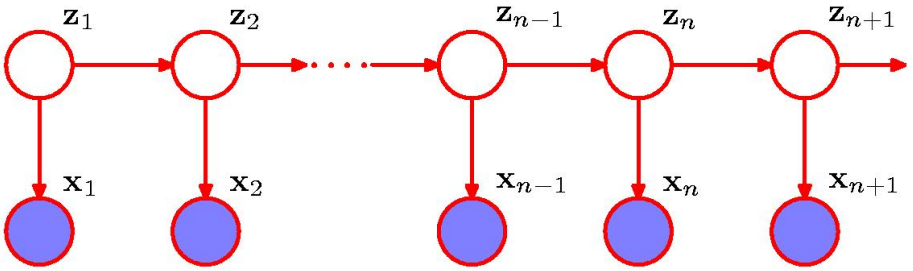
Hidden Markov Models

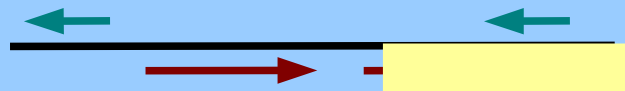
Hand In 3 - Gene finding



C: coding left-to-right

Even more biology

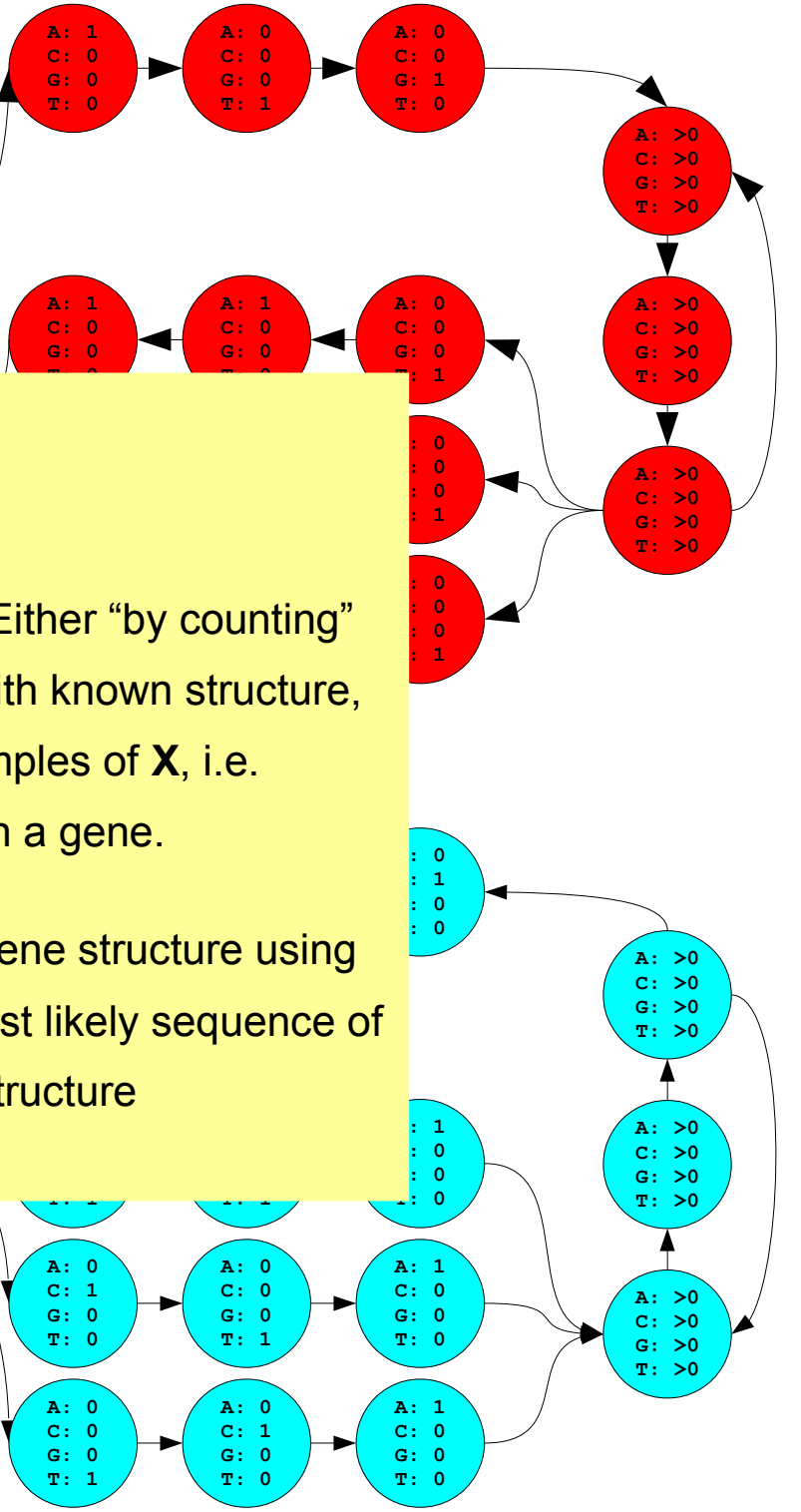
There can be genes in both directions



Gene finding

- Select initial model structure
- Select model parameters by training. Either “by counting” from examples of (**X**,**Z**)'s, i.e. genes with known structure, or by EM- or Viterbi-training from examples of **X**, i.e. sequences which are known to contain a gene.
- Given a new sequence **X**, predict its gene structure using the Viterbi algorithm for finding the most likely sequence of underlying latent states, i.e. its gene structure

N: No



$\pi_N = 1$

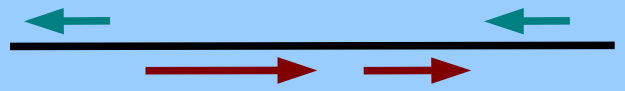
$\pi_C = 0$

R: coding right-to-left

Selecting an initial model

Even more biology

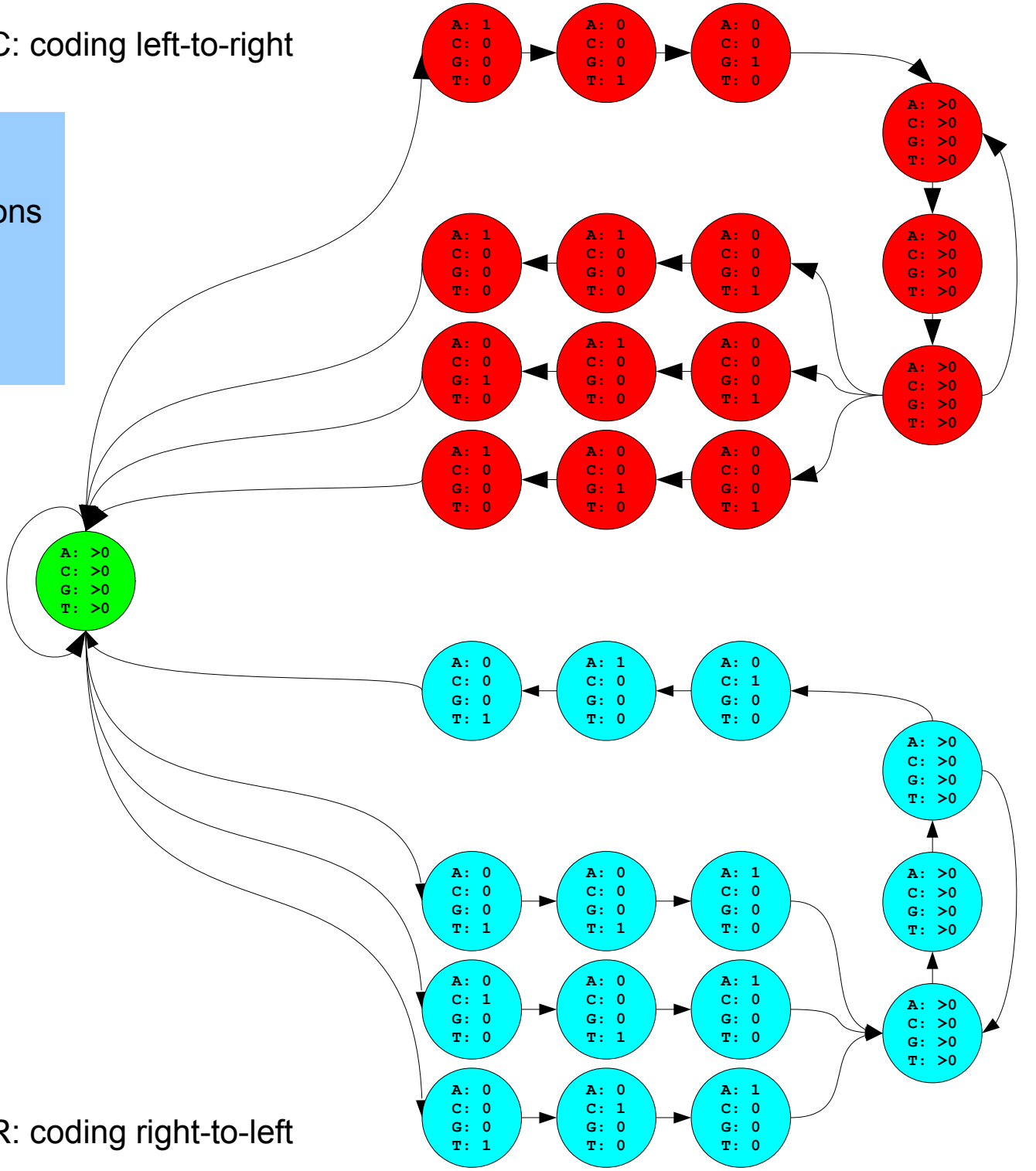
There can be genes in both directions



C: coding left-to-right

N: Non-coding

R: coding right-to-left



$\pi_N = 1$

$\pi_C = 0$

Analysis of the training data

```
Length of genome1: 1852441 (1852441)
Length of genome2: 2211485 (2211485)
Length of genome3: 2499279 (2499279)
Length of genome4: 1796846 (1796846)
Length of genome5: 2685015 (2685015)
Length of genome6: 2127839 (2127839)
Length of genome7: 2742531 (2742531)
Length of genome8: 2046115 (2046115)
Length of genome9: 2388435 (2388435)
Length of genome10: 1570485 (1570485)
Length of genome11: 2096309 (2096309)
```

We observe

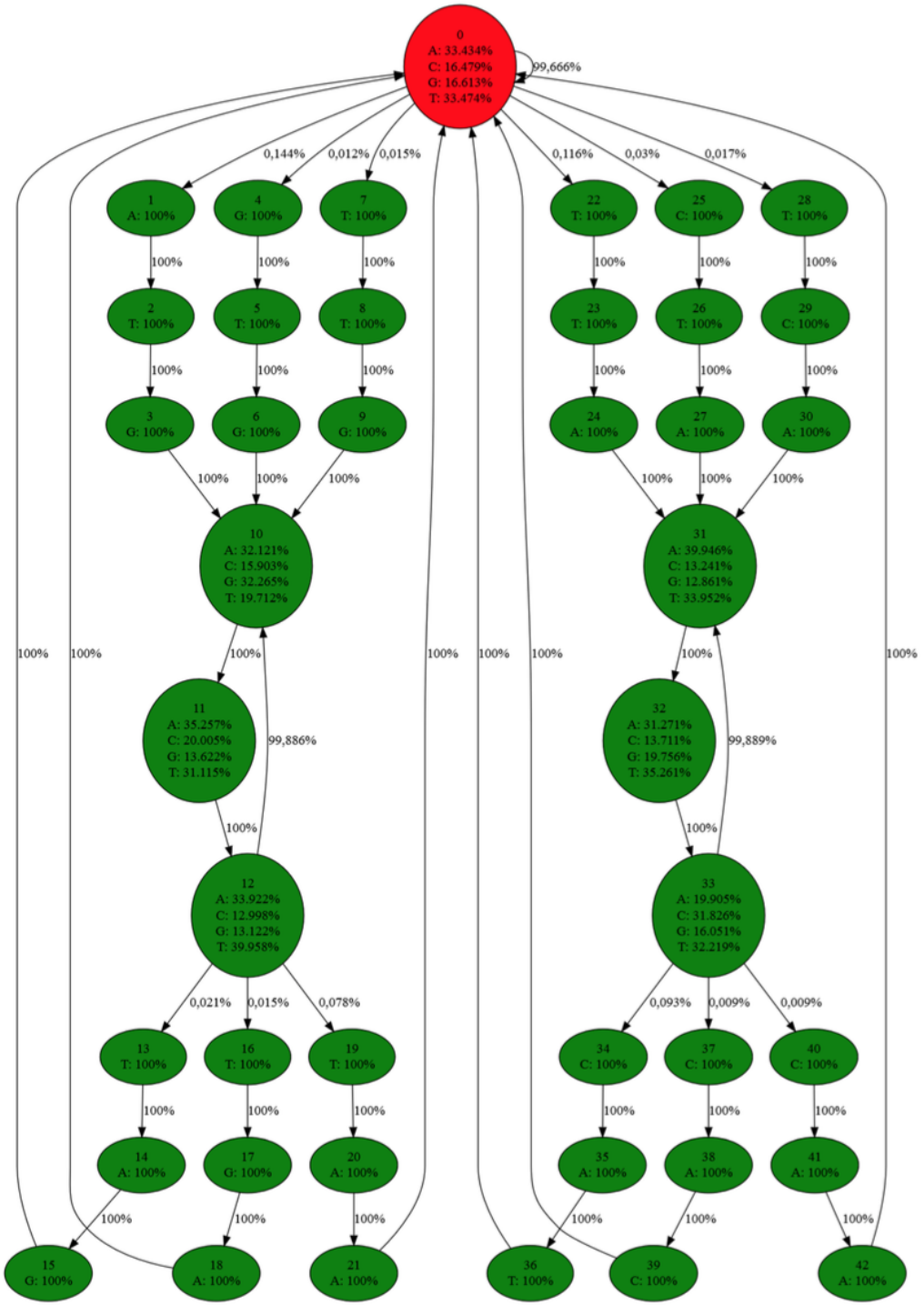
3 typical start-codons

3 stop-codons

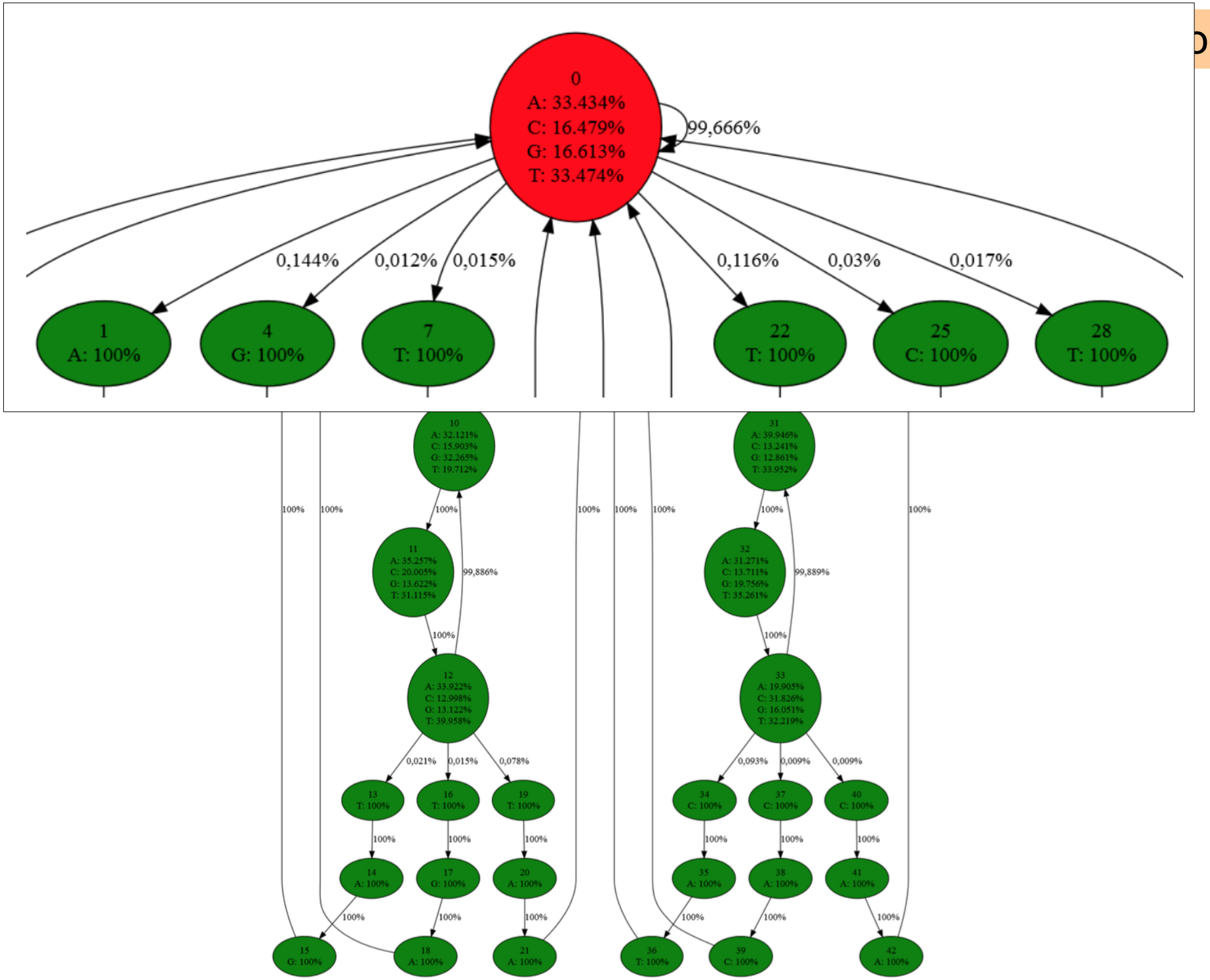
We may ignore rare start and stop codons, i.e. a gene that starts (or ends) with a ignored start (or stop) codon does not correspond to a path in your model.

```
Start-codon in normal genes:
ATG [8423, 'NCCC']
ATC [3, 'NCCC']
ATA [1, 'RCCC']
GTG [713, 'NCCC']
ATT [3, 'NCCC']
CTG [2, 'NCCC']
GTT [1, 'NCCC']
CTC [1, 'NCCC']
TTA [1, 'NCCC']
TTG [1020, 'NCCC']

Stop-codon in normal genes:
TAG [1949, 'CCCN']
TGA [1531, 'CCCN']
TAA [6686, 'CCCN']

Reversed stop-codon in reversed genes:
TTA (reverse-complement: TAA) [6596, 'NRRR']
CTA (reverse-complement: TAG) [2014, 'NRRR']
TCA (reverse-complement: TGA) [1148, 'NRRR']

Reversed start-codon in reversed genes:
TAT (reverse-complement: ATA) [2, 'RRRN']
ATG (reverse-complement: CAT) [1, 'RRRN']
GAT (reverse-complement: ATC) [1, 'RRRN']
CAT (reverse-complement: ATG) [8077, 'RRRN']
AAT (reverse-complement: ATT) [4, 'RRRN']
TAC (reverse-complement: GTA) [1, 'RRRN']
CAC (reverse-complement: GTG) [715, 'RRRN']
CAA (reverse-complement: TTG) [953, 'RRRN']
CAG (reverse-complement: CTG) [4, 'RRRN']
```

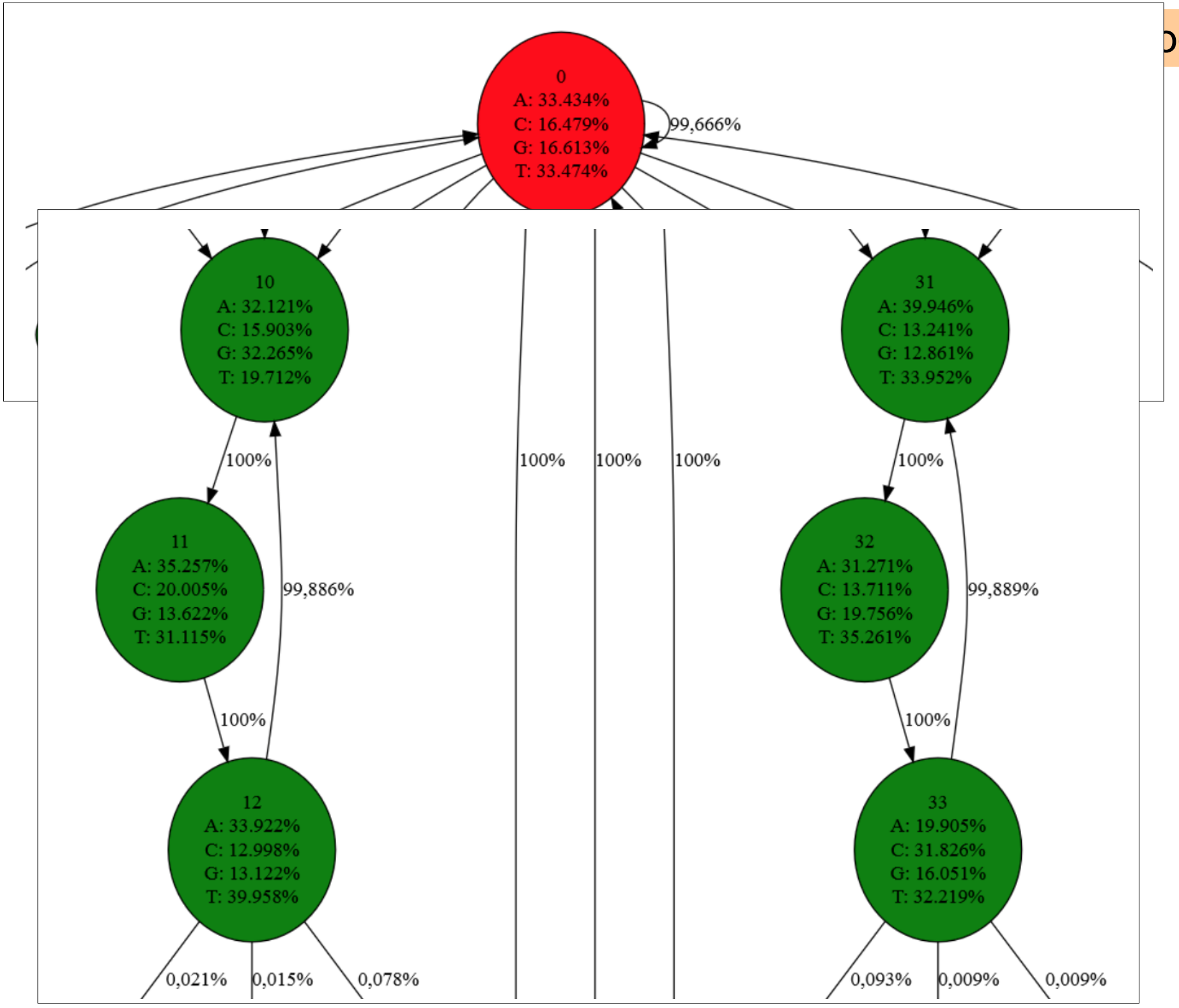
A typical model



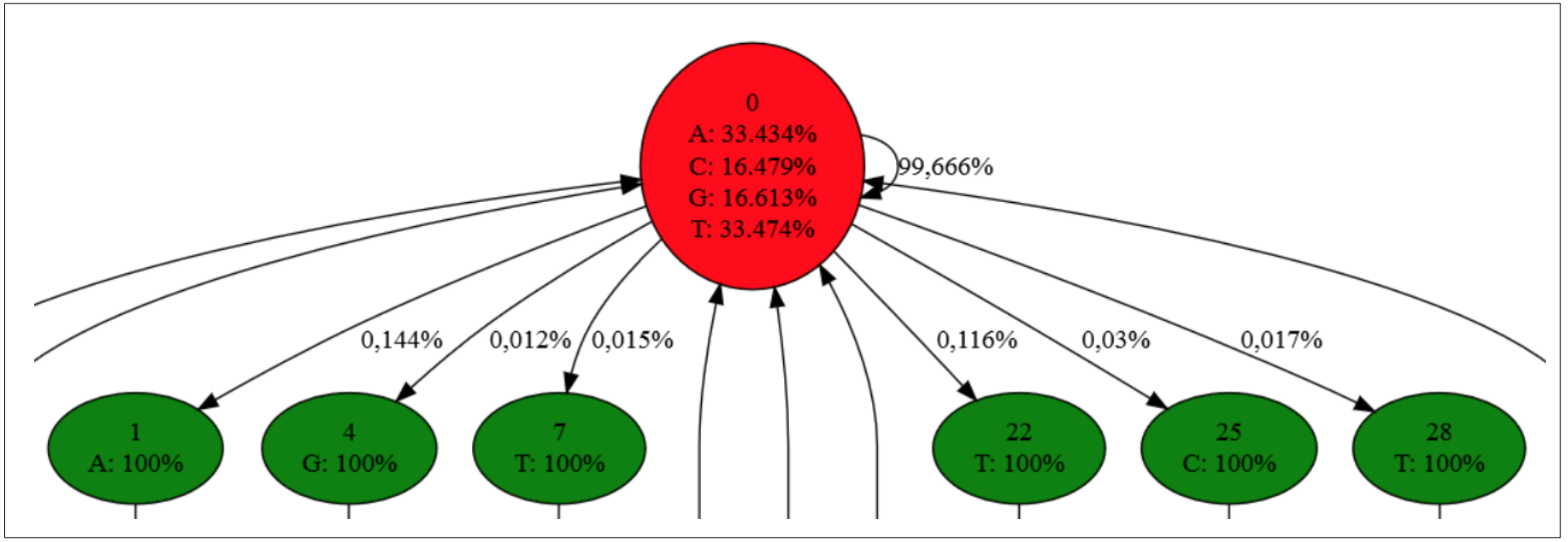
odel


0
A: 33.434%
C: 16.479%
G: 16.613%
T: 33.474%
99,666%
0,144%
0,012%
0,015%
0,116%
0,03%
0,017%
1
A: 100%
4
G: 100%
7
T: 100%
22
T: 100%
25
C: 100%
28
T: 100%
10
A: 32.121%
C: 15.903%
G: 32.265%
T: 19.712%
11
A: 35.257%
C: 20.005%
G: 13.622%
T: 31.115%
99,886%
12
A: 33.922%
C: 12.998%
G: 13.122%
T: 39.958%
0,021%
0,015%
0,078%
13
T: 100%
16
T: 100%
19
T: 100%
14
A: 100%
17
G: 100%
20
A: 100%
15
G: 100%
18
A: 100%
21
A: 100%
31
A: 39.946%
C: 13.241%
G: 12.861%
T: 33.952%
32
A: 31.271%
C: 13.711%
G: 19.756%
T: 35.261%
99,889%
33
A: 19.905%
C: 31.826%
G: 16.051%
T: 32.219%
0,093%
0,009%
0,009%
34
C: 100%
37
C: 100%
40
C: 100%
35
A: 100%
38
A: 100%
41
A: 100%
36
T: 100%
39
C: 100%
42
A: 100%
100%

odel


0
A: 33.434%
C: 16.479%
G: 16.613%
T: 33.474%
99,666%
10
A: 32.121%
C: 15.903%
G: 32.265%
T: 19.712%
100%
11
A: 35.257%
C: 20.005%
G: 13.622%
T: 31.115%
99,886%
100%
12
A: 33.922%
C: 12.998%
G: 13.122%
T: 39.958%
0,021%
0,015%
0,078%
100%
100%
100%
31
A: 39.946%
C: 13.241%
G: 12.861%
T: 33.952%
100%
32
A: 31.271%
C: 13.711%
G: 19.756%
T: 35.261%
99,889%
100%
33
A: 19.905%
C: 31.826%
G: 16.051%
T: 32.219%
0,093%
0,009%
0,009%

Training by counting



In theory, we are given (**X**,**Z**) pairs, but we are given **(X**, "**Z**") pairs, where the NCR-annotations have to be translated to **Z**s.

Also, we may ignore rare start and stop codons, i.e. a gene that starts (or ends) with a ignored start (or stop) codon does not correspond to a path in your model.

Training by counting – Typical solution

To set the transition probabilities:

N → N	N → CCC, where CCC is ATG	N → RRR, where RRR is TTA
	N → CCC, where CCC is GTG	N → RRR, where RRR is CTA
	N → CCC, where CCC is TTG	N → RRR, where RRR is TCA

We count:

#(N->N) = no. of occurrences of “NN” our annotations.

#(N → CCC, were CCC is XYZ) = no. of occurrence “NCCC” in our annotations, where CCC is an annotation of XYZ (in our training data).

#(N → RRR, were RRR is XYZ) = no. of occurrence “NRRR” in our annotations where, RRR is an annotation of XYZ (in our training data).

We compute:

Total = #(N → N) + #(N → CCC where CCC is XYZ) + #(N → RRR, where RRR is XYZ)

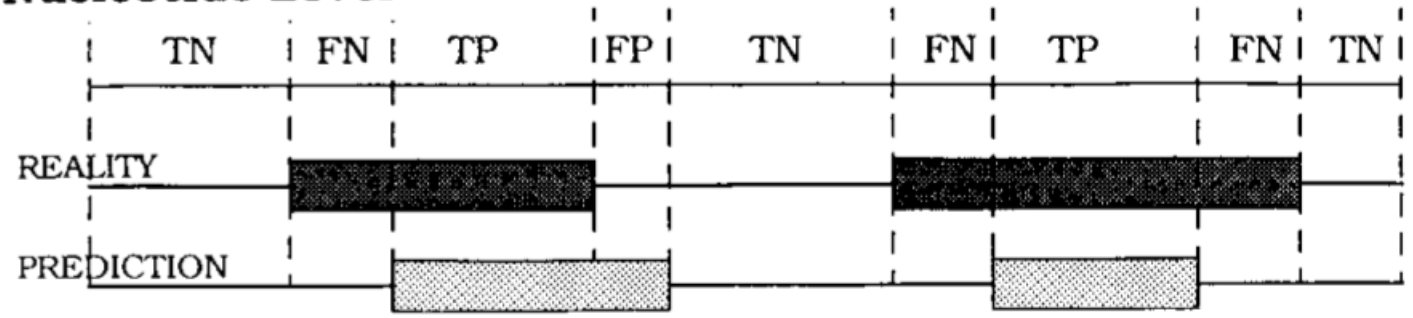
We set:

P(N → X) = #(N → X) / Total for each of the 7 transitions

Evaluating performance

Evaluation of Gene Structure Prediction Programs (Burset and Guigo, 1996)

Nucleotide Level



PREDICTION \ REALITY	coding	no coding	
coding	TP	FP	TP+FP
no coding	FN	TN	FN+TN
	TP+FN	TF+TN	

$$Sn = \frac{TP}{TP + FN}$$

Sensitivity

$$Sp = \frac{TP}{TP + FP}$$

Specificity

$$CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}}$$

Correlation Coefficient

$$ACP = \frac{1}{4}\left[\frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN}\right]$$

$$AC = (ACP - 0.5) \times 2$$

Approximate Correlation

compare_anns.py

```
$ python compare_anns.py true-ann6.fa pred-ann6.fa
Cs   (tp=757332, fp=164766, tn=305197, fn=57217): Sn = 0.9298, Sp = 0.8213, AC = 0.6213
Rs   (tp=715865, fp=127462, tn=304830, fn=57584): Sn = 0.9255, Sp = 0.8489, AC = 0.6603
Both (tp=1473197, fp=292228, tn=247613, fn=114801): Sn = 0.9277, Sp = 0.8345, AC = 0.4520
```

k-fold cross validation

For each genome 1 to 5, you train your model by training-by-counting on the remaining 4 genomes, and predict the gene structure on the genome you picked. You compute and report the approximate correlation coefficient (AC) between your prediction and the true annotation using the small python program compare_anns.py. Include a table with the computed ACs in your report.

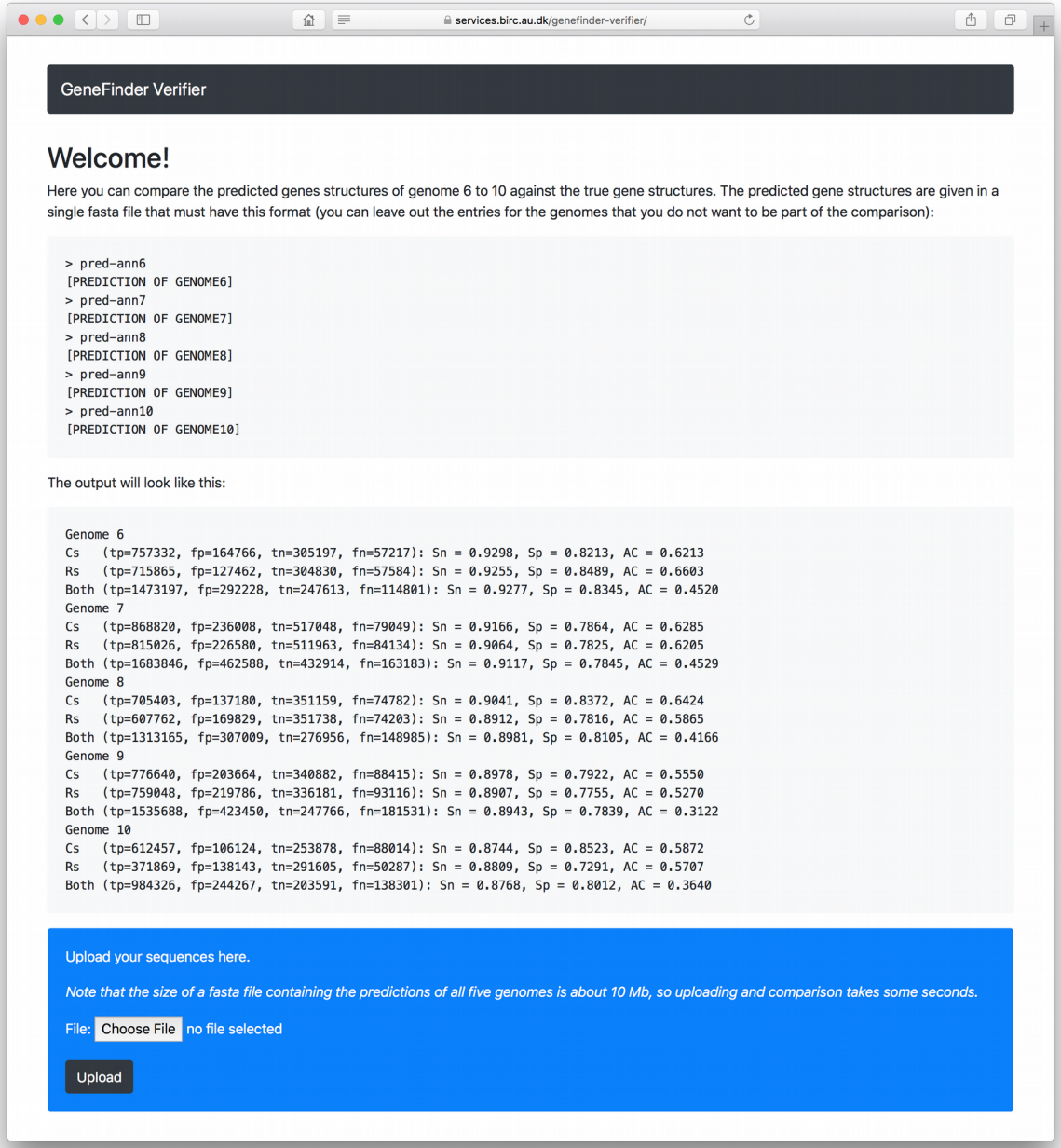
	Round 1	Round 2	Round 3	Round 4	Round 5
Genome 1	Validate	Train	Train	Train	Train
Genome 2	Train	Validate	Train	Train	Train
Genome 3	Train	Train	Validate	Train	Train
Genome 4	Train	Train	Train	Validate	Train
Genome 5	Train	Train	Train	Train	Validate

Table 1: ACs between the predictions on the genomes 1 to 5 and their true annotation. The gene predictor was trained by the remaining four genomes respectively.

	AC		
	Only Cs	**Only Rs**	**Both**
Genome 1	0.5983	0.6470	0.4347
Genome 2	0.6305	0.6529	0.4510
Genome 3	0.6552	0.6510	0.4805
Genome 4	0.6240	0.6002	0.4079
Genome 5	0.6550	0.6027	0.4302

	Round 1	**Round 2**	**Round 3**	**Round 4**	**Round 5**
Genome 1	Validate	Train	Train	Train	Train
Genome 2	Train	Validate	Train	Train	Train
Genome 3	Train	Train	Validate	Train	Train
Genome 4	Train	Train	Train	Validate	Train
Genome 5	Train	Train	Train	Train	Validate

Performance on genome 6-10



https://services.birc.au.dk/genefinder-verifier/

Performance on genome 6-10

Input is a fasta-file containing your predictions. It is about 10 Mb big!

```
> pred-ann6
[PREDICTION OF GENOME6]
> pred-ann7
[PREDICTION OF GENOME7]
> pred-ann8
[PREDICTION OF GENOME8]
> pred-ann9
[PREDICTION OF GENOME9]
> pred-ann10
[PREDICTION OF GENOME10]
```

```
$ cat pred-ann6.fa pred-ann7.fa pred-ann8.fa pred-ann9.fa pred-ann10.fa > pred-ann6-10.fa
```

Beware of the newlines!

pred-annX.fa should end with a newline in order for the above to work

Last year

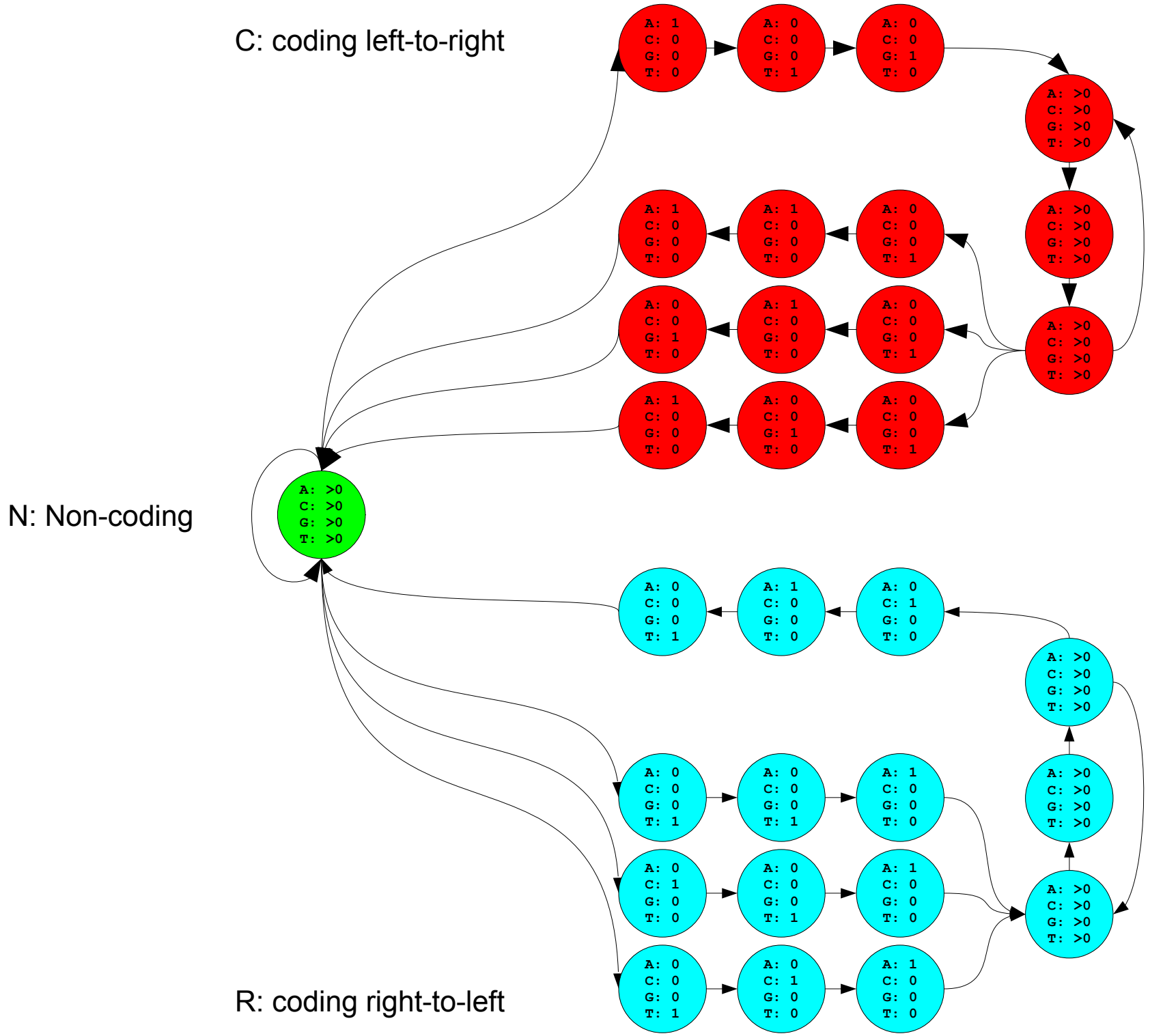
ACs between predictions and true annotations sorted by average

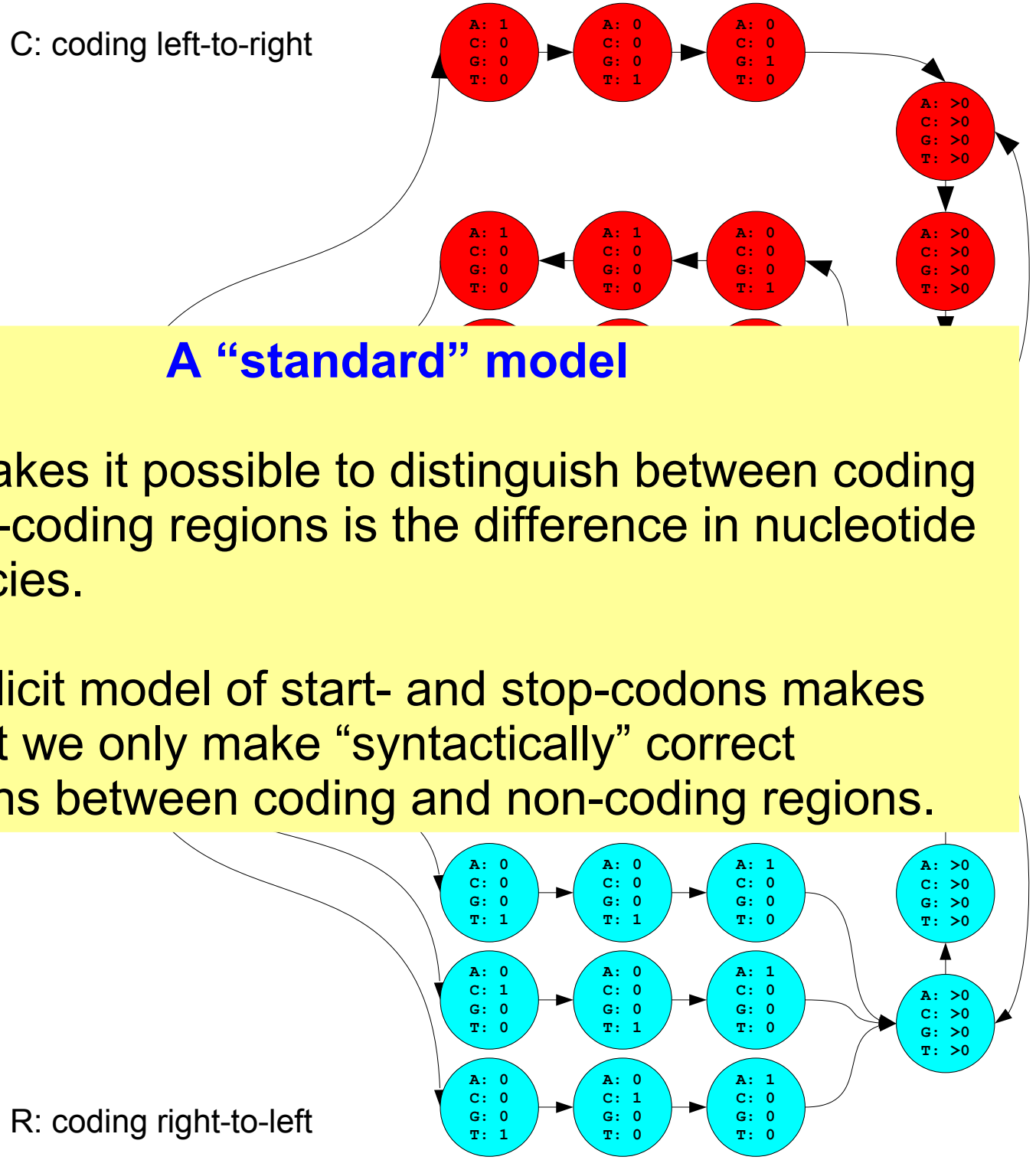
	Genome 6	Genome 7	Genome 8	Genome 9	Genome 10	Average
3	0,6313	0,6378	0,6007	0,5310	0,4762	**0,5754**
22	0,6313	0,6378	0,6007	0,5310	0,4762	**0,5754**
9	0,6053	0,6147	0,5708	0,4919	0,4197	**0,5405**
10	0,4628	0,4587	0,4245	0,3244	0,3611	**0,4063**
21	0,4619	0,4582	0,4219	0,3239	0,3603	**0,4052**
13	0,4539	0,4547	0,4159	0,3121	0,3650	**0,4003**
11	0,4522	0,4504	0,4172	0,3139	0,3643	**0,3997**
1	0,4520	0,4529	0,4165	0,3122	0,3646	**0,3996**
25	0,4543	0,4515	0,4144	0,3133	0,3596	**0,3986**
18	0,4521	0,4481	0,4112	0,3095	0,3619	**0,3966**
15	0,4501	0,4449	0,4187	0,2974	0,3586	**0,3939**
24	0,4237	0,4326	0,4063	0,2769	0,3816	**0,3842**
14	0,4359	0,4383	0,3974	0,2642	0,3467	**0,3765**
2	0,4286	0,4548	0,3725	0,2473	0,3738	**0,3753**
19	0,3856	0,4075	0,3727	0,2926	0,3059	**0,3529**
23	0,3854	0,4073	0,3551	0,2336	0,3710	**0,3505**
5	0,3850	0,4008	0,3619	0,2559	0,3119	**0,3431**
6	0,3821	0,3914	0,3504	0,2365	0,3263	**0,3373**
16	0,2917	0,3462	0,2618	0,2240	0,3213	**0,2890**
7	0,0247	0,0978	0,0311	0,0642	0,1441	**0,0724**
4	0,0254	0,0939	0,0253	0,0631	0,1387	**0,0693**
20	0,0775	0,0526	0,1344	-0,0246	0,0766	**0,0633**
12	0,0344	0,0820	0,0089	0,0549	0,1144	**0,0589**
17	-0,2748	-0,2571	-0,2635	-0,2657	-0,3075	**-0,2737**
8						

Codon-models

Models inspired by the model presented in class maybe allowing more start/stop codons.

Problems with training or prediction


C: coding left-to-right
N: Non-coding
R: coding right-to-left
A: 1 C: 0 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: 0 C: 0 G: 1 T: 0
A: >0 C: >0 G: >0 T: >0
A: 1 C: 0 G: 0 T: 0
A: 1 C: 0 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: >0 C: >0 G: >0 T: >0
A: 0 C: 0 G: 1 T: 0
A: 1 C: 0 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: >0 C: >0 G: >0 T: >0
A: 1 C: 0 G: 0 T: 0
A: 0 C: 0 G: 1 T: 0
A: 0 C: 0 G: 0 T: 1
A: >0 C: >0 G: >0 T: >0
A: >0 C: >0 G: >0 T: >0
A: 0 C: 0 G: 0 T: 1
A: 1 C: 0 G: 0 T: 0
A: 0 C: 1 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: 0 C: 1 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: 0 C: 0 G: 1 T: 0
A: 0 C: 1 G: 0 T: 0
A: 0 C: 0 G: 0 T: 1
A: 1 C: 0 G: 0 T: 0
A: 1 C: 0 G: 0 T: 0
A: 1 C: 0 G: 0 T: 0
A: 0 C: 1 G: 0 T: 0
A: 1 C: 0 G: 0 T: 0
A: 1 C: 0 G: 0 T: 0
A: >0 C: >0 G: >0 T: >0
A: >0 C: >0 G: >0 T: >0
A: >0 C: >0 G: >0 T: >0

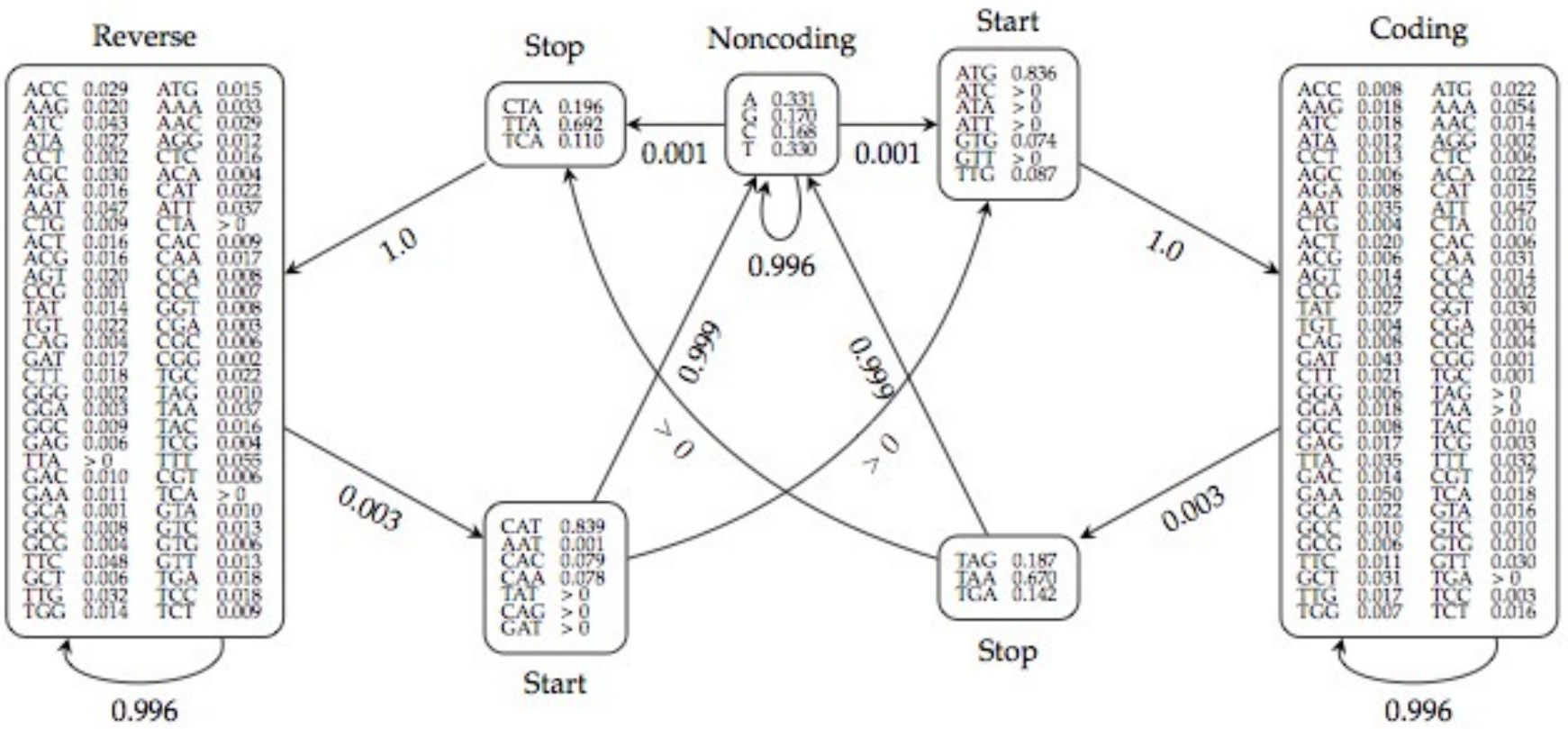


A “standard” model

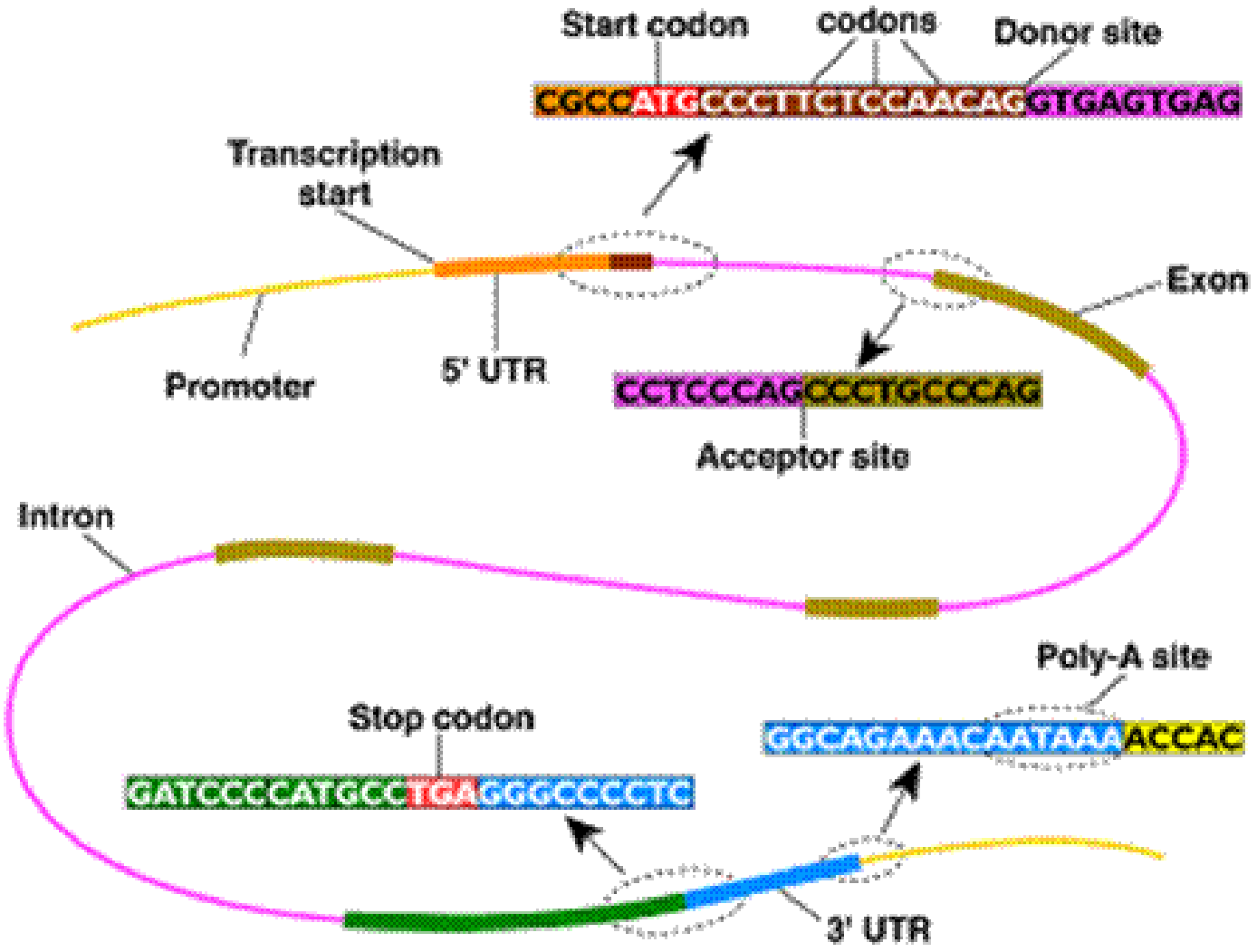
What makes it possible to distinguish between coding and non-coding regions is the difference in nucleotide frequencies.

The explicit model of start- and stop-codons makes sure that we only make “syntactically” correct transitions between coding and non-coding regions.

A “codon” model

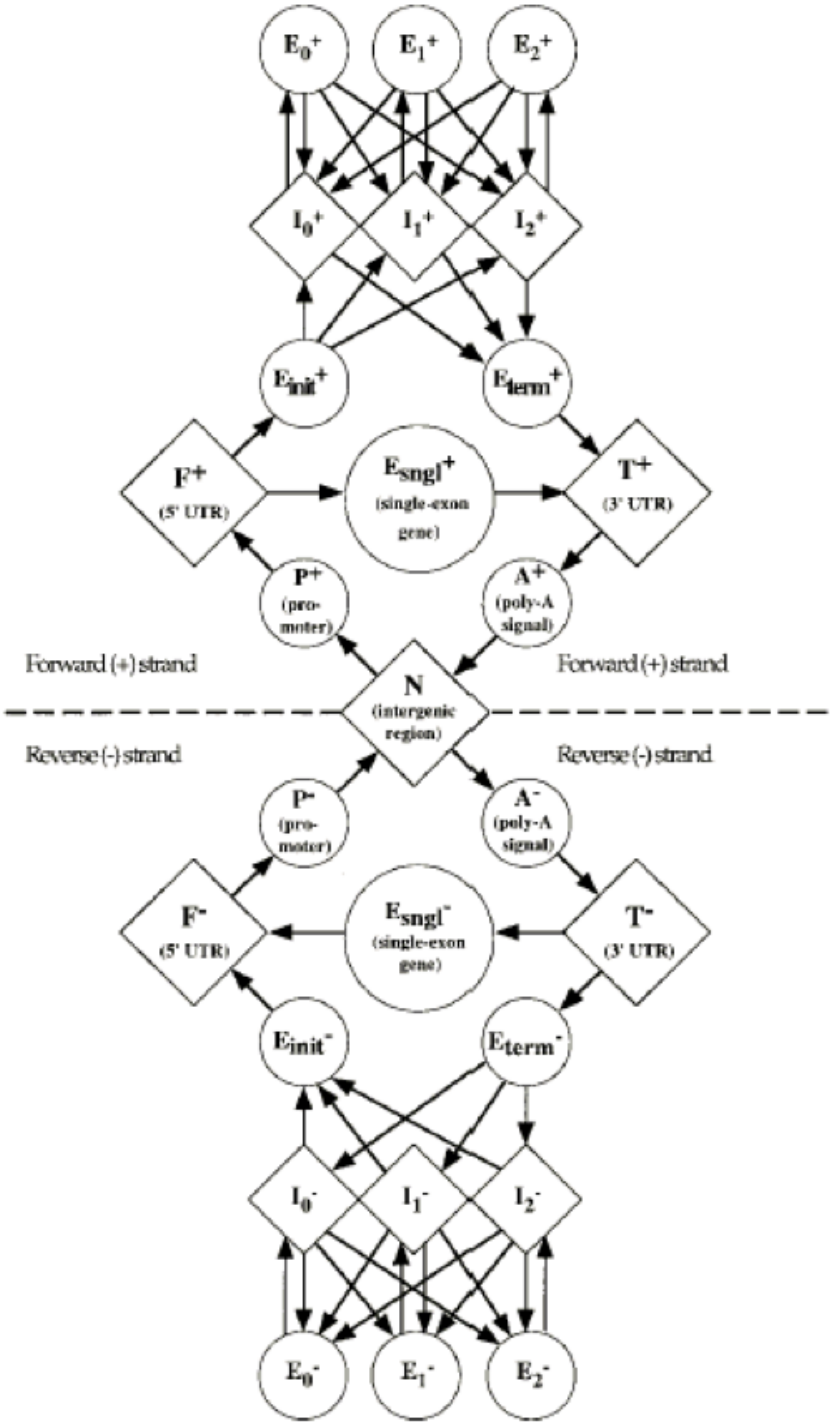


More complex gene finding problems



Eukaryotic gene structure

Eukaryotic gene structure in both directions (GenScan)



Exam

Overall: Oral, 20 min all included, 10 min for your **prepared** presentation, 5 min for discussion/question, you may be interrupted along the way. See Blackboard for all the details.

HMM Question: Hidden Markov Models (Basic algorithms and applications, Building models and selecting model parameters)

What is a HMM, explain/show parameters and assumptions

How to compute the joint probability P(X,Z)

Basic problems: Decoding and selecting models parameter, e,g:

How to to compute a Viterbi decoding, Z*, where $P(Z^*) = \max_Z P(X,Z)$

How to compute $P(X) = \text{sum}_Z P(X,Z)$ using the forward algorithm

How to compute a posterior decoding using the forward and backward algorithms

How to select parameters from (X,Z) using training-by-counting, or from X using EM

Remember to be **precise**, and remember **what, why, and how**, when you explain