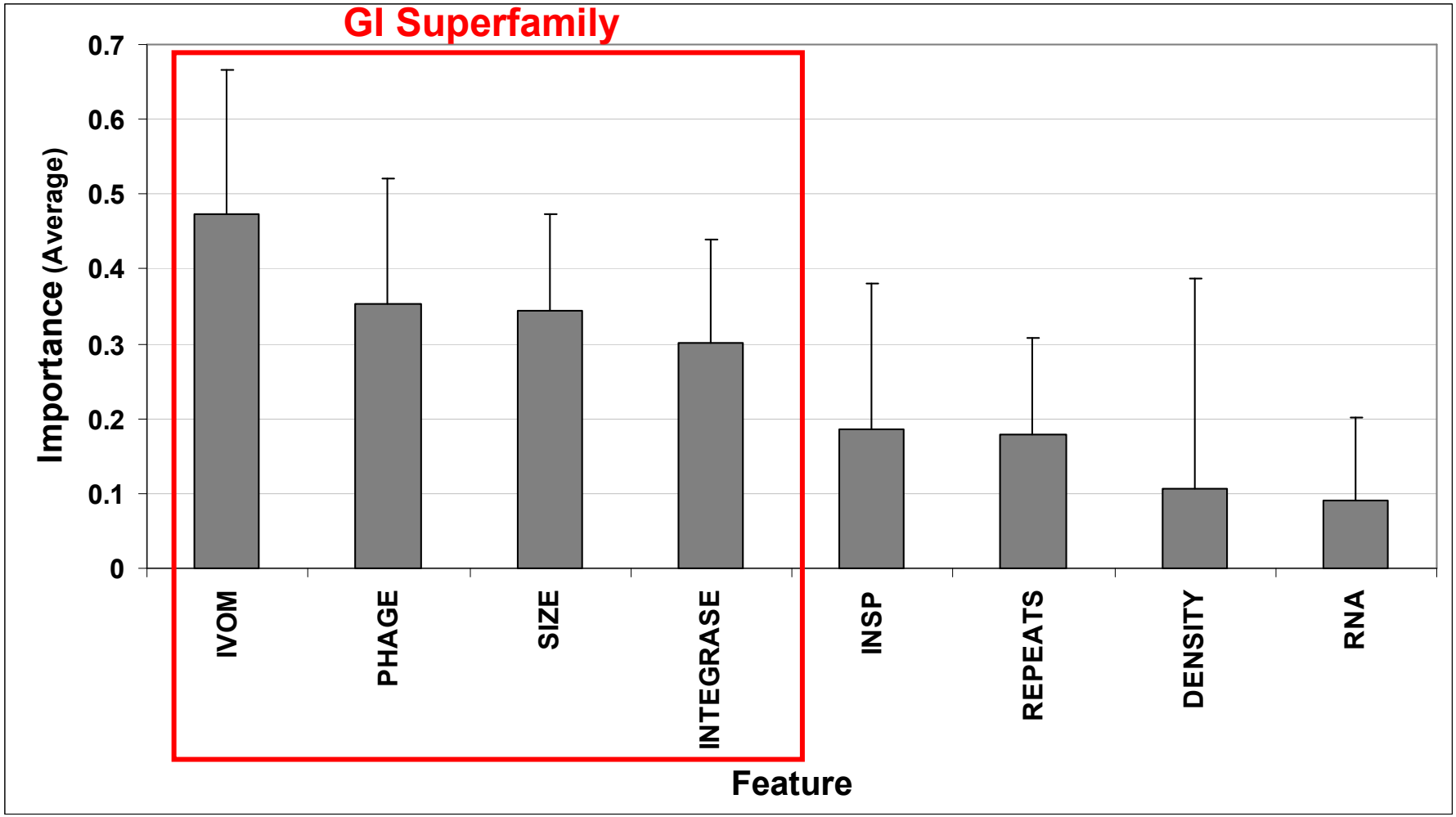
Class 0



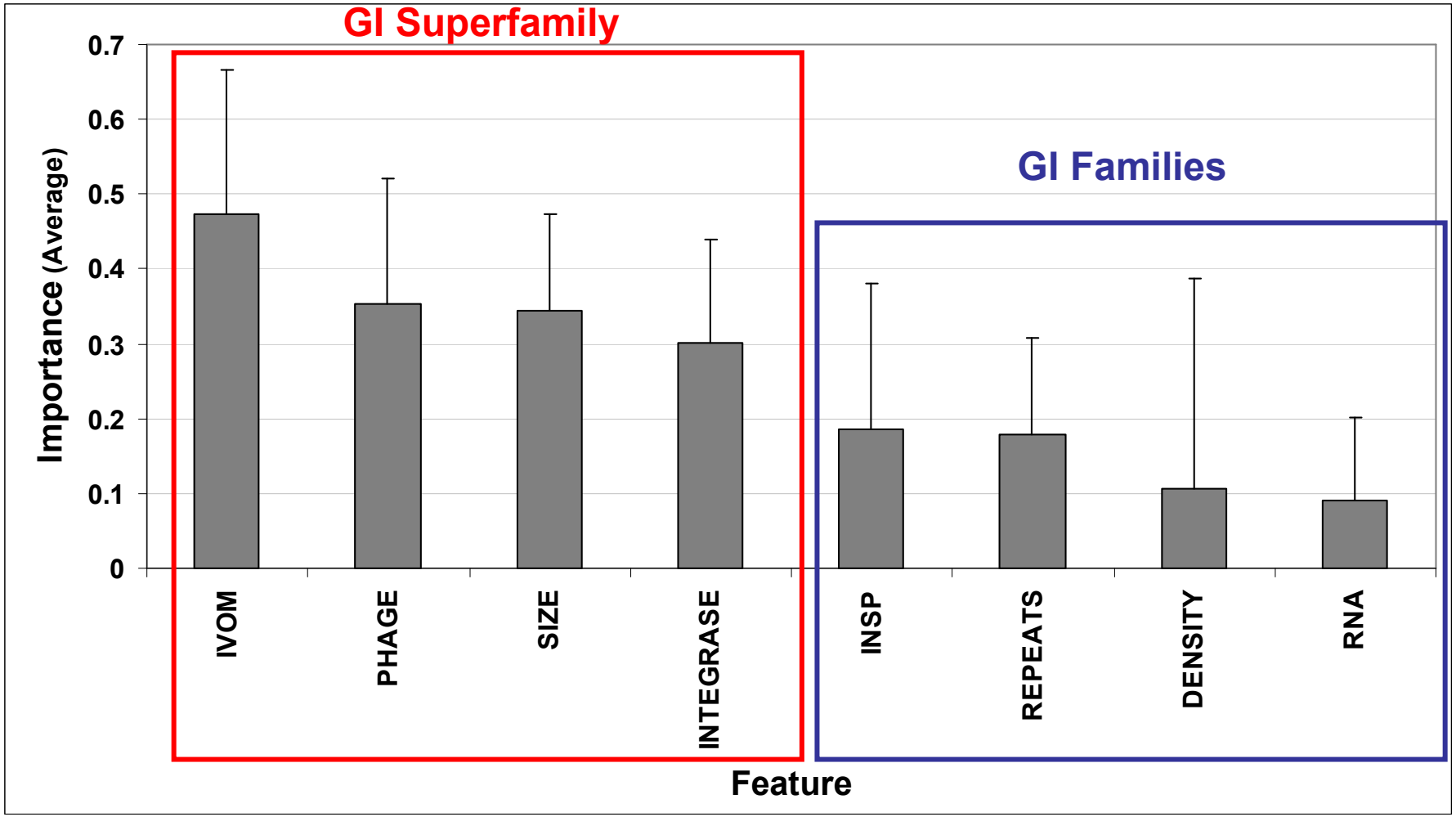
Feature contribution to the model (RVM weights)



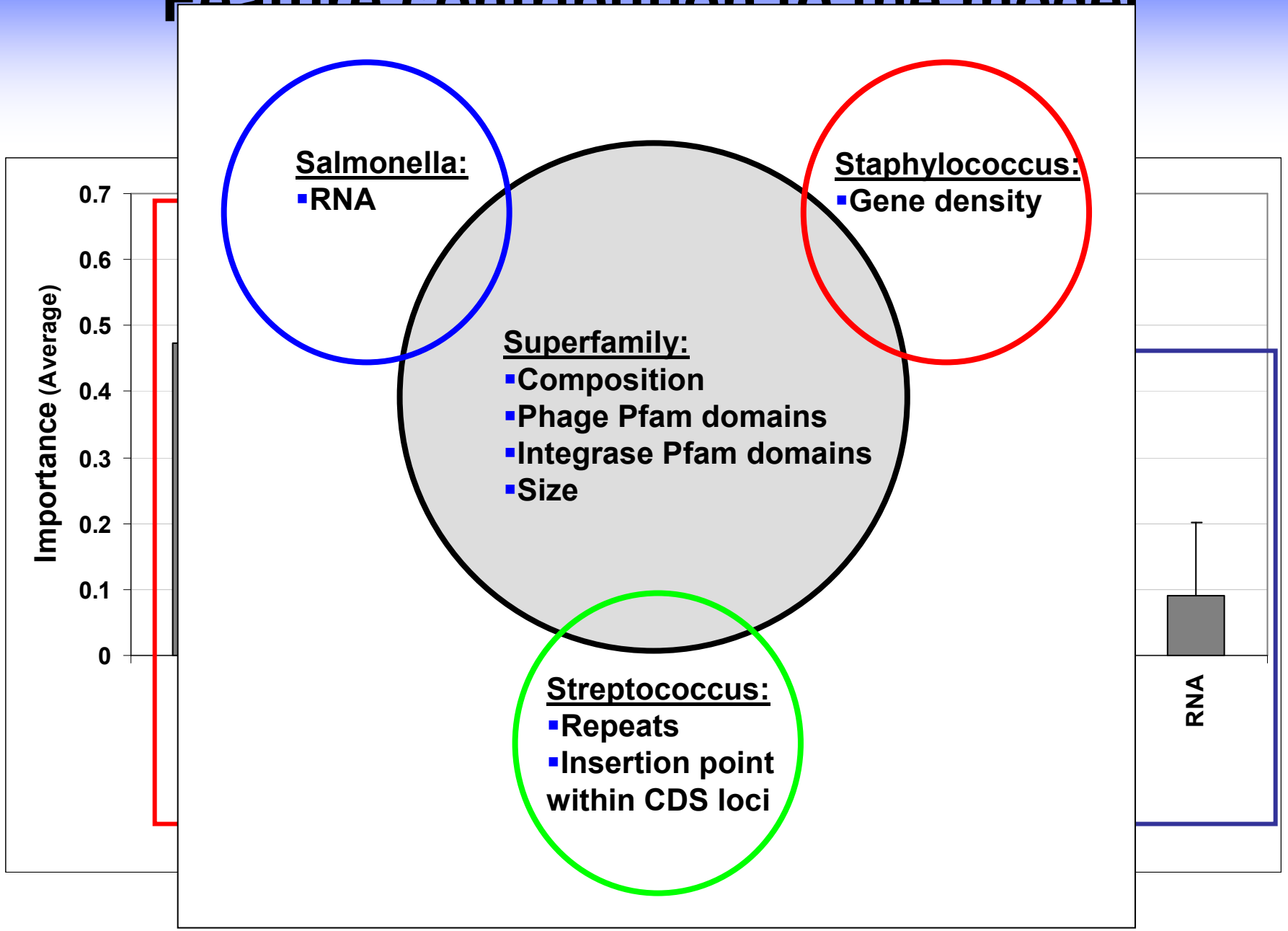
Feature contribution to the model (RVM weights)



Feature contribution to the model (RVM weights)



Feature contribution to the model



Error Margin

1. 10-20%
2. Structural intersection between true GIs and random regions
3. Some random regions were sampled close (e.g. tRNA locus) to true GIs
4. Phylogenetic resolution:
    A. Some GIs might not be true GIs if we increase the resolution
    B. Some random regions might be sampled over ancient GIs (not included in the true GI dataset)

Summary

✓Training on cross-genera dataset → GLMs converge over similar GI structure

✓GIs represent a superfamily of mobile elements with core and variable structural features rather than a well-defined family

✓When the taxa resolution increases, i.e. looking within genera/species boundaries, distinct families of GI structures emerge