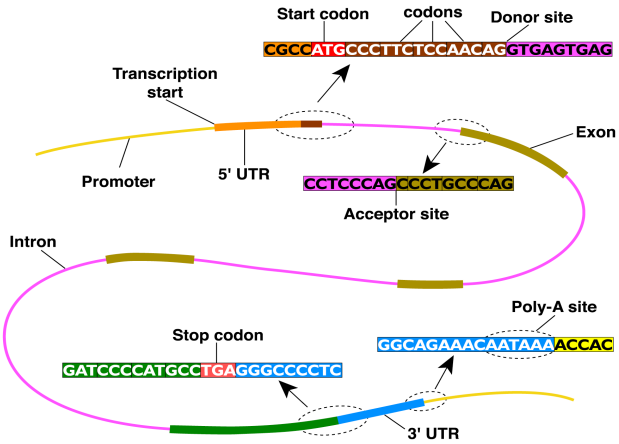
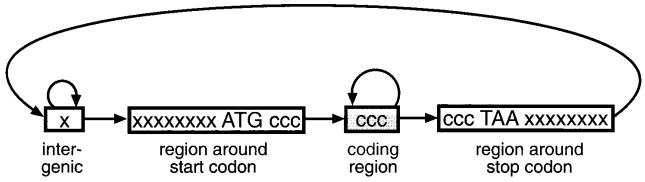
Gene finding with HMM?



Bacterial Gene finding as an HMM

- Nucleotides *{A,C,G,T}* are the observables
- Different states generates generate nucleotides at different frequencies

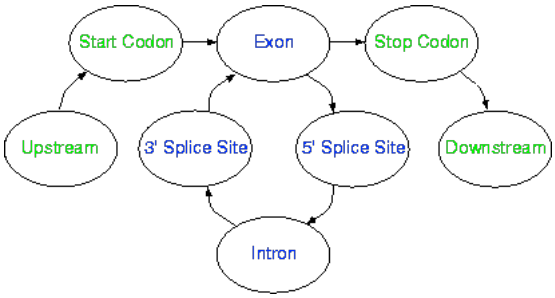
A simple HMM for unspliced genes:



AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

- The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

What about splicing?



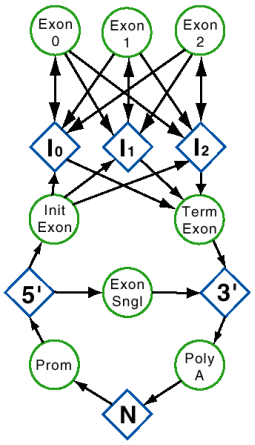
Note that transitions to the same state are left off simply to make the diagram simpler.

Genscan Example

- Developed by Chris Burge 1997
- One of the most accurate *ab initio* programs
- Uses explicit state duration HMM to model gene structure (different length distributions for exons)
- Different model parameters for regions with different GC content

Genscan (Burge and Karlin, 1998)

- Dramatic improvement over previous methods
- Generalised HMM
- Different parameter sets for different GC content regions (intron length distribution and exon stats)

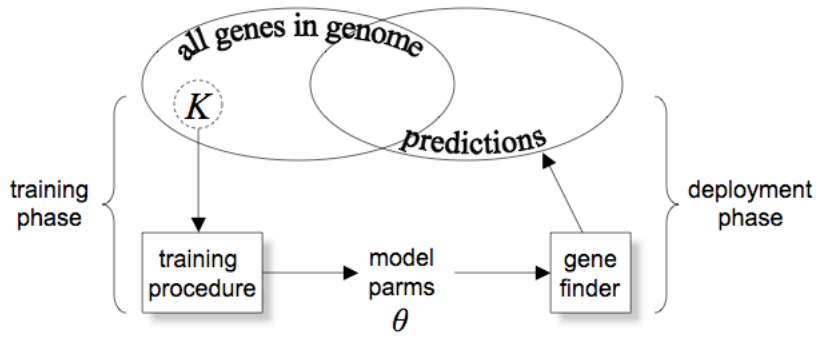


Desire a generalizable model.

- Trained on one dataset
- But performs well on new, never seen before data.

- So what constitutes good performance?

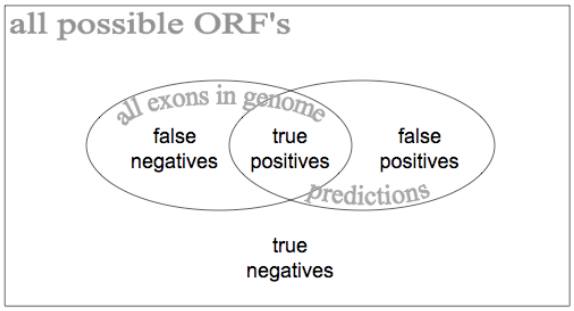
During *training* of a gene finder, only a subset *K* of an organism's gene set will be available for training. The gene finder will later be *deployed* for use in predicting the rest of the organism's genes. Alternatively, the training will use a closely related organism's gene set.



The way in which the *model parameters* are inferred during training can significantly affect the accuracy of the deployed program.

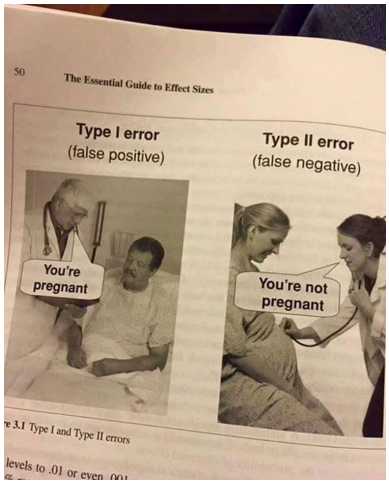
TP, FP, FN, TN

Gene predictions can be evaluated in terms of *true positives* (predicted features that are real), *true negatives* (non-predicted features that are not real), *false positives* (predicted features that are not real), and *false negatives* (real features that were not predicted:



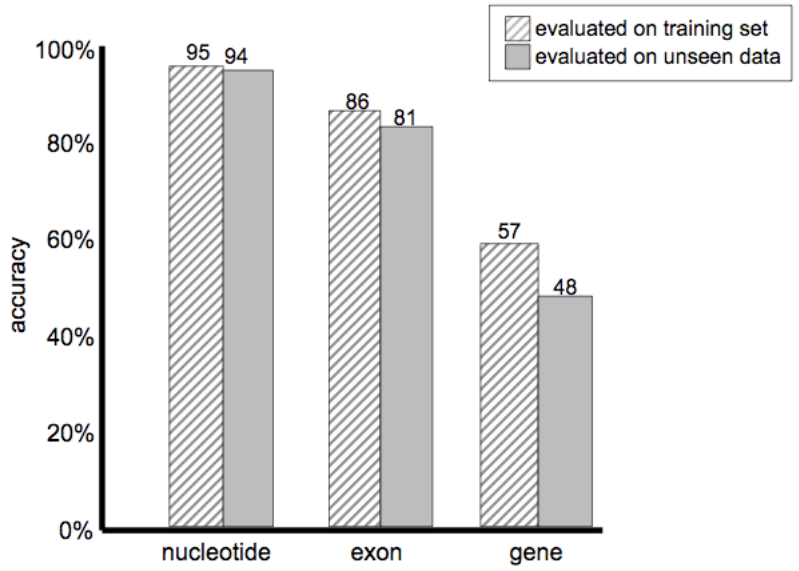
These definitions can be applied at the *whole-gene*, *whole-exon*, or *individual nucleotide* level to arrive at three sets of statistics.

Two distinct types of error …



Must first determine what predictions are actually correct.

Never Test on the Training Set!



		Gold Standard		
		Positive	Negative	
Test Outcome	Positive	True Positives	False Positives (Type I errors) (P-value)	Positive Predictive Value (PPV)
	Negative	False Negatives (Type II Errors)	True Negatives	Negative Predictive Value
		Sensitivity	Specificity	

Sensitivity = TP / (TP + FN)

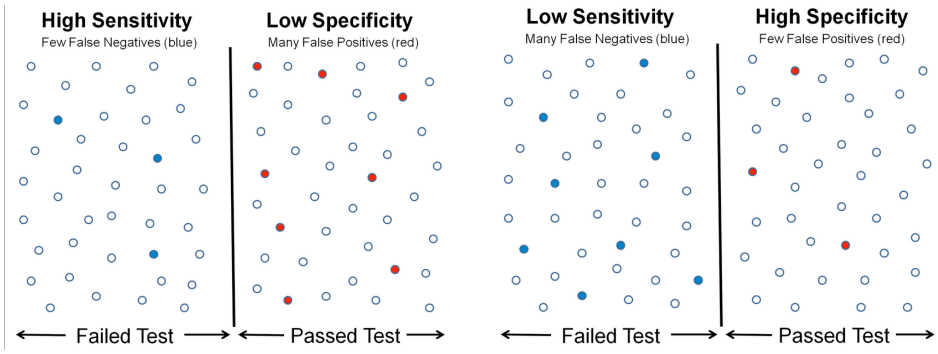
		Gold Standard		
		Positive	Negative	
Test Outcome	Positive	True Positives	False Positives (Type I errors) (P-value)	Positive Predictive Value (PPV)
	Negative	False Negatives (Type II Errors)	True Negatives	Negative Predictive Value
		Sensitivity	Specificity	

Sensitivity = TP / (TP + FN)

Specificity = TN / (TN + FP)

Sensitivity = TP / (TP + FN)

Specificity = TN / (TN + FP)



		Gold Standard		
		Positive	Negative	
Test Outcome	Positive	True Positives	False Positives (Type I errors) (P-value)	Positive Predictive Value (PPV)
	Negative	False Negatives (Type II Errors)	True Negatives	Negative Predictive Value
		Sensitivity	Specificity	

Primary Performance Metrics

Sensitivity = TP / (TP + FN)

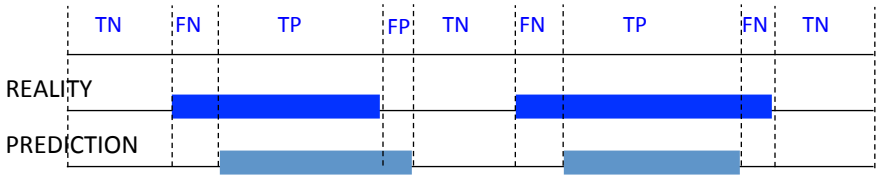
Specificity = TN / (TN + FP)

PPV = TP / (FP + TP)

Related Metrics

- False negative rate = FN / (TP + FN)
  = 1 − sensitivity
- False discovery rate = *FP* / (*FP* + *TP*)
  = 1 - PPV
- Accuracy = *(TP + TN) / ((TP+FN) + (FP+TN))*

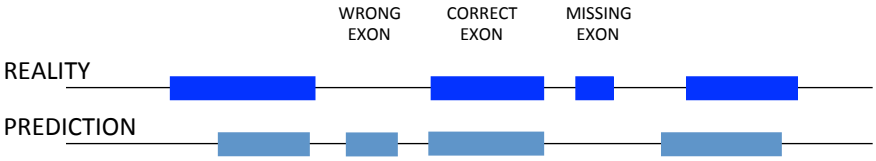
Nucleotide level accuracy



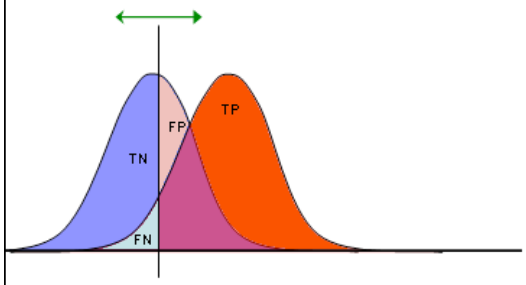
Sensitivity $Sn = \frac{TP}{TP + FN}$ $\frac{\text{number of correct nucleotides}}{\text{number of actual nucleotides}}$

PPV $Sp = \frac{TP}{TP + FP}$ $\frac{\text{number of correct nucleotides}}{\text{number of predicted nucleotides}}$

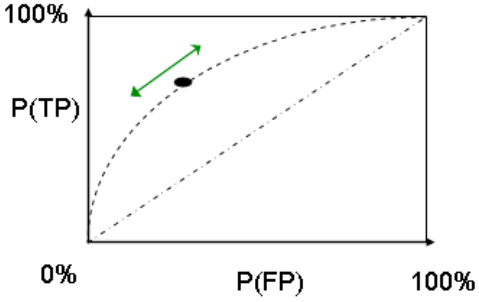
Exon level accuracy



$ESn = \frac{TE}{AE}$ $ESp = \frac{TE}{PE}$

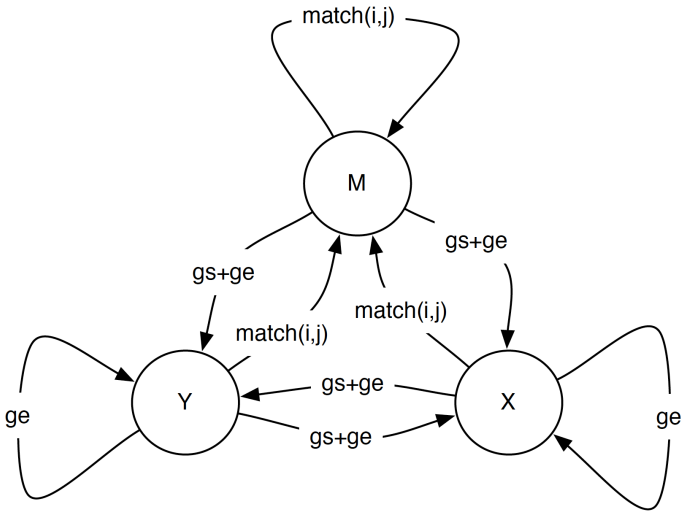


TP	FP
FN	TN
1	1



AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

Pairwise and other "fancy" Emissions: Affine Gap HMM



Dynamic Programming for the Affine Gap Penalty Case

- to do in $O(n^2)$ time, need 3 matrices instead of 1

$M(i,j)$	best score given that $x[i]$ is aligned to $y[j]$
$I_x(i,j)$	best score given that $x[i]$ is aligned to a gap
$I_y(i,j)$	best score given that $y[j]$ is aligned to a gap

Global Alignment DP for the Affine Gap Penalty Case

$$M(i,j) = \max\begin{cases} M(i-1,j-1) + S(x_i,y_j) \\ I_x(i-1,j-1) + S(x_i,y_j) \\ I_y(i-1,j-1) + S(x_i,y_j) \end{cases}$$

$$I_x(i,j) = \max\begin{cases} M(i-1,j) + g + s \\ I_x(i-1,j) + s \end{cases}$$

$$I_y(i,j) = \max\begin{cases} M(i,j-1) + g + s \\ I_y(i,j-1) + s \end{cases}$$

The M matrix

$$M(i,j) = \max\begin{cases} M(i-1,j-1) + S(x_i,y_j) \\ I_x(i-1,j-1) + S(x_i,y_j) \\ I_y(i-1,j-1) + S(x_i,y_j) \end{cases}$$

Any kind of alignment is allowed before the match.

The gap matricies (I_x and I_y)

If previous alignment ends in a match, this must be a new gap.

$$I_x(i,j) = \max\begin{cases} M(i-1,j) + g + s \\ I_x(i-1,j) + s \end{cases}$$

Otherwise we must be extending an existing gap.

Multiple Emissions: ChromHMM