ALGORITHMS OF BIOINFORMATICS

2

Hidden Messages

23 October 2025

Prof. Dr. Sebastian Wild

CS594 (Winter 2025-26)
Philipps-Universität Marburg
version `2025-11-06 12:07`

2.1 Biology Big Picture

Biology

- *biology* = the scientific study of *living* things
  - originally *naturalists*: individual people manually **observing** plants and animals
    e. g., *Darwin's finches*
  - gradually more scientific: controlled experiments, isolated mechanisms
    e. g., *Mendel's inheritance experiments on peas*
  - gradually more focus on molecular/chemical mechanisms: microscopes, biochemisty

Biology

- *biology* = the scientific study of *living* things
  - originally *naturalists*: individual people manually **observing** plants and animals
    e. g., *Darwin's finches*
  - gradually more scientific: controlled experiments, isolated mechanisms
    e. g., *Mendel's inheritance experiments on peas*
  - gradually more focus on molecular/chemical mechanisms: microscopes, biochemisty

- now clear: fundamental mechanisms (and origins!) of life are microscopic

⇝ fundamental mechanisms to be found in *molecular biology*

Bioinformatics

- 20th Century: discovery of DNA and genes
  - DNA stores information about biomolecules in **discrete form**
    human genome: 3.055 billion letter string over alphabet `{A,C,G,T}` (!)
  - ⇝ genetic information can **copied** precisely
    mutations are errors in the copying
  - double strands (backup!) and "coiling up" into chromosomes protects data
  - production of chemicals in living cells (*proteins*) is determined by *genes* (parts of DNA)

Zoom in on DNA
https://youtu.be/wZozOrFluiw

Bioinformatics

- 20th Century: discovery of DNA and genes
  - DNA stores information about biomolecules in **discrete form**
    human genome: 3.055 billion letter string over alphabet `{A,C,G,T}` (!)
  - ⇝ genetic information can **copied** precisely
    mutations are errors in the copying
  - double strands (backup!) and "coiling up" into chromosomes protects data
  - production of chemicals in living cells (*proteins*) is determined by *genes* (parts of DNA)

Zoom in on DNA
`https://youtu.be/wZozOrFluiw`

⇝ *Life itself has inherently **computational** components!* 🤯

Bioinformatics

- 20th Century: discovery of DNA and genes
  - DNA stores information about biomolecules in **discrete form**
    human genome: 3.055 billion letter string over alphabet `{A,C,G,T}` (!)
  - ⇝ genetic information can **copied** precisely
    mutations are errors in the copying
  - double strands (backup!) and "coiling up" into chromosomes protects data
  - production of chemicals in living cells (*proteins*) is determined by *genes* (parts of DNA)

Zoom in on DNA
`https://youtu.be/wZozOrFluiw`

⇝ *Life itself has inherently **computational** components!* 🤯

⇝ Computer science can contribute to the understanding these! ⇝ *bioinformatics*

Bioinformatics

Zoom in on DNA
https://youtu.be/wZozOrFluiw

- 20th Century: discovery of DNA and genes
  - DNA stores information about biomolecules in **discrete form**
    human genome: 3.055 billion letter string over alphabet {A, C, G, T} (!)
  - ⇝ genetic information can **copied** precisely
    mutations are errors in the copying
  - double strands (backup!) and "coiling up" into chromosomes protects data
  - production of chemicals in living cells (*proteins*) is determined by *genes* (parts of DNA)

⇝ *Life itself has inherently **computational** components!* 🤯

⇝ Computer science can contribute to the understanding these! ⇝ *bioinformatics*

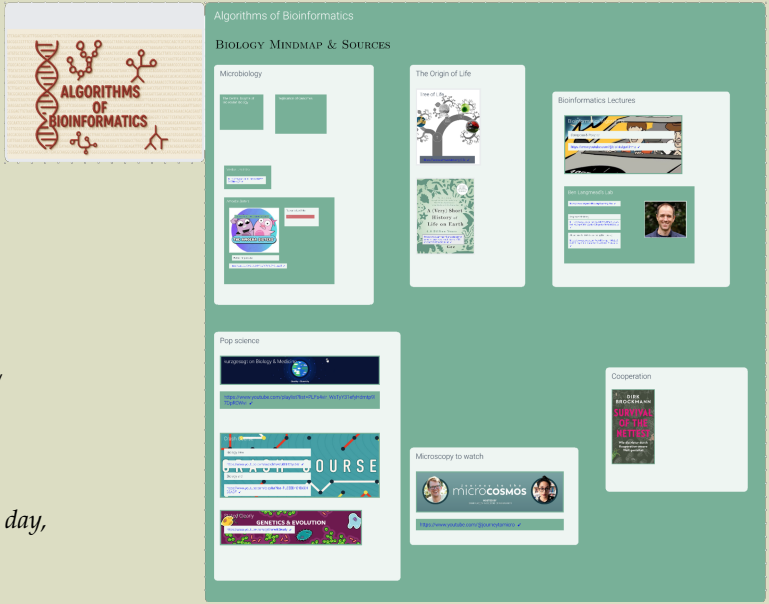
- But also: biology increasingly a data-centric field
  - much of knowledge discovery intrinsically reliant on computational analysis of collected data
  - e. g., reading the 3 billion letters of DNA is not possible with current lab techniques
    - ⇝ use computers to puzzle it together (see *Sequencing Unit*)
  - *"in silico"* experiments

Collection of (more or less) Fun Sources

Collaborative Mindmap on infinity maps

- Share useful resources
- Structure knowledge hierarchically
- Link on Campuswire / ILIAS

There's tons to learn,
new things discovered every day,
and it's about life itself!



Molecular Biology 101

Molecular Biology (Britannica concise)

- concerned with chemical structures and processes of biological phenomena at the molecular level
- developed out of biochemistry, genetics, and biophysics
- particularly concerned with the study of **proteins**, nucleic acids, and enzymes

Molecular Biology 101

Molecular Biology (Britannica concise)

- concerned with chemical structures and processes of biological phenomena at the molecular level
- developed out of biochemistry, genetics, and biophysics
- particularly concerned with the study of **proteins**, nucleic acids, and enzymes

Biology = lots of terminology and names . . .

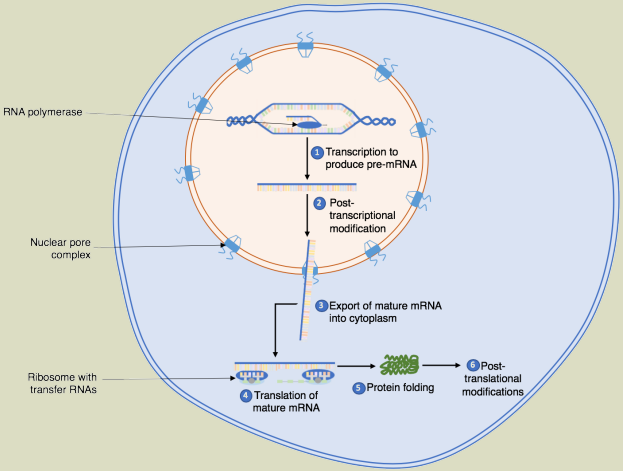
We will focus on mechanisms over terms, but a bit of context helps
let's make it at least whimsical (and maybe memorable)

Biomolecules (Updated 2023)
https://youtu.be/1Dx7LDwINLU

2.2 What are Genes?

The Central Dogma of Molecular Biology

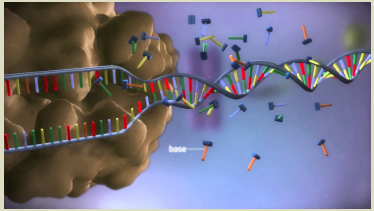
DNA makes **RNA** makes **Protein**



https://commons.wikimedia.org/wiki/File:Summary_of_the_protein_biosynthesis_process.png

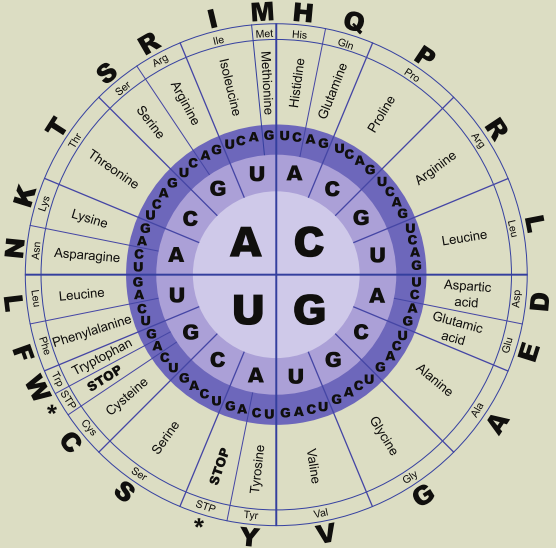
Protein Biosynthesis

- mechanism to produce *protein* a according to recipe stored in a *gene*



From DNA to protein - 3D
https://youtu.be/gG7uCskUOrA

Genetic Code



Compeau & Pevzner, *Bioinformatics Algorithms*, Fig. 4.1
https://cogniterra.org/lesson/29910/step/2?unit=22007

Within *ribosomes* (protein factories)

- translation
  - from RNA bases {A, C, G, U}
  - to amino acids (peptide) {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}
- uses *transfer RNA* "*chemical finite state transducer*"
- ***Genetic Code***: 3-base *codons* → amino acid

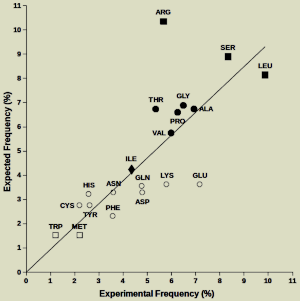
Inverse Codon Table

#Codons	Amino Acid (abbr.)		Codons
1	*Start*	>	AUG
4	Ala	*A*	GCU GCC GCA GCG
2	Cys	*C*	UGU UGC
2	Asp	*D*	GAU GAC
2	Glu	*E*	GAA GAG
2	Phe	*F*	UUU UUC
4	Gly	*G*	GGU GGC GGA GGG
2	His	*H*	CAU CAC
3	Ile	*I*	AUU AUC AUA
2	Lys	*K*	AAA AAG
6	Leu	*L*	CUU CUC CUA CUG UUA UUG
1	Met	*M*	AUG
2	Asn	*N*	AAU AAC
4	Pro	*P*	CCU CCC CCA CCG
2	Gln	*Q*	CAA CAG
6	Arg	*R*	CGU CGC CGA CGG AGA AGG
6	Ser	*S*	UCU UCC UCA UCG AGU AGC
4	Thr	*T*	ACU ACC ACA ACG
4	Val	*V*	GUU GUC GUA GUG
1	Trp	*W*	UGG
2	Tyr	*Y*	UAU UAC
3	*Stop*	<	UAA UAG UGA
1	Sec	*U*	(UGA)
1	Pyl	*O*	(UAG)

Inverse Codon Table

#Codons	Amino Acid (abbr.)		Codons
1	*Start*	>	AUG
4	Ala	*A*	GCU GCC GCA GCG
2	Cys	*C*	UGU UGC
2	Asp	*D*	GAU GAC
2	Glu	*E*	GAA GAG
2	Phe	*F*	UUU UUC
4	Gly	*G*	GGU GGC GGA GGG
2	His	*H*	CAU CAC
3	Ile	*I*	AUU AUC AUA
2	Lys	*K*	AAA AAG
6	Leu	*L*	CUU CUC CUA CUG UUA UUG
1	Met	*M*	AUG
2	Asn	*N*	AAU AAC
4	Pro	*P*	CCU CCC CCA CCG
2	Gln	*Q*	CAA CAG
6	Arg	*R*	CGU CGC CGA CGG AGA AGG
6	Ser	*S*	UCU UCC UCA UCG AGU AGC
4	Thr	*T*	ACU ACC ACA ACG
4	Val	*V*	GUU GUC GUA GUG
1	Trp	*W*	UGG
2	Tyr	*Y*	UAU UAC
3	*Stop*	<	UAA UAG UGA
1	Sec	*U*	(UGA)
1	Pyl	*O*	(UAG)

Amino Acid Frequencies in Human Proteins



https://doi.org/10.1371/journal.pone.0148174.g001

Some amino acids have several codons
(most frequent amino acids receive strongest error protection!)

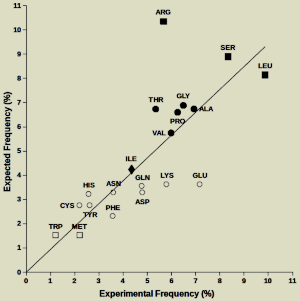
Inverse Codon Table

#Codons	Amino Acid (abbr.)		Codons
1	*Start*	>	AUG
4	Ala	*A*	GCU GCC GCA GCG
2	Cys	*C*	UGU UGC
2	Asp	*D*	GAU GAC
2	Glu	*E*	GAA GAG
2	Phe	*F*	UUU UUC
4	Gly	*G*	GGU GGC GGA GGG
2	His	*H*	CAU CAC
3	Ile	*I*	AUU AUC AUA
2	Lys	*K*	AAA AAG
6	Leu	*L*	CUU CUC CUA CUG UUA UUG
1	Met	*M*	AUG
2	Asn	*N*	AAU AAC
4	Pro	*P*	CCU CCC CCA CCG
2	Gln	*Q*	CAA CAG
6	Arg	*R*	CGU CGC CGA CGG AGA AGG
6	Ser	*S*	UCU UCC UCA UCG AGU AGC
4	Thr	*T*	ACU ACC ACA ACG
4	Val	*V*	GUU GUC GUA GUG
1	Trp	*W*	UGG
2	Tyr	*Y*	UAU UAC
3	*Stop*	<	UAA UAG UGA
1	Sec	*U*	(UGA)
1	Pyl	*O*	(UAG)

Some amino acids have several codons
(most frequent amino acids receive strongest error protection!)

Sometimes, stop codon UGA instead codes 21st amino acid *Selenocystein*...

Amino Acid Frequencies in Human Proteins



https://doi.org/10.1371/journal.pone.0148174.g001

But:

- non-ribosomal peptides (proteins not made according to central dogma)
- epigenetics (which genes are expressed)
- horizontal gene transfer (change genome during lifetime)
- retro viruses (inserts its one genes into host's genome!)
- proteins are also not the only active molecules (e. g., functional RNA)

Life finds a way . . . or a few dozen, just to be sure

2.3 Gene Detection

How can we find genes?

Recall: Gene = protein-coding region of DNA

Central options:

1. *ab initio* ("from the beginning"): just using the DNA
   - search for start (AUG) and stop codons (UAA, UAG, UGA) ⇝ *open reading frame*
   - search for promoter binding sites (docking station for transcription molecules)
   - bias of base frequencies in coding vs non-coding regions (hidden Markov models)
2. extrinsic methods: using additional (lab) data
   - e. g.sequencing messenger RNA from live cells (many more options)
   - comparison of genome to other species with known genes

Focus for today: Ab initio options

Why should there be any hope of finding hidden messages?

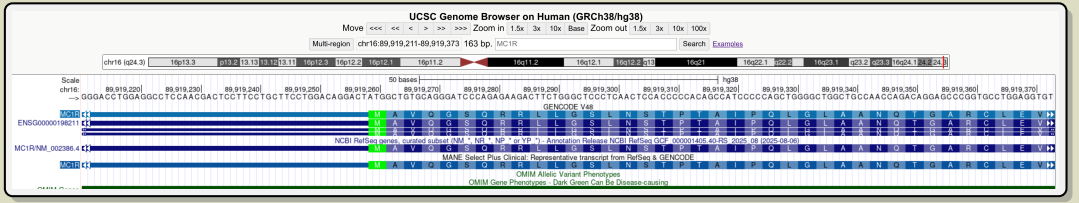
- Evolution!
- Random mutations always at play
- If functional part becomes dysfunctional, individual does not produce offspring
- other parts might be subject to random modifications

⇝ *signal*: property in a text that us unlikely to be present in random strings (noise)

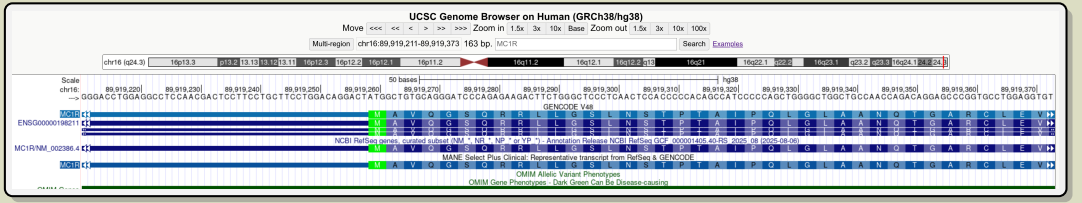
⇝ noise / *null model:* unused DNA is random

2.4 Waiting for Words

How big are genes?



How big are genes?

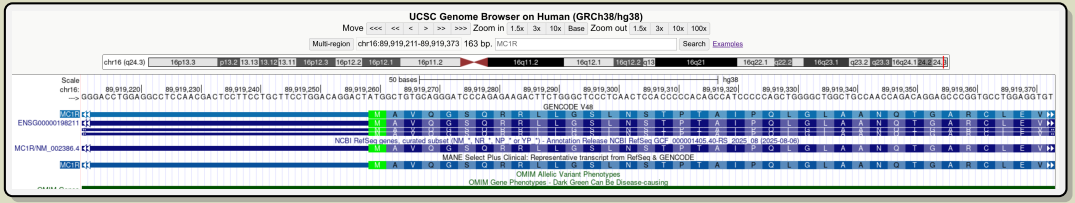


- only ~10% of human genome are genes
- length of (human) genes highly variable

base pairs (DNA strands!)

  - shortest known gene (*U7 snRNA*) has only 63 bp
  - longest gene (dystrophin) over 2M bp
    - but: 99% of that are *introns* (cut out before translation)! "split genes"
    - ⇝ transcription takes several hours (!)
  - moretypical: ~20K bp

How big are genes?



- only ~10% of human genome are genes
- length of (human) genes highly variable
  - shortest known gene (*U7 snRNA*) has only 63 bp
  - longest gene (dystrophin) over 2M bp
    - but: 99% of that are *introns* (cut out before translation)! "split genes"
    - ⇝ transcription takes several hours (!)
  - moretypical: ~20K bp

base pairs (DNA strands!)

⇝ *Can we reliably distinguish genes from randomly occurring open reading frames?*

Open Reading Frame Gene Detection

- ***Random RNA Model:*** String $D[0..N)$ generated i.i.d. uniformly
  i.e., each $D[i] \overset{\mathcal{D}}{=} \text{Uniform}(\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\})$
- ***Random Open Reading Frame:*** How many bp should we expect in **random RNA** between occurrences of the start codon `ATG` and **first** occurrence of any stop codon (`TAA`, `TAG`, `TGA`)?

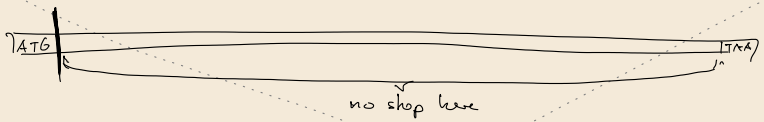
  (Recall: `U` in mRNA is `T` in DNA)

Clicker Question

In a string of random (i.i.d. uniform) DNA, what is the expected length of an open reading frame?

→ *sli.do/cs594*

Back-of-the-envelope

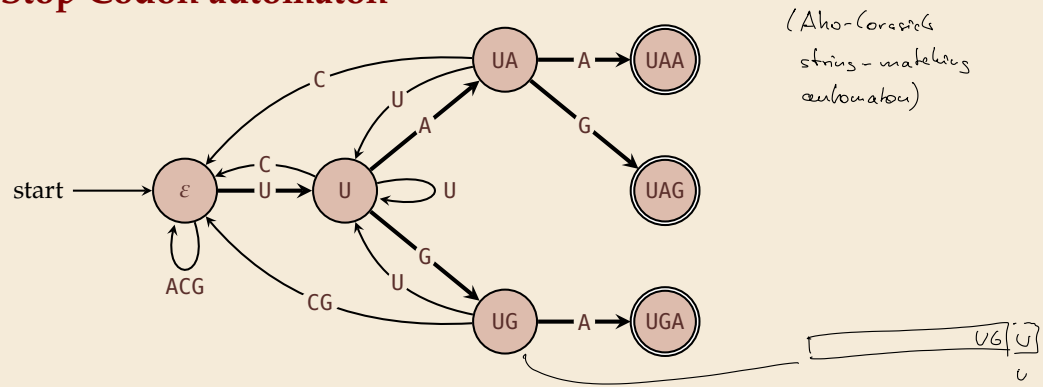


$$P\left[\boxed{?}\boxed{?}\boxed{?} \in \{TAA, TGA, TAG\}\right] = \frac{3}{64} \approx 0.05$$

maybe $\simeq$ 20 codons before stop

$\simeq$ 60 bp

Stop Codon automaton

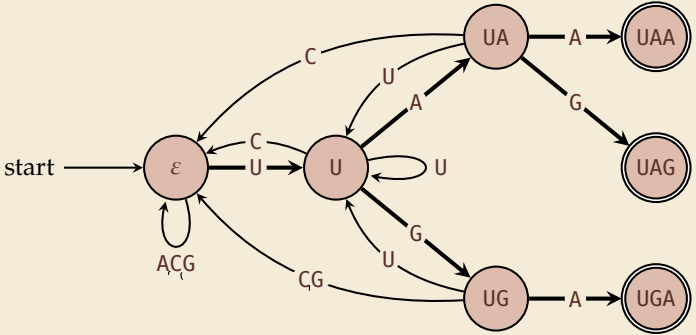


(Aho-Corasick string-matching automaton)

After seeing a start codon AUG, we accept the language of all strings that

- end with a stop codon **and**
- do not contain a stop codon earlier.

Stop Codon automaton

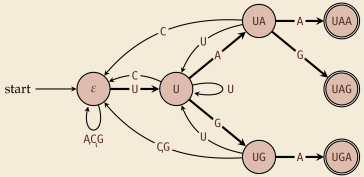


Expected number of characters from state q to some accepting state. (We want q_ε)

T_x := expected # chars from q_x to ◎

After seeing a start codon AUG, we accept the language of all strings that

- end with a stop codon **and**
- do not contain a stop codon earlier.



$$T_{UAA} = T_{UAG} = T_{UGA} = 0$$

$$T_{UA} = \frac{1}{4} \cdot T_{UAA} + \frac{1}{4} T_{UAG} + \frac{1}{4} T_U + \frac{1}{4} T_\varepsilon + 1$$

$$T_{UG} = \frac{1}{4} T_{UGA} + \frac{1}{4} T_U + \frac{1}{2} T_\varepsilon + 1$$

$$\vdots$$

$$T_\varepsilon = \frac{64}{3} = 21.\overline{3} \quad << 60$$

2.5 Probability Generating Functions

Probability Generating Functions

Expected values do not tell the full story . . . can we get at the distribution?

Probability Generating Functions

Expected values do not tell the full story . . . can we get at the distribution?

Definition 2.1 (pgf)

For $X \in \mathbb{N}_{\geq 0}$ a random variable, define its *probability generating function (pgf)* as

chars

$$G_X(z) = \sum_{k \geq 0} \mathbb{P}[X = k] \cdot z^k$$

Probability Generating Functions

Expected values do not tell the full story . . . can we get at the distribution?

Definition 2.1 (pgf)

For $X \in \mathbb{N}_{\geq 0}$ a random variable, define its *probability generating function (pgf)* as

$$G_X(z) = \sum_{k \geq 0} \mathbb{P}[X = k] \cdot z^k$$

$$\mathbb{E}[X] = \sum_{k \geq 0} \mathbb{P}[X = k] \cdot k$$

$$G_X'(z) \overset{1}{=} \sum_{k \geq 0} \frac{d}{dz} \mathbb{P}[X = k] \cdot z^k$$

$$= \mathbb{P}[X = k] \cdot k \cdot z^{k-1}$$

Lemma 2.2 (Moments from pgf)

1. The expected value of X is $\boxed{\mathbb{E}[X] = G_X'(1)}$
2. The variance of X is $\boxed{\mathrm{Var}[X] = G_X''(1) + G_X'(1) - \left(G_X'(1)\right)^2}$

$$\frac{d^2}{dz^2} \mathbb{P}[X = k] \cdot z^k = \mathbb{P}[X = k]\, k \cdot (k-1)\, z^{k-2} \qquad (k \geq 2)$$

$$G_X''(1)$$

$$\mathrm{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}X)^2\right] = \sum_{k \geq 0} \mathbb{P}[X = k] \left(k - \mathbb{E}[X]\right)^2 = \sum_{k \geq 0} \mathbb{P}[X = k] \left(k^2 - 2k\, \mathbb{E}[X] + \mathbb{E}[X]^2\right)$$

$$= \sum_{k \geq 0} \mathbb{P}[X=k]\, k^2 \;-\; 2\,\mathbb{E}[X] \underbrace{\sum_{k \geq 0} \mathbb{P}[X=k]\, k}_{\mathbb{E}[X]^2} \;+\; \mathbb{E}[X]^2 \underbrace{\sum_{k \geq 0} \mathbb{P}[X=k]}_{=1}$$

$k^2 = k(k-1) + k$

Example: Uniform Distribution

$U_n \stackrel{\mathcal{D}}{=} \text{Uniform}([0..n))$ (each value $u \in [0..n)$ with prob. $\frac{1}{n}$)

$\rightsquigarrow$ pgf: $U_n(z) = \frac{1}{n}\left(z^0 + z^1 + \cdots + z^{n-1}\right)$

Example: Uniform Distribution

$U_n \stackrel{\mathcal{D}}{=} \text{Uniform}([0..n))$ (each value $u \in [0..n)$ with prob. $\frac{1}{n}$)

$\rightsquigarrow$ pgf: $U_n(z) = \frac{1}{n}\left(z^0 + z^1 + \cdots + z^{n-1}\right) = \frac{1}{n}\frac{z^n - 1}{z - 1} \qquad (n \geq 1)$

Example: Uniform Distribution

$U_n \overset{\mathcal{D}}{=} \text{Uniform}([0..n))$ (each value $u \in [0..n)$ with prob. $\frac{1}{n}$)

$\rightsquigarrow$ pgf: $U_n(z) = \frac{1}{n}\left(z^0 + z^1 + \cdots + z^{n-1}\right) = \frac{1}{n}\frac{z^n - 1}{z - 1} \qquad (n \geq 1)$

- $\mathbb{E}[U_n] = U_n'(1) = \frac{n-1}{2}$

Example: Uniform Distribution

$U_n \stackrel{\mathcal{D}}{=} \text{Uniform}([0..n))$ (each value $u \in [0..n)$ with prob. $\frac{1}{n}$)

$\rightsquigarrow$ pgf: $U_n(z) = \frac{1}{n}\left(z^0 + z^1 + \cdots + z^{n-1}\right) = \frac{1}{n}\frac{z^n - 1}{z - 1} \qquad (n \geq 1)$

- $\mathbb{E}[U_n] = U_n'(1) = \frac{n-1}{2}$
- $\text{Var}[U_n] = U_n''(1) + U_n'(1) - U_n'(1)^2 = \frac{n^2 - 1}{12}$

Operations of pgfs

Lemma 2.3 (pgf of ind. r.v.)

Let $X, Y \in \mathbb{N}_{\geq 0}$ be *independent* random variables. Then $G_{X+Y}(z) = G_X(z) \cdot G_Y(z)$ ◀

$$\left(\sum_{k\geq 0} a_k z^k\right)\cdot\left(\sum_{k\geq 0} b_k z^k\right) = \sum_{k\geq 0} z^k \sum_{\ell=0}^{k} a_\ell\, b_{k-\ell}$$

convolution

Operations of pgfs

Lemma 2.3 (pgf of ind. r.v.)

Let $X, Y \in \mathbb{N}_{\geq 0}$ be *independent* random variables. Then $G_{X+Y}(z) = G_X(z) \cdot G_Y(z)$ ◀

- For $X_i \overset{\mathcal{D}}{=} \mathrm{B}(p)$ (1 with prob. p, 0 otherwise)
  we have $G_{X_i}(z) = pz + (1-p)z^0 = p(z-1) + 1$
- $Y = X_1 + \cdots + X_n \overset{\mathcal{D}}{=} \mathrm{Bin}(n, p)$
  we have $G_Y(z) = \prod_{i=1}^{n} G_{X_i}(z) = \big(pz + 1 - p\big)^n$

Operations of pgfs

Lemma 2.3 (pgf of ind. r.v.)

Let $X, Y \in \mathbb{N}_{\geq 0}$ be *independent* random variables. Then $G_{X+Y}(z) = G_X(z) \cdot G_Y(z)$ ◀

- For $X_i \overset{\mathcal{D}}{=} \mathrm{B}(p)$ (1 with prob. p, 0 otherwise)
  we have $G_{X_i}(z) = pz + (1-p)z^0 = p(z-1) + 1$
- $Y = X_1 + \cdots + X_n \overset{\mathcal{D}}{=} \mathrm{Bin}(n, p)$
  we have $G_Y(z) = \prod_{i=1}^{n} G_{X_i}(z) = \big(pz + 1 - p\big)^n$

$$\rightsquigarrow \mathbb{E}[Y] = G_Y'(1) = p \cdot n\big(pz + 1 - p\big)^{n-1}\Big|_{z=1} = np$$

Operations of pgfs

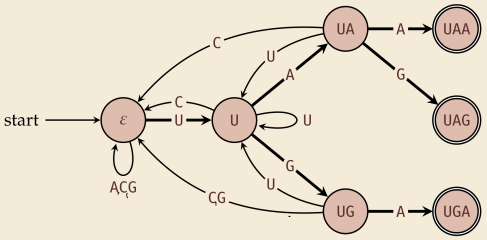
Lemma 2.3 (pgf of ind. r.v.)

Let $X, Y \in \mathbb{N}_{\geq 0}$ be *independent* random variables. Then $G_{X+Y}(z) = G_X(z) \cdot G_Y(z)$ ◀

- For $X_i \overset{\mathcal{D}}{=} \mathrm{B}(p)$ (1 with prob. p, 0 otherwise)
  we have $G_{X_i}(z) = pz + (1-p)z^0 = p(z-1)+1$
- $Y = X_1 + \cdots + X_n \overset{\mathcal{D}}{=} \mathrm{Bin}(n,p)$
  we have $G_Y(z) = \prod_{i=1}^{n} G_{X_i}(z) = \big(pz+1-p\big)^n$

$\rightsquigarrow$ $\mathbb{E}[Y] = G'_Y(1) = p \cdot n\big(pz+1-p\big)^{n-1}\Big|_{z=1} = np$

- $\mathrm{Var}[Y] = G''_Y(1)+G'_Y(1)-G'_Y(1)^2 = p^2n(n-1)+np-(np)^2 = np-np^2 = np(1-p)$



X_p = # steps from state p to $\textcircled{6}$

$G_p(z)$ = pgf of X_p

$$G_{UAA}(z) = 1$$

$$G_{UA}(z) = \frac{1}{4} \cdot z \cdot G_{UAA}(z) + \frac{1}{4} z\, G_{UAG}(z) + \frac{1}{4} z \cdot G_U(z) + \frac{1}{4} z\, G_\varepsilon(z)$$

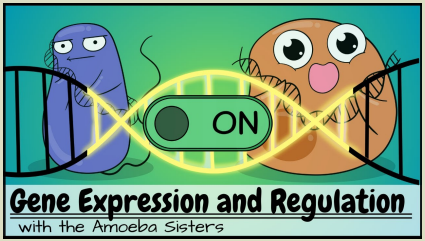
2.6 Motif finding

Gene regulation

- For gene expression, *RNA polymerase* needs to bind to DNA at beginning of a gene (to start transcription of gene in DNA into messenger RNA)
- *promoter* molecule can **stop transcription** by binding to this start
  - ⇝ RNA polymerase can't bind ⇝ no mRNA created ⇝ no protein made

⇝ *negative control*

- can also have promoters **enable** or **encourage** transcription ⇝ *positive control*



Gene Expression and Regulation
https://youtu.be/ebIpkw3XapE

Clock Genes in Plants

- most living beings have a *circadian rhythm*
  Who controls that?
  "roughly 24h"
  (more details in Chapter 2 of *Bioinformatics Algorithms*)
- in plants, negative feedback loop of 3 proteins
  1. *TOC1* **promotes** expression of *LHY* and *CCA1*
  2. sunlight triggers transcription *LHY* abd *CCA1*
  3. *LHY* abd *CCA1* **repress** expression of *TOC1*
  4. without sunlight, *LHY* abd *CCA1* production deminishes
  5. *TOC1* no longer blocked, can accumulate at night
  6. *TOC1* triggers expression of *LHY* and *CCA1* (ahead of light!)
- *TOC1*, *CCA1*, and *LHY* turn other genes on or off (*promoters*)

Clock Genes in Plants

- most living beings have a *circadian rhythm*
  Who controls that?
  "roughly 24h"
  (more details in Chapter 2 of *Bioinformatics Algorithms*)
- in plants, negative feedback loop of 3 proteins
  1. *TOC1* **promotes** expression of *LHY* and *CCA1*
  2. sunlight triggers transcription *LHY* abd *CCA1*
  3. *LHY* abd *CCA1* **repress** expression of *TOC1*
  4. without sunlight, *LHY* abd *CCA1* production deminishes
  5. *TOC1* no longer blocked, can accumulate at night
  6. *TOC1* triggers expression of *LHY* and *CCA1* (ahead of light!)
- *TOC1*, *CCA1*, and *LHY* turn other genes on or off (*promoters*)

same substring

⇝ genes with day/night rhythm should have **repeated binding sites** for *TOC1*/*CCA1*/*LHY*!

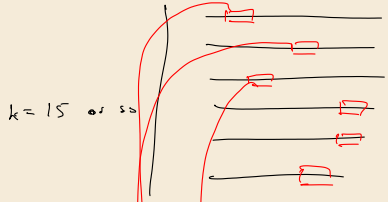
⇝ called a ***motif***

Motif Finding

Typical complication in bioinformatics: Nothing is exact . . .

Motif Finding

Typical complication in bioinformatics: Nothing is exact . . .



Consensus Pattern Problem

- **Given:** Collection of strings $G_1, \ldots, G_t \in \Sigma^n$, integer k
- **Goal:** Offsets $s_1, \ldots, s_t \in [0..n-k]$ such that $d_H\big(G_1[s_1..s_1+k), \ldots, G_t[s_t..s_n+k)\big)$ is minimized
- $d_H(T_1, \ldots, T_n)$ = "total Hamming distance" = total number of non-majority chars

d_H motif score

Motifs

T	C	G	G	G	G	g	T	T	T	t	t
c	C	G	G	t	G	A	c	T	T	a	C
a	C	G	G	G	G	A	T	T	T	t	C
T	t	G	G	G	G	A	c	T	T	t	t
a	a	G	G	G	G	A	c	T	T	C	C
T	t	G	G	G	G	A	c	T	T	C	C
T	C	G	G	G	G	A	T	T	c	a	t
T	C	G	G	G	G	A	T	T	c	C	t
T	a	G	G	G	G	A	a	c	T	a	C
T	C	G	G	G	t	A	T	a	a	C	C

Score(*Motifs*) $3 + 4 + 0 + 0 + 1 + 1 + 1 + 5 + 2 + 3 + 6 + 4 = 30$

Compeau & Pevzner, *Bioinformatics Algorithms*, Fig. 2.2

https://cogniterra.org/lesson/29868/step/2?unit=21966

Median String

- Equivalently:

$$d_H(T_1, \ldots, T_t) = \sum_{i=1}^{t} d_H(\bar{T}, T_t)$$

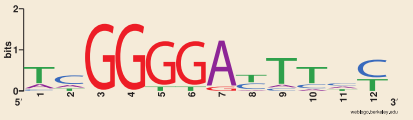
for the *consensus string* $\bar{T}$:

for all $j \in [0..n)$:
$\bar{T}[j] = \text{majority}(T_1[j], \ldots, T_t[j])$

Motif with consensus and profile

3

Motifs	T	C	G	G	G	G	g	T	T	T	t	t
	c	C	G	G	t	G	A	c	T	T	a	C
	a	C	G	G	G	G	A	T	T	T	t	C
	T	t	G	G	G	G	A	c	T	T	t	t
	a	a	G	G	G	G	A	c	T	T	C	C
	T	t	G	G	G	G	A	c	T	T	C	C
	T	C	G	G	G	G	A	T	T	c	a	t
	T	C	G	G	G	G	A	T	T	c	C	t
	T	a	G	G	G	G	A	a	c	T	a	C
	T	C	G	G	G	t	A	T	a	a	C	C
Score(*Motifs*)	3	+ 4	+ 0	+ 0	+ 0	+ 1	+ 1	+ 1	+ 5	+ 2	+ 3	+ 6	+ 4	= 30
Count(*Motifs*)	A:	2	2	0	0	0	0	9	1	1	1	3	0
	C:	1	6	0	0	0	0	0	4	1	2	4	6
	G:	0	0	10	10	9	9	1	0	0	0	0	0
	T:	7	2	0	0	1	1	0	5	8	7	3	4
Profile(*Motifs*)	A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
	C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
	G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
	T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4
Consensus(*Motifs*)	T	C	G	G	G	G	A	T	T	T	C	C



Bad News

- Consensus Pattern Problem is NP-hard (even for binary alphabet)

  Elias: *Settling the Intractability of Multiple Alignment* , J. of Computational Biology **2006**

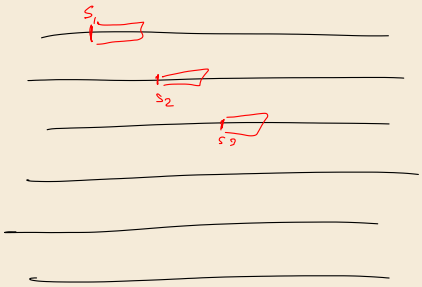
- even W[1]-hard for parameter ~~n~~ t (# strings) unbounded alphabet size

Brute-Force Options

There are two brute-force options:

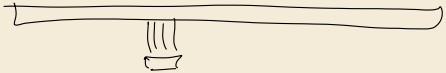
1. Try all combinations of starting indices
   - Each $s_i \in [0..n-k]$ $\rightsquigarrow$ search space $(n-k+1)^t$
   - Computing score for each is effort $O(t \cdot k)$

   $\rightsquigarrow$ Total cost $O(n^t t k)$ $\quad (k \ll n, t)$



Brute-Force Options

There are two brute-force options:



1. Try all combinations of starting indices
   - Each $s_i \in [0..n-k]$ $\rightsquigarrow$ search space $(n-k+1)^t$
   - Computing score for each is effort $O(t \cdot k)$

   $\rightsquigarrow$ Total cost $O(n^t t k)$ $\quad (k \ll n, t)$

2. Try all consensus strings.
   - Try all $\bar{T} \in \Sigma^k$ $\rightsquigarrow$ search space σ^k (for $\sigma = |\Sigma|$)
   - for each, $\bar{T}$ and each string G_i, find best s_i $\rightsquigarrow$ $t \cdot (n-k+1)$ options

     Note: Crucial that for given $\bar{T}$, scores from G_i are independent

   $\rightsquigarrow$ Total cost $O(\sigma^k t n)$ $\quad (k \ll n, t)$

Brute-Force Options

There are two brute-force options:

1. Try all combinations of starting indices
   - Each $s_i \in [0..n-k]$ $\rightsquigarrow$ search space $(n-k+1)^t$
   - Computing score for each is effort $O(t \cdot k)$

   $\rightsquigarrow$ Total cost $O(n^t t k)$ $\quad (k \ll n, t)$

2. Try all consensus strings.
   - Try all $\bar{T} \in \Sigma^k$ $\rightsquigarrow$ search space σ^k (for $\sigma = |\Sigma|$)
   - for each, $\bar{T}$ and each string G_i, find best s_i $\rightsquigarrow$ $t \cdot (n-k+1)$ options

     Note: Crucial that for given $\bar{T}$, scores from G_i are independent

   $\rightsquigarrow$ Total cost $O(\sigma^k t n)$ $\quad (k \ll n, t)$

Better, but still not feasible for $t \geq 15$ (or so) and $\sigma = 4 \ldots$

2.7 Local search heuristics

Heuristic motif finding

Exact solutions for Consensus Pattern Problem seem out of reach.

⇝ *Give up optimality guarantee.*

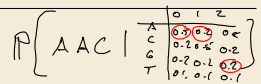
heuristic

\+

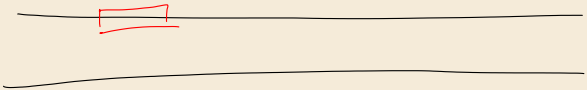
approximation

Greedy Incremental

```
1  procedure greedyMotif(G_1, ..., G_t, k)
2      s*_1, ..., s*_t := 0 // best so far
3      for s_1 := 0, ..., n − k // try all s_1
4          for i := 2, ..., t
5              Compute profile P[0..k) from G_j[s_j..s_j + k) for j ∈ [1..i)
6              s_i := arg max_s ℙ[G_i[s..s + k) | P]   // most-likely next row
7          if d_H(G_1[s_1..s_1 + k), ..., G_t[s_t..s_t + k)) < d_H(G_1[s*_1..s*_1 + k), ..., G_t[s*_t..s*_t + k))
8              s*_1, ..., s*_t := s_1, ..., s_t // better
9      return s*_1, ..., s*_t
```



$= 0.5 \cdot 0.2 \cdot 0.2$



Laplace

start counting at 1 instead of 0

removes 0s from profile

Motif with consensus and profile

	5'	T	C	G	G	G	G	g	T	T	T	t	t	3'
		c	C	G	G	t	G	A	c	T	T	a	C	
		a	C	G	G	G	G	A	T	T	T	t	C	
Motifs		T	t	G	G	G	G	A	c	T	T	t	t	
		a	a	G	G	G	G	A	c	T	T	C	C	
		T	t	G	G	G	G	A	c	T	T	C	C	
		T	C	G	G	G	G	A	T	T	c	a	t	
		T	C	G	G	G	G	A	T	T	c	C	t	
		T	a	G	G	G	G	A	a	c	T	a	C	
		T	C	G	G	G	t	A	T	a	a	C	C	

Score(*Motifs*)	3	+	4	+	0	+	0	+	1	+	1	+	1	+	5	+	2	+	3	+	6	+	4	=	30

Count(*Motifs*)													
A:	2	2	0	0	0	0	9	1	1	1	3	0
C:	1	6	0	0	0	0	0	4	1	2	4	6
G:	0	0	10	10	9	9	1	0	0	0	0	0
T:	7	2	0	0	1	1	0	5	8	7	3	4

Profile(*Motifs*)												
A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4

Consensus(*Motifs*)												
	T	C	G	G	G	A	T	T	T	C	C

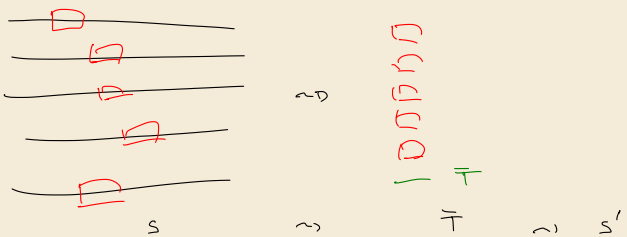
Greedy Incremental

```
1 procedure greedyMotif(G_1, ..., G_t, k)
2     s*_1, ..., s*_t := 0 // best so far
3     for s_1 := 0, ..., n − k // try all s_1
4         for i := 2, ..., t
5             Compute profile P[0..k) from G_j[s_j..s_j + k) for j ∈ [1..i)
6             s_i := arg max_s ℙ[G_i[s..s + k) | P]
7         if d_H(G_1[s_1..s_1 + k), ..., G_t[s_t..s_t + k)) < d_H(G_1[s*_1..s*_1 + k), ..., G_t[s*_t..s*_t + k))
8             s*_1, ..., s*_t := s_1, ..., s_t // better
9     return s*_1, ..., s*_t
```

- deterministic
- highly sensitive to order of genomes ...
- easy to get stuck in local optimum (wrt to order)

Hill Climbing

```
1  procedure randomLocalSearch(G_1, ..., G_t, k)
2      Randomly choose s_1, ..., s_t ∈ [0..n − k]
3      s*_1, ..., s*_t := s_1, ..., s_t // Remember best so far
4      repeat forever
5          Compute profile P[0..k) from G_j[s_j..s_j + k) for j ∈ [1..i)
6          for i := 1, ..., t:
7              s_i := arg max_s ℙ[G_i[s..s + k) | P]
8          if d_H(G_1[s_1..s_1 + k), ..., G_t[s_t..s_t + k)) < d_H(G_1[s*_1..s*_1 + k), ..., G_t[s*_t..s*_t + k))
9              s*_1, ..., s*_t := s_1, ..., s_t
10         else
11             return s*_1, ..., s*_t
```



Hill Climbing

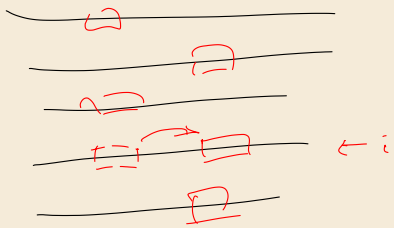
```
1  procedure randomLocalSearch(G_1, ..., G_t, k)
2      Randomly choose s_1, ..., s_t ∈ [0..n − k]
3      s*_1, ..., s*_t := s_1, ..., s_t // Remember best so far
4      repeat forever
5          Compute profile P[0..k) from G_j[s_j..s_j + k) for j ∈ [1..i)
6          for i := 1, ..., t:
7              s_i := arg max_s ℙ[G_i[s..s + k) | P]
8          if d_H(G_1[s_1..s_1 + k), ..., G_t[s_t..s_t + k)) < d_H(G_1[s*_1..s*_1 + k), ..., G_t[s*_t..s*_t + k))
9              s*_1, ..., s*_t := s_1, ..., s_t
10         else
11             return s*_1, ..., s*_t
```

- deterministic for a given starting point
- always terminates in local optimum
- must be repeated many times to not be stuck in bad local optimum

=> repeat and merge stop hill climb early

Gibbs Sampler

1 **procedure** gibbsSampler($G_1, \ldots, G_t, k, R$)
2 Randomly choose $s_1, \ldots, s_t \in [0..n-k]$
3 $s_1^*, \ldots, s_t^* := s_1, \ldots, s_t$ // *Remember best so far*
4 repeat R times:
5 $i :=$ random $[1..t]$
6 Compute profile $P[0..k)$ from $G_j[s_j..s_j+k)$ **for** $j \in [t] \setminus \{i\}$
7 $s_i :=$ random in $[0..n-k]$ w/p $\propto \mathbb{P}\left[G_i[s..s+k) \,\middle|\, P\right]$
8 **if** $d_H\big(G_1[s_1..s_1+k), \ldots, G_t[s_t..s_t+k)\big) < d_H\big(G_1[s_1^*..s_1^*+k), \ldots, G_t[s_t^*..s_t^*+k)\big)$
9 $s_1^*, \ldots, s_t^* := s_1, \ldots, s_t$ // *better*
10 **return** $s_1^*, \ldots, s_t^*$



Gibbs Sampler

1 **procedure** gibbsSampler($G_1, \ldots, G_t, k, R$)
2 Randomly choose $s_1, \ldots, s_t \in [0..n-k]$
3 $s_1^*, \ldots, s_t^* := s_1, \ldots, s_t$ // *Remember best so far*
4 repeat R times:
5 $i :=$ random $[1..t]$
6 Compute profile $P[0..k)$ from $G_j[s_j..s_j+k)$ **for** $j \in [t] \setminus \{i\}$
7 $s_i :=$ random in $[0..n-k]$ w/p $\propto \mathbb{P}\left[G_i[s..s+k) \,\middle|\, P\right]$
8 **if** $d_H\big(G_1[s_1..s_1+k), \ldots, G_t[s_t..s_t+k)\big) < d_H\big(G_1[s_1^*..s_1^*+k), \ldots, G_t[s_t^*..s_t^*+k)\big)$
9 $s_1^*, \ldots, s_t^* := s_1, \ldots, s_t$ // *better*
10 **return** $s_1^*, \ldots, s_t^*$


Less prone to get stuck in local optima


still no performance guarantee