CS2220 Introduction to Computational Biology

Lecture 7: Gene Finding by Computational Analysis

Limsoon Wong

Outline

- **Gene structure basics**
- **Gene finding overview**
- **GRAIL**
- **Indel & frame-shift in coding regions**

Gene Structure Basics

A brief refresher

Some slides here are “borrowed” from Ken Sung

Gene

- **A gene is a sequence of DNA that encodes a protein or an RNA molecule**
- **About 30,000 – 35,000 (protein-coding) genes in human genome**
- **For gene that encodes protein**
  - In Prokaryotic genome, one gene corresponds to one protein
  - In Eukaryotic genome, one gene can corresponds to more than one protein because of the process “alternative splicing”

Introns and Exons

- **Eukaryotic genes contain introns & exons**
  - Introns are seq that are ultimately spliced out of mRNA
  - Introns normally satisfy GT-AG rule, viz. begin w/ GT & end w/ AG
  - Each gene can have many introns & each intron can have thousands bases
- **Introns can be very long**
- **An extreme example is a gene associated with cystic fibrosis in human:**
  - Length of 24 introns ~1Mb
  - Length of exons ~1kb

Typical Eukaryotic Gene Structure

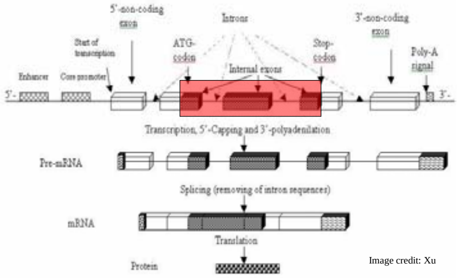


Image credit: Xu

- **Unlike eukaryotic genes, a prokaryotic gene typically consists of only one contiguous coding region**

Reading Frame

- **Each DNA segment has six possible reading frames**

Forward strand: ATGGCTTACGCTTGA

Reading frame #1	Reading frame #2	Reading frame #3
ATG	TGG	GGC
GCT	CTT	TTA
TAC	ACG	CGC
GCT	CTT	TTG
TGC	GA.	A..

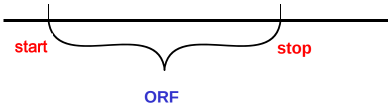
How do I get this reverse strand?

Reverse strand: TCAAGCGTAAGCCAT

Reading frame #4	Reading frame #5	Reading frame #6
TCA	CAA	AAG
AGC	GCG	CGT
GTA	TAA	AAG
AGC	GCC	CCA
CAT	AT.	T..

Open Reading Frame (ORF)

- **ORF is a segment of DNA with a start codon and an in-frame stop codon at the two ends and no in-frame stop codon in the middle**



- **Each ORF has a fixed reading frame**

NB: Other definitions are also used. Most impt aspect is that there is no stop codon in the middle.

Coding Region

- **Each coding region (exon or whole gene) has a fixed translation frame**
- **A coding region always sits inside an ORF of same reading frame**
- **All exons of a gene are on the same strand**
- **Neighboring exons of a gene could have different reading frames**

Frame Consistency

- **Neighboring exons of a gene should be frame-consistent**

Exercise: Define frame consistency mathematically

Overview of Gene Finding

Some slides here are "borrowed" from Mark Craven

What is Gene Finding?

- **Find all coding regions from a stretch of DNA sequence, and construct gene structures from the identified exons**
- **Can be decomposed into**
  - Find coding potential of a region in a frame
  - Find boundaries betw coding & non-coding regions

```
atgaacagacgcgatcttcttttacaagaaatgggcatttcccagtgggaattatatcgc
cccgaggtactgcaaggttcagtaggaattagtgtggcagagaatattcgccttatcact
gtttccgatgaaaatatcagtagctcgcctttgttggctgatgtgctgttaagccttaat
cttaaaaaagaaaattgtttatgtttgaattacgatcaaatccagcatatggaatgtaaa
tagcctattcgttattggttactatcagaaaatagcgaccaaattgaccgcactttgcca
ttttgcaagcaggctgagcaggtttatcgctcgccaagttggcagcaatttcaatctaat
catcaagccaaacgagcgttgtggcaataaattcagcagcctaa
```

Image credit: Xu

Approaches

- **Search-by-signal: find genes by identifying the sequence signals involved in gene expression**
- **Search-by-content: find genes by statistical properties that distinguish protein coding DNA from non-coding DNA**
- **Search-by-homology: find genes by homology (after translation) to proteins**

- **State-of-the-art systems for gene finding usually combine these strategies**

Relevant Signals for Search-by-Signals

- **Transcription initiation**
  - Promoter
- **Transcription termination**
  - Terminators
- **Translation initiation**
  - Ribosome binding sites
  - Initiation codons
- **Translation termination**
  - Stop codons
- **RNA processing**
  - Splice junction

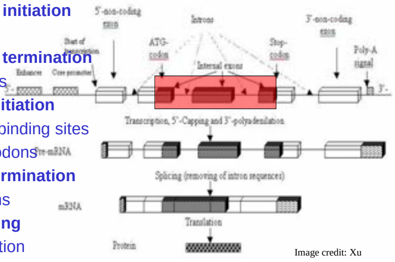


Image credit: Xu

How Search-by-Signal Works

- **There are 2 impt regions in a promoter seq**
  - –10 region, ~10bp before TSS
  - –35 region, ~35bp before TSS

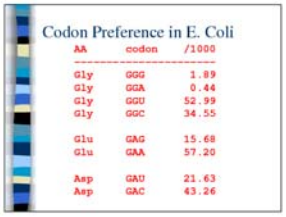
- **Consensus for –10 region in E. coli is TATAAT, but few promoters actually have this seq**

- **Recognize promoters by**
  - weight matrices
  - probabilistic models
  - neural networks, …

How Search-by-Content Works

- **Encoding a protein affects stats properties of a DNA seq**
  - some amino acids used more frequently
  - diff number of codons for diff amino acids
  - for given protein, usually one codon is used more frequently than others

⇒ **Estimate prob that a given region of seq was "caused by" its being a coding seq**



Codon Preference in E. Coli

AA	codon	/1000
Gly	GGG	1.89
Gly	GGA	0.44
Gly	GGU	52.99
Gly	GGC	34.55
Glu	GAG	15.68
Glu	GAA	57.20
Asp	GAU	21.63
Asp	GAC	43.26

Image credit: Craven

How Search-by-Homology Works

- **Translate DNA seq in all reading frames**

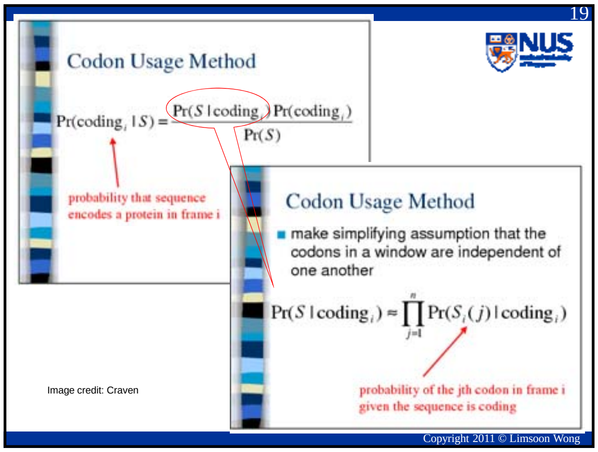
- **Search against protein db**

- **High-scoring matches suggest presence of homologous genes in DNA**

⇒ **You can use BLASTX for this**

Search-by-Content Example: Codon Usage Method

- **Staden & McLachlan, 1982**
- **Process a seq w/ "window" of length L**
- **Assume seq falls into one of 7 categories, viz.**
  - Coding in frame 0, frame 1, …, frame 5
  - Non-coding
- **Use Bayes' rule to determine prob of each category**
- **Assign seq to category w/ max prob**



Codon Usage Method

$$\Pr(\text{coding}_i \mid S) = \frac{\Pr(S \mid \text{coding}_i)\Pr(\text{coding}_i)}{\Pr(S)}$$

probability that sequence encodes a protein in frame i

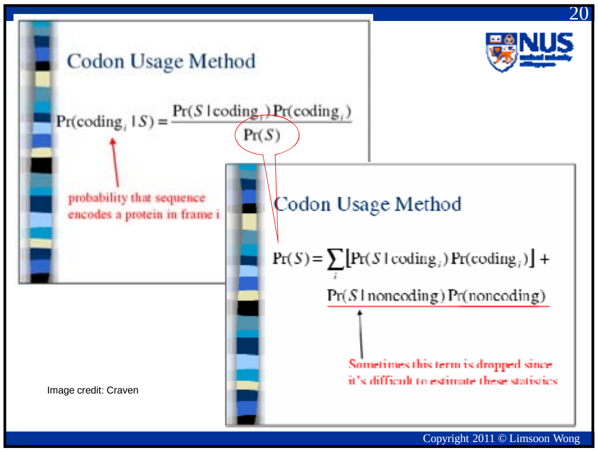
Codon Usage Method

- make simplifying assumption that the codons in a window are independent of one another

$$\Pr(S \mid \text{coding}_i) \approx \prod_{j=1}^{n} \Pr(S_i(j) \mid \text{coding}_i)$$

probability of the jth codon in frame i given the sequence is coding

Image credit: Craven



Codon Usage Method

$$\Pr(\text{coding}_i \mid S) = \frac{\Pr(S \mid \text{coding}_i)\Pr(\text{coding}_i)}{\Pr(S)}$$

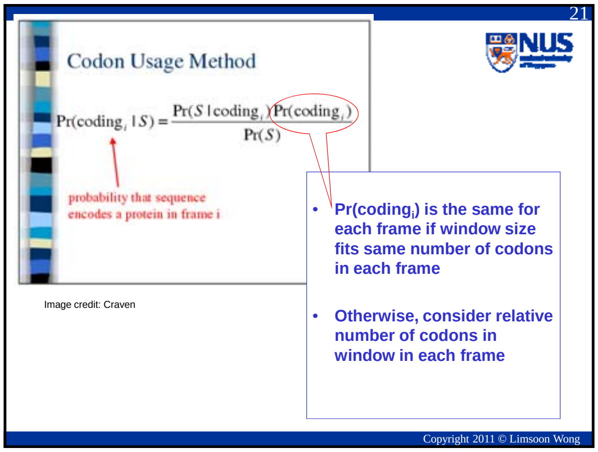
probability that sequence encodes a protein in frame i

Codon Usage Method

$$\Pr(S) = \sum_i \left[\Pr(S \mid \text{coding}_i)\Pr(\text{coding}_i)\right] + \underline{\Pr(S \mid \text{noncoding})\Pr(\text{noncoding})}$$

Sometimes this term is dropped since it's difficult to estimate these statistics

Image credit: Craven



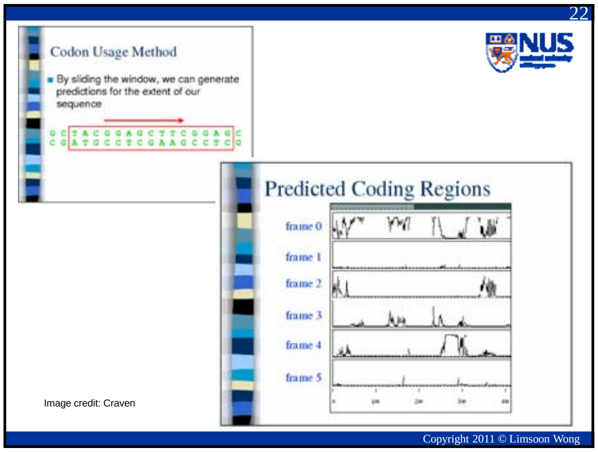
Codon Usage Method

$$\Pr(\text{coding}_i \mid S) = \frac{\Pr(S \mid \text{coding}_i)\Pr(\text{coding}_i)}{\Pr(S)}$$

probability that sequence encodes a protein in frame i

Image credit: Craven

- **Pr(coding$_i$) is the same for each frame if window size fits same number of codons in each frame**
- **Otherwise, consider relative number of codons in window in each frame**

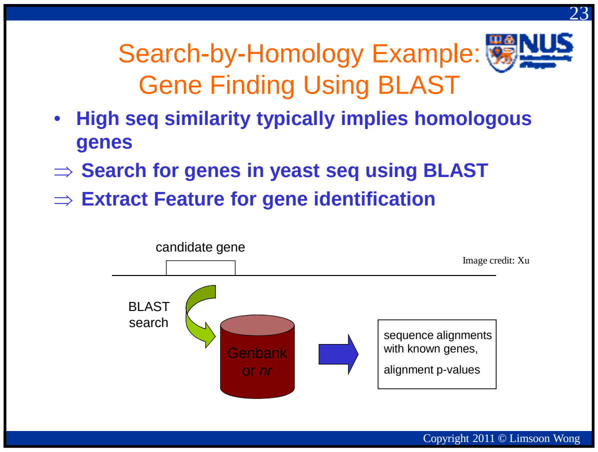


Codon Usage Method

- By sliding the window, we can generate predictions for the extent of our sequence

Predicted Coding Regions

Image credit: Craven



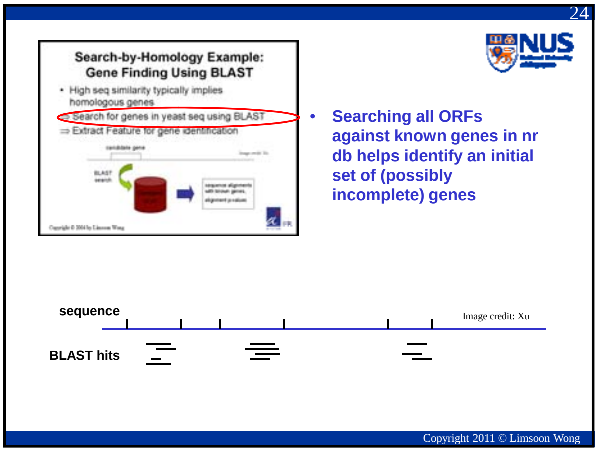
Search-by-Homology Example: Gene Finding Using BLAST

- **High seq similarity typically implies homologous genes**

⇒ Search for genes in yeast seq using BLAST

⇒ Extract Feature for gene identification

Image credit: Xu



Search-by-Homology Example: Gene Finding Using BLAST

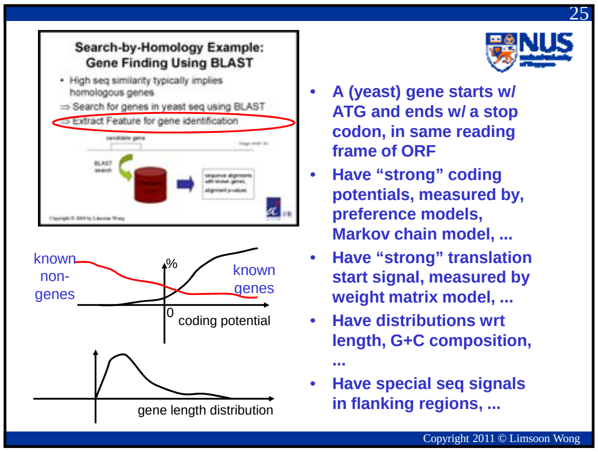
- High seq similarity typically implies homologous genes

⇒ Search for genes in yeast seq using BLAST

⇒ Extract Feature for gene identification

- **Searching all ORFs against known genes in nr db helps identify an initial set of (possibly incomplete) genes**

Image credit: Xu



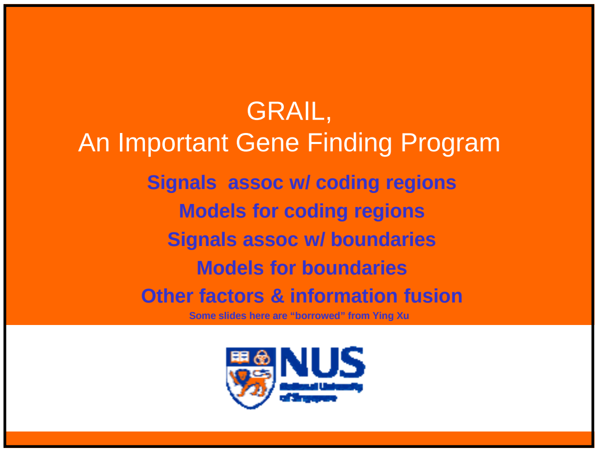
Search-by-Homology Example: Gene Finding Using BLAST

- High seq similarity typically implies homologous genes

⇒ Search for genes in yeast seq using BLAST

⇒ Extract Feature for gene identification

- **A (yeast) gene starts w/ ATG and ends w/ a stop codon, in same reading frame of ORF**
- **Have “strong” coding potentials, measured by, preference models, Markov chain model, ...**
- **Have “strong” translation start signal, measured by weight matrix model, ...**
- **Have distributions wrt length, G+C composition, ...**
- **Have special seq signals in flanking regions, ...**



GRAIL, An Important Gene Finding Program

Signals assoc w/ coding regions

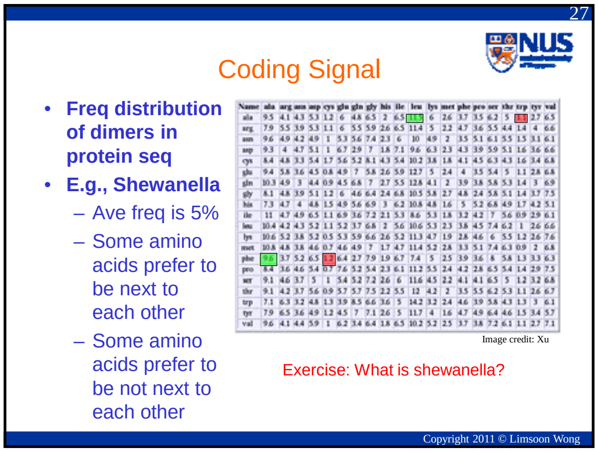
Models for coding regions

Signals assoc w/ boundaries

Models for boundaries

Other factors & information fusion

Some slides here are “borrowed” from Ying Xu



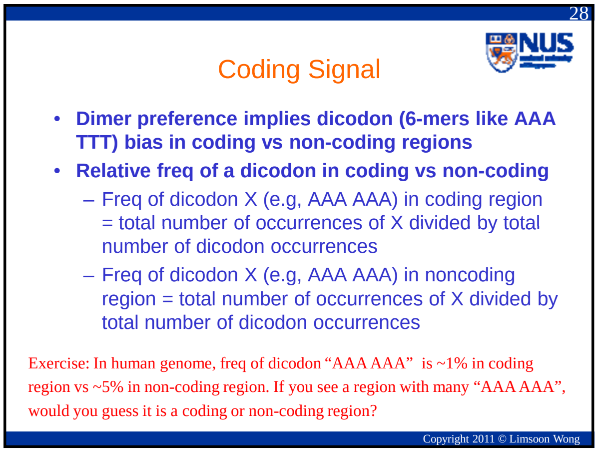
Coding Signal

- **Freq distribution of dimers in protein seq**
- **E.g., Shewanella**
  - Ave freq is 5%
  - Some amino acids prefer to be next to each other
  - Some amino acids prefer to be not next to each other

Name	ala	arg	asn	asp	cys	glu	gln	gly	his	ile	leu	lys	met	phe	pro	ser	thr	trp	tyr	val
ala	9.5	4.1	4.3	5.3	1.2	6	4.8	6.5	2	6.5	11.5	6	2.6	3.7	3.5	6.2	5	0.8	2.7	6.5
arg	7.9	5.5	3.9	5.3	1.1	6	5.5	5.9	2.6	6.5	11.4	5	2.2	4.7	3.6	5.5	4.4	1.4	4	6.6
asn	9.6	4.9	4.2	4.9	1	5.3	5.6	7.4	2.3	6	10	4.9	2	3.5	5.1	6.1	5.5	1.5	3.1	6.1
asp	9.3	4	4.7	5.1	1	6.7	2.9	7	1.8	7.1	9.6	6.3	2.3	4.3	3.9	5.9	5.1	1.6	3.6	6.6
cys	8.4	4.8	3.3	5.4	1.7	5.6	5.2	8.1	4.3	5.4	10.2	3.8	1.8	4.1	4.5	6.3	4.3	1.6	3.4	6.8
glu	9.4	5.8	3.6	4.5	0.8	4.9	7	5.8	2.6	5.9	12.7	5	2.4	4	3.5	5.4	5	1.1	2.8	6.8
gln	10.3	4.9	3	4.4	0.9	4.5	6.8	7	2.7	5.5	12.8	4.1	2	3.9	3.8	5.8	5.3	1.4	3	6.9
gly	8.1	4.8	3.9	5.1	1.2	6	4.6	6.4	2.4	6.8	10.5	5.8	2.7	4.8	2.4	5.8	5.1	1.4	3.7	7.5
his	7.3	4.7	4	4.8	1.5	4.9	5.6	6.9	3	6.2	10.8	4.8	1.6	5	5.2	6.8	4.9	1.7	4.2	5.1
ile	11	4.7	4.9	6.5	1.1	6.9	3.6	7.2	2.1	5.3	8.6	5.3	1.8	3.2	4.2	7	5.6	0.9	2.9	6.1
leu	10.4	4.2	4.3	5.2	1.1	5.2	3.7	6.8	2	5.6	10.6	5.3	2.3	3.8	4.5	7.4	6.2	1	2.6	6.6
lys	10.6	5.2	3.8	5.2	0.5	5.3	5.9	6.6	2.6	5.2	11.3	4.7	1.9	2.8	4.6	6	5.5	1.2	2.6	7.6
met	10.8	4.8	3.8	4.6	0.7	4.6	4.9	7	1.7	4.7	11.4	5.2	2.8	3.3	5.1	7.4	6.3	0.9	2	6.8
phe	9.6	3.7	5.2	6.5	1.2	6.4	2.7	7.9	1.9	6.7	7.4	5	2.5	3.9	3.6	8	5.8	1.3	3.3	6.3
pro	8.4	3.6	4.6	5.4	0.7	7.6	5.2	5.4	2.3	6.1	11.2	5.5	2.4	4.2	2.8	6.5	5.4	1.4	2.9	7.5
ser	9.1	4.6	3.7	5	1	5.4	5.2	7.2	2.6	6	11.6	4.5	2.2	4.1	4.1	6.5	5	1.2	3.2	6.8
thr	9.1	4.2	3.7	5.6	0.9	5.7	5.7	7.5	2.2	5.5	12	4.2	2	3.5	5.5	6.2	5.3	1.1	2.6	6.7
trp	7.1	6.3	3.2	4.8	1.3	3.9	8.5	6.6	3.6	5	14.2	3.2	2.4	4.6	3.9	5.8	4.3	1.3	3	6.1
tyr	7.9	6.5	3.6	4.9	1.2	4.5	7	7.1	2.6	5	11.7	4	1.6	4.7	4.9	6.4	4.6	1.5	3.4	5.7
val	9.6	4.1	4.4	5.9	1	6.2	3.4	6.4	1.8	6.5	10.2	5.2	2.5	3.7	3.8	7.2	6.1	1.1	2.7	7.1

Image credit: Xu

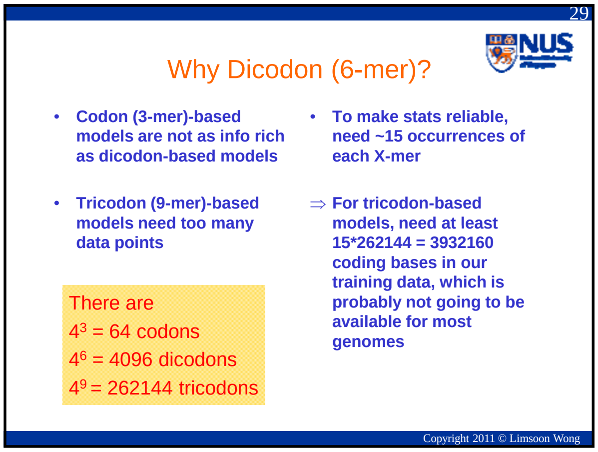
Exercise: What is shewanella?



Coding Signal

- **Dimer preference implies dicodon (6-mers like AAA TTT) bias in coding vs non-coding regions**
- **Relative freq of a dicodon in coding vs non-coding**
  - Freq of dicodon X (e.g, AAA AAA) in coding region = total number of occurrences of X divided by total number of dicodon occurrences
  - Freq of dicodon X (e.g, AAA AAA) in noncoding region = total number of occurrences of X divided by total number of dicodon occurrences

Exercise: In human genome, freq of dicodon “AAA AAA” is ~1% in coding region vs ~5% in non-coding region. If you see a region with many “AAA AAA”, would you guess it is a coding or non-coding region?



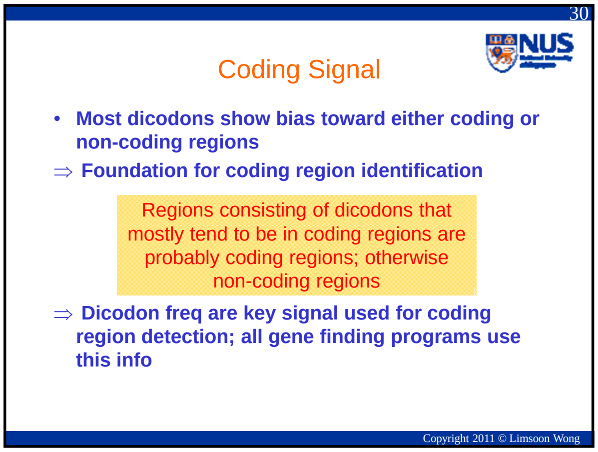
Why Dicodon (6-mer)?

- **Codon (3-mer)-based models are not as info rich as dicodon-based models**
- **Tricodon (9-mer)-based models need too many data points**

There are
$4^3 = 64$ codons
$4^6 = 4096$ dicodons
$4^9 = 262144$ tricodons

- **To make stats reliable, need ~15 occurrences of each X-mer**

⇒ **For tricodon-based models, need at least 15*262144 = 3932160 coding bases in our training data, which is probably not going to be available for most genomes**



Coding Signal

- **Most dicodons show bias toward either coding or non-coding regions**

⇒ **Foundation for coding region identification**

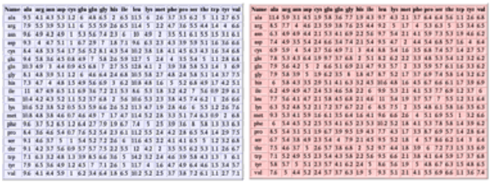
Regions consisting of dicodons that mostly tend to be in coding regions are probably coding regions; otherwise non-coding regions

⇒ **Dicodon freq are key signal used for coding region detection; all gene finding programs use this info**

Coding Signal

- **Dicodon freq in coding vs non-coding are genome-dependent**

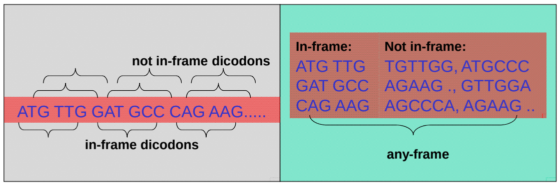
Image credit: Xu



Shewanella Bovine

Coding Signal

- **In-frame vs any-frame dicodons**
- **In-frame dicodon freq provide a more sensitive measure than any-frame dicodon freq**



not in-frame dicodons

ATG TTG GAT GCC CAG AAG.....

in-frame dicodons

In-frame: ATG TTG GAT GCC CAG AAG

Not in-frame: TGTTGG, ATGCCC AGAAG ., GTTGGA AGCCCA, AGAAG ..

any-frame

Dicodon Preference Model

- **The preference value P(X) of a dicodon X is defined as**

  P(X) = log FC(X)/FN(X)

  where

  FC(X) is freq of X in coding regions

  FN(X) is freq of X in non-coding regions

Dicodon Preference Model's Properties

- **P(X) = 0 if X has same freq in coding and non-coding regions**
- **P(X) > 0 if X has higher freq in coding than in non-coding region; the larger the diff, the more positive the score is**
- **P(X) < 0 if X has higher freq in non-coding than in coding region; the larger the diff, the more negative the score is**

Dicodon Preference Model Example

- **Suppose AAA ATT, AAA GAC, AAA TAG have the following freq:**

  FC(AAA ATT) = 1.4%
  FN(AAA ATT) = 5.2%

  FC(AAA GAC) = 1.9%
  FN(AAA GAC) = 4.8%

  FC(AAA TAG) = 0.0%
  FN(AAA TAG) = 6.3%

- **Then**

  P(AAA ATT) = –0.57
  P(AAA GAC) = –0.40
  P(AAA TAG) = –∞, treating STOP codons differently

⇒ **A region consisting of only these dicodons is probably a non-coding region**

Frame-Insensitive Coding Region Preference Model

- **A frame-insensitive coding preference $S_{is}(R)$ of a region R can be defined as**

$$S_{is}(R) = \Sigma_{X \text{ is a dicodon in } R} P(X)$$

- **R is predicted as coding region if $S_{is}(R) > 0$**

NB. This model is not commonly used

In-Frame Dicodon Preference Model

- **The in-frame + i preference value $P_i(X)$ of a dicodon X is defined as**

  $P_i(X) = \log FC_i(X)/FN(X)$

  where

  $FC_i(X)$ is freq of X in coding regions at in-frame + i positions

  $FN(X)$ is freq of X in non-coding regions

In-Frame Coding Region Preference Model

- **The in-frame + i preference $S_i(R)$ of a region R can be defined as**

  $S_i(R) = \Sigma_{X \text{ is a dicodon at in-frame + i position in } R}\ P_i(X)$

- **R is predicted as coding if $\Sigma_{i=0,1,2}\ S_i(R)/|R| > 0$**

NB. This coding preference model is commonly used

Coding Region Prediction: An Example Procedure

- **Calculate all ORFs of a DNA segment**
- **For each ORF**
  - Slide thru ORF w/ increment of 10bp
  - Calculate in-frame coding region preference score, in same frame as ORF, within window of 60bp
  - Assign score to center of window
- **E.g., forward strand in a particular frame...**

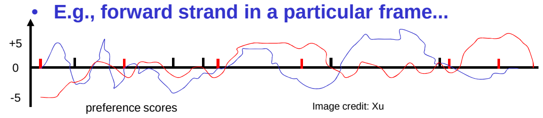


Image credit: Xu

Problem with Coding Region Boundaries

- **Making the call: coding or non-coding and where the boundaries are**

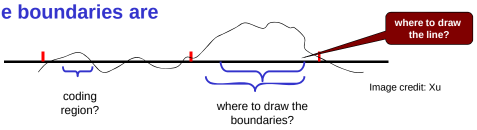


Image credit: Xu

⇒ Need training set with known coding and non-coding regions to select threshold that includes as many known coding regions as possible, and at the same time excludes as many known non-coding regions as possible

---

Types of Coding Region Boundaries

- **Knowing boundaries of coding regions helps identify them more accurately**
- **Possible boundaries of an exon**

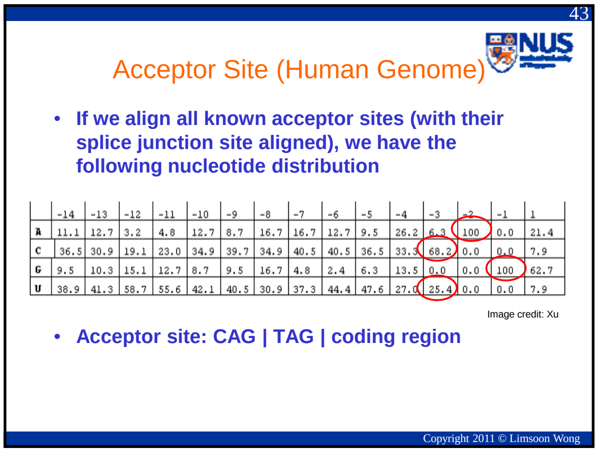
Image credit: Xu

- **Splice junctions:**
  - Donor site: coding region | GT
  - Acceptor site: CAG | TAG | coding region
- **Translation start**
  - in-frame ATG

What do you expect at translation stop?

Signals for Coding Region Boundaries

- **Splice junction sites and translation starts have certain distribution profiles**
- **For example, ...**



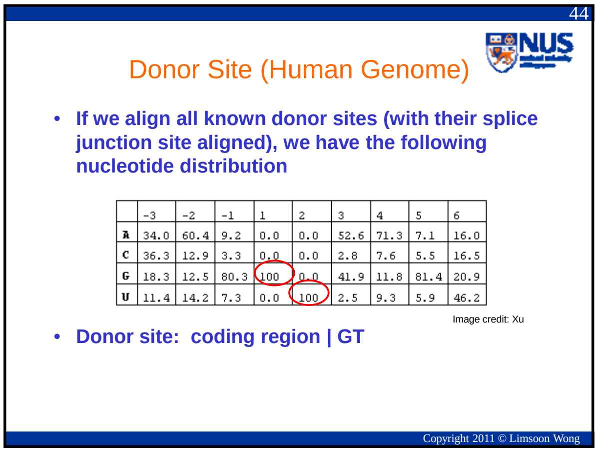
Acceptor Site (Human Genome)

- **If we align all known acceptor sites (with their splice junction site aligned), we have the following nucleotide distribution**

	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	1
A	11.1	12.7	3.2	4.8	12.7	8.7	16.7	16.7	12.7	9.5	26.2	6.3	100	0.0	21.4
C	36.5	30.9	19.1	23.0	34.9	39.7	34.9	40.5	40.5	36.5	33.3	68.2	0.0	0.0	7.9
G	9.5	10.3	15.1	12.7	8.7	9.5	16.7	4.8	2.4	6.3	13.5	0.0	0.0	100	62.7
U	38.9	41.3	58.7	55.6	42.1	40.5	30.9	37.3	44.4	47.6	27.0	25.4	0.0	0.0	7.9

Image credit: Xu

- **Acceptor site: CAG | TAG | coding region**



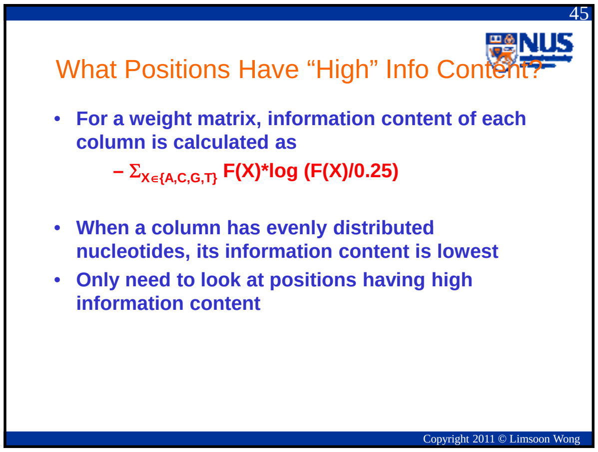
Donor Site (Human Genome)

- **If we align all known donor sites (with their splice junction site aligned), we have the following nucleotide distribution**

	-3	-2	-1	1	2	3	4	5	6
A	34.0	60.4	9.2	0.0	0.0	52.6	71.3	7.1	16.0
C	36.3	12.9	3.3	0.0	0.0	2.8	7.6	5.5	16.5
G	18.3	12.5	80.3	100	0.0	41.9	11.8	81.4	20.9
U	11.4	14.2	7.3	0.0	100	2.5	9.3	5.9	46.2

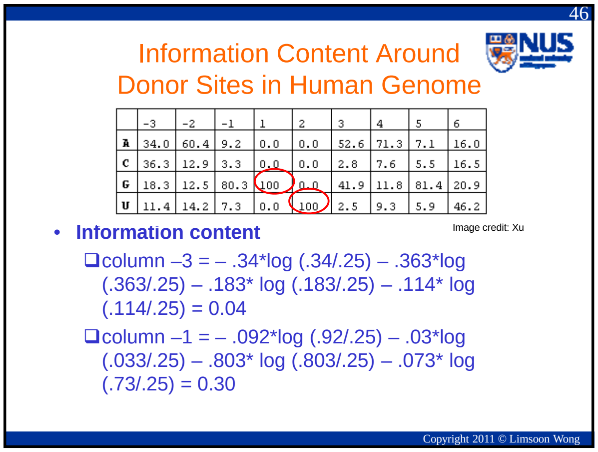
Image credit: Xu

- **Donor site: coding region | GT**



What Positions Have “High” Info Content?

- **For a weight matrix, information content of each column is calculated as**
  - $-\Sigma_{X \in \{A,C,G,T\}} F(X)*\log(F(X)/0.25)$
- **When a column has evenly distributed nucleotides, its information content is lowest**
- **Only need to look at positions having high information content**

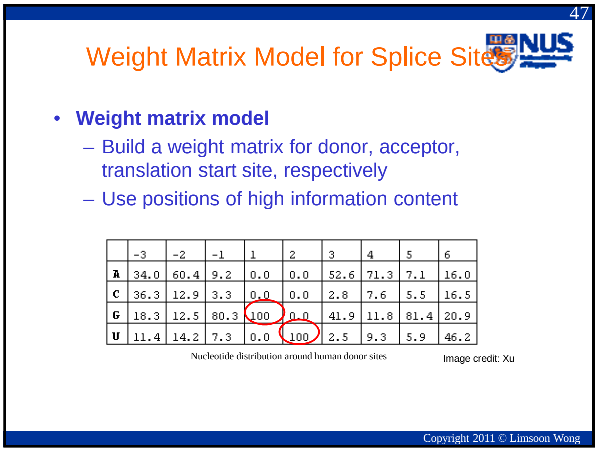


Information Content Around Donor Sites in Human Genome

	-3	-2	-1	1	2	3	4	5	6
A	34.0	60.4	9.2	0.0	0.0	52.6	71.3	7.1	16.0
C	36.3	12.9	3.3	0.0	0.0	2.8	7.6	5.5	16.5
G	18.3	12.5	80.3	100	0.0	41.9	11.8	81.4	20.9
U	11.4	14.2	7.3	0.0	100	2.5	9.3	5.9	46.2

Image credit: Xu

- **Information content**
  - ❑column –3 = – .34*log (.34/.25) – .363*log (.363/.25) – .183* log (.183/.25) – .114* log (.114/.25) = 0.04
  - ❑column –1 = – .092*log (.92/.25) – .03*log (.033/.25) – .803* log (.803/.25) – .073* log (.73/.25) = 0.30



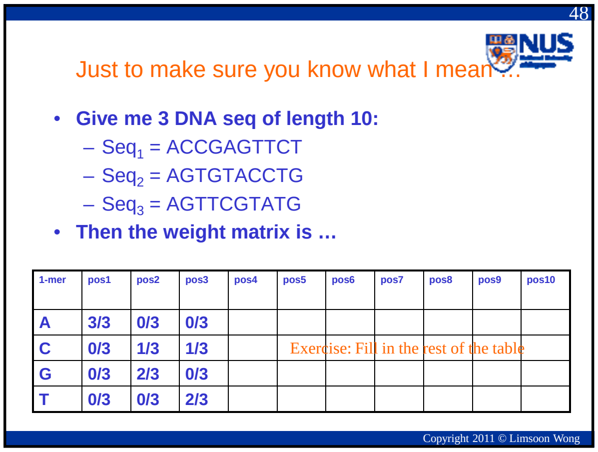
Weight Matrix Model for Splice Sites

- **Weight matrix model**
  - Build a weight matrix for donor, acceptor, translation start site, respectively
  - Use positions of high information content

	-3	-2	-1	1	2	3	4	5	6
A	34.0	60.4	9.2	0.0	0.0	52.6	71.3	7.1	16.0
C	36.3	12.9	3.3	0.0	0.0	2.8	7.6	5.5	16.5
G	18.3	12.5	80.3	100	0.0	41.9	11.8	81.4	20.9
U	11.4	14.2	7.3	0.0	100	2.5	9.3	5.9	46.2

Nucleotide distribution around human donor sites

Image credit: Xu



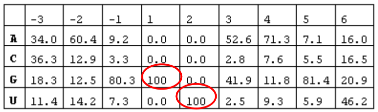
Just to make sure you know what I mean …

- **Give me 3 DNA seq of length 10:**
  - Seq_1 = ACCGAGTTCT
  - Seq_2 = AGTGTACCTG
  - Seq_3 = AGTTCGTATG
- **Then the weight matrix is …**

1-mer	pos1	pos2	pos3	pos4	pos5	pos6	pos7	pos8	pos9	pos10
A	3/3	0/3	0/3							
C	0/3	1/3	1/3							
G	0/3	2/3	0/3							
T	0/3	0/3	2/3							

Exercise: Fill in the rest of the table

Splice Site Prediction: A Procedure



	-3	-2	-1	1	2	3	4	5	6
A	34.0	60.4	9.2	0.0	0.0	52.6	71.3	7.1	16.0
C	36.3	12.9	3.3	0.0	0.0	2.8	7.6	5.5	16.5
G	18.3	12.5	80.3	100	0.0	41.9	11.8	81.4	20.9
U	11.4	14.2	7.3	0.0	100	2.5	9.3	5.9	46.2

Nucleotide distribution around human donor sites

Image credit: Xu

- **Add up freq of corr letter in corr positions:**

  AAGGTAAGT: .34 + .60 + .80 +1.0 + 1.0 + .52 + .71 + .81 + .46 = 6.24

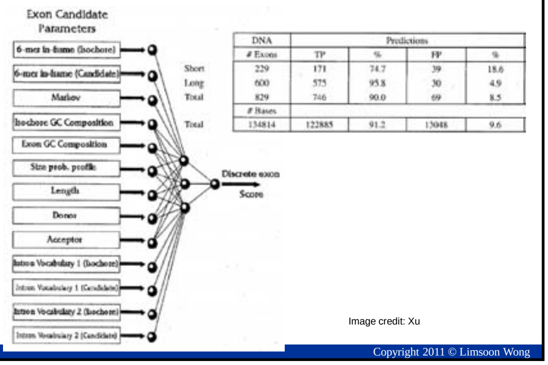
  TGTGTCTCA: .11 + .12 + .03 +1.0 + 1.0 + .02 + .07 + .05 + .16 = 2.56

- **Make prediction on splice site based on some threshold**

Other Factors Considered by GRAIL

- **G+C composition affects dicodon distributions**
- **Length of exons follows certain distribution**
- **Other signals associated with coding regions**
  - periodicity
  - structure information
  - .....
- **Pseudo genes**
- **........**

Info Fusion by ANN in GRAIL



	DNA	Predictions			
	# Exons	TP	%	FP	%
Short	229	171	74.7	39	18.6
Long	600	575	95.8	30	4.9
Total	829	746	90.0	69	8.5
	# Bases				
Total	134814	122885	91.2	13048	9.6

Image credit: Xu

Remaining Challenges in GRAIL

- **Initial exon**
- **Final exon**
- **Indels & frame shifts**

Indel & Frame-Shift in Coding Regions

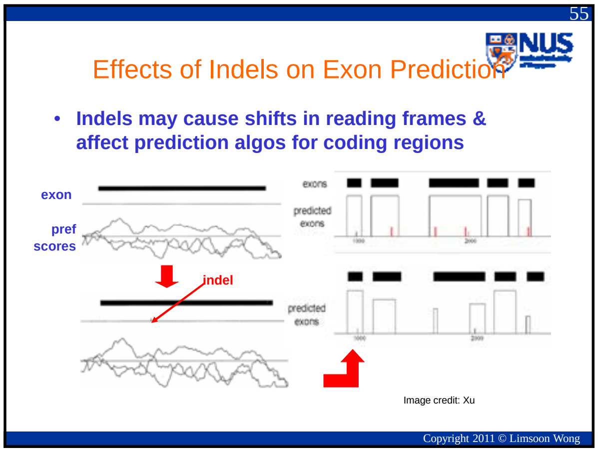
Problem definition
Indel & frameshift identification
Indel correction
An iterative strategy

Some slides here are "borrowed" from Ying Xu

Indels in Coding Regions

- **Indel = insertion or deletion in coding region**
- **Indels are usually caused by seq errors**

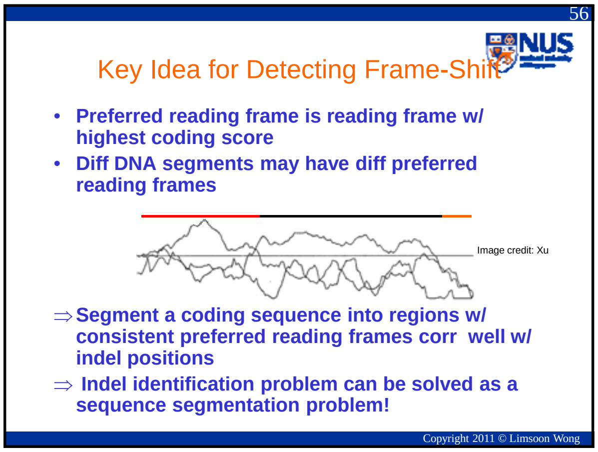
ATG GAT CCA CAT ..... ➡ **ATG GAT CA CAT .....**
ATG GAT CTCA CAT .....



Effects of Indels on Exon Prediction

- **Indels may cause shifts in reading frames & affect prediction algos for coding regions**

Image credit: Xu



Key Idea for Detecting Frame-Shift

- **Preferred reading frame is reading frame w/ highest coding score**
- **Diff DNA segments may have diff preferred reading frames**

Image credit: Xu

⇒ Segment a coding sequence into regions w/ consistent preferred reading frames corr well w/ indel positions

⇒ Indel identification problem can be solved as a sequence segmentation problem!

Frame-Shift Detection by Seq Segmentation

- **Partition seq into segs so that**
  - Chosen frames of adjacent segs are diff
  - Each segment has >30 bps to avoid small fluctuations
  - Sum of coding scores in the chosen frames over all segments is maximized

---

Frame-Shift Detection: A Simplified Treatment

- **Given DNA sequence $a_1 \ldots a_n$**
- **Define key quantities**

$C(i, r)$ = max score on $a_1 \ldots a_i$, w/ the last segment in frame r

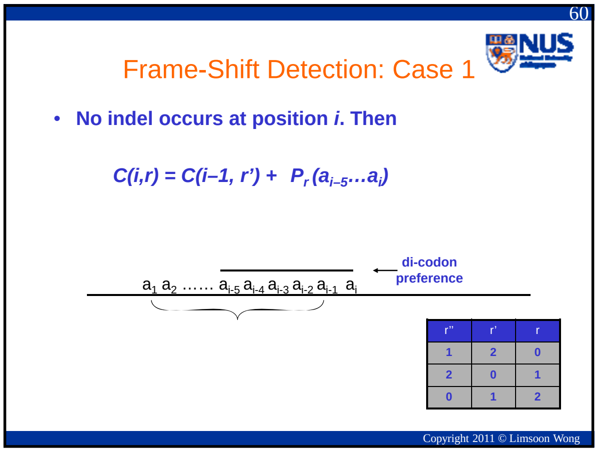
- **Then**

$\max_{r \in \{0, 1, 2\}} C(n, r)$ is optimal solution

Frame-Shift Detection: *C(i,r)*

- **To calculate $C(i,r)$, there are 3 possible cases for each position i:**
  - Case 1: no indel occurred at position i
  - Case 2: a_i is an inserted base
  - Case 3: a base has been deleted in front of a_i

⇒ $C(i, r)$ = max { Case 1, Case 2, Case 3 }



Frame-Shift Detection: Case 1

- **No indel occurs at position i. Then**

$$C(i,r) = C(i-1, r') + P_r(a_{i-5} \ldots a_i)$$

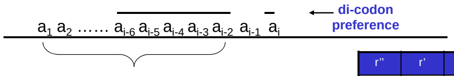
$$\underbrace{a_1\ a_2\ \ldots\ldots\ \overline{a_{i-5}\ a_{i-4}\ a_{i-3}\ a_{i-2}\ a_{i-1}}}\ a_i$$

r''	r'	r
1	2	0
2	0	1
0	1	2

Frame-Shift Detection: Case 2

- **a_{i-1} is an inserted base. Then**

$$C(i,r) = C(i-2, r') + P_r(a_{i-6}\ldots a_{i-2}a_i)$$

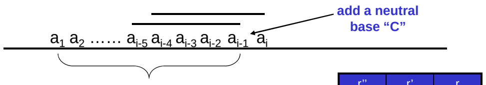


r''	r'	r
1	2	0
2	0	1
0	1	2

Frame-Shift Detection: Case 3

- **A base has been deleted in front of a_i. Then**

$$C(i, r) = C(i-1, r'') + P_{r'}(a_{i-5}\ldots a_{i-1}C) + P_r(a_{i-4}\ldots a_{i-1}Ca_i)$$



Exercise: why is "C" is best choice for the purpose above?

r''	r'	r
1	2	0
2	0	1
0	1	2

Frame-Shift Detection: Initiation

- **Initial conditions,**

  $C(k, r) = -\infty,\ k < 6$

  $C(6, r) = P_r(a_1 \ldots a_6)$

- **This is a dynamic programming (DP) algorithm; the equations are DP recurrences**

Exercise: How to modified the recurrence so that each fragment is at least 30bp?

Frame-Shift Detection: Determining Indel Positions

- **Calculation of $\max_{r \in \{0, 1, 2\}} C(i, r)$ gives an optimal segmentation of a DNA sequence**
- **Tracing back the transition points---viz. case 2 & case 3---gives the segmentation results**

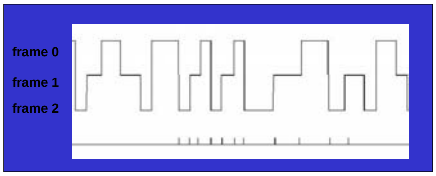


Image credit: Xu

Frame-Shift Detection: Determine Coding Regions

- **For given H_1 and H_2 (e.g., = 0.25 for noncoding and 0.75 for coding), partition a DNA seq into segs so that each seg has >30 bases & coding values of each seg are consistently closer to one of H_1 or H_2 than the other**

Image credit: Xu

Frame-Shift Detection: Finally…

- **Overlay "preferred reading-frame segs" & "coding segs" gives coding region predictions regions w/ indels**

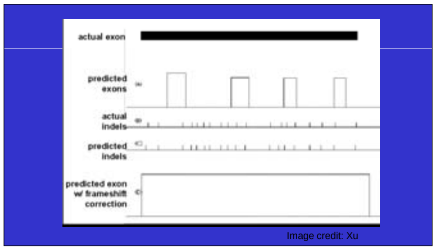
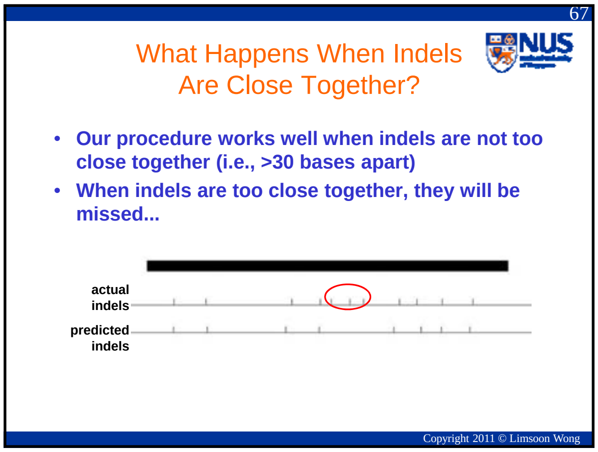
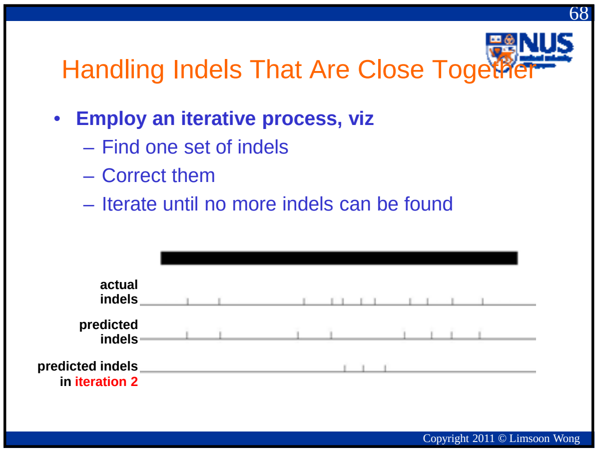


Image credit: Xu



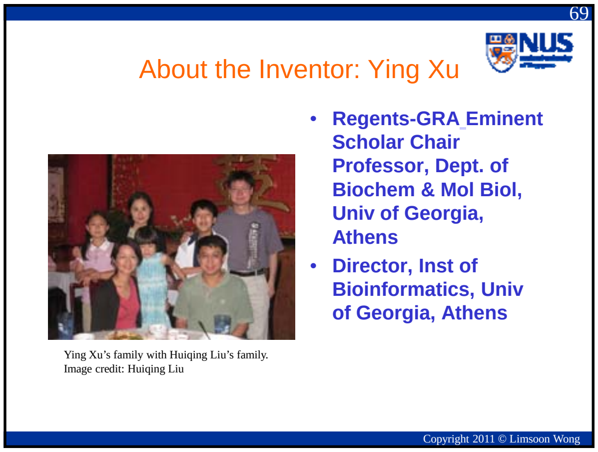
What Happens When Indels Are Close Together?

- **Our procedure works well when indels are not too close together (i.e., >30 bases apart)**
- **When indels are too close together, they will be missed...**



Handling Indels That Are Close Together

- **Employ an iterative process, viz**
  - Find one set of indels
  - Correct them
  - Iterate until no more indels can be found

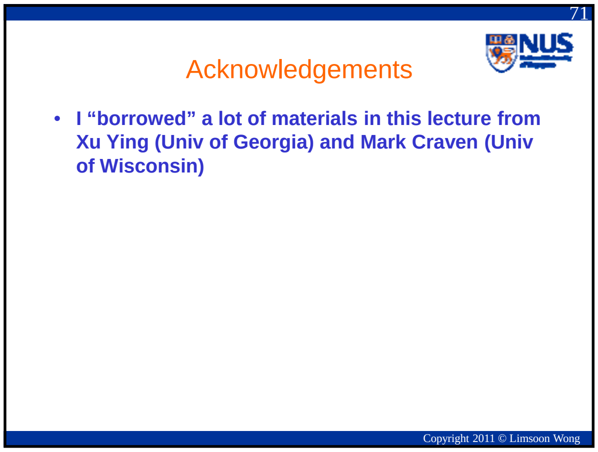


About the Inventor: Ying Xu

- **Regents-GRA Eminent Scholar Chair Professor, Dept. of Biochem & Mol Biol, Univ of Georgia, Athens**
- **Director, Inst of Bioinformatics, Univ of Georgia, Athens**

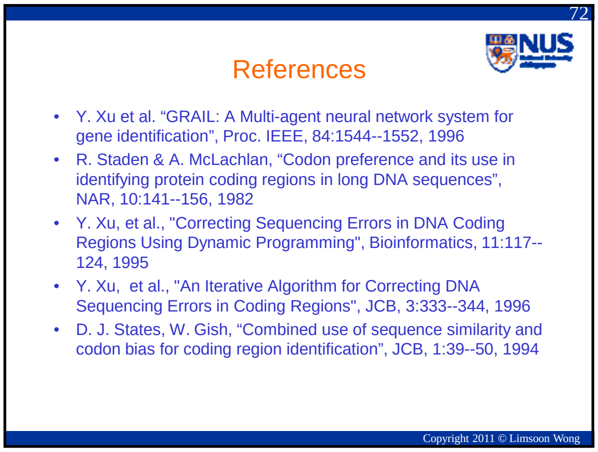
Ying Xu's family with Huiqing Liu's family.
Image credit: Huiqing Liu

Any Question?



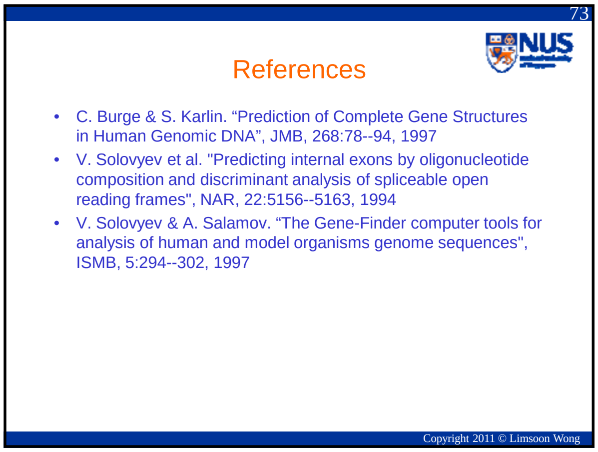
Acknowledgements

- **I "borrowed" a lot of materials in this lecture from Xu Ying (Univ of Georgia) and Mark Craven (Univ of Wisconsin)**



References

- Y. Xu et al. "GRAIL: A Multi-agent neural network system for gene identification", Proc. IEEE, 84:1544--1552, 1996
- R. Staden & A. McLachlan, "Codon preference and its use in identifying protein coding regions in long DNA sequences", NAR, 10:141--156, 1982
- Y. Xu, et al., "Correcting Sequencing Errors in DNA Coding Regions Using Dynamic Programming", Bioinformatics, 11:117--124, 1995
- Y. Xu, et al., "An Iterative Algorithm for Correcting DNA Sequencing Errors in Coding Regions", JCB, 3:333--344, 1996
- D. J. States, W. Gish, "Combined use of sequence similarity and codon bias for coding region identification", JCB, 1:39--50, 1994



References

- C. Burge & S. Karlin. “Prediction of Complete Gene Structures in Human Genomic DNA”, JMB, 268:78--94, 1997
- V. Solovyev et al. "Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames", NAR, 22:5156--5163, 1994
- V. Solovyev & A. Salamov. “The Gene-Finder computer tools for analysis of human and model organisms genome sequences", ISMB, 5:294--302, 1997