Tiny gifts from my holiday

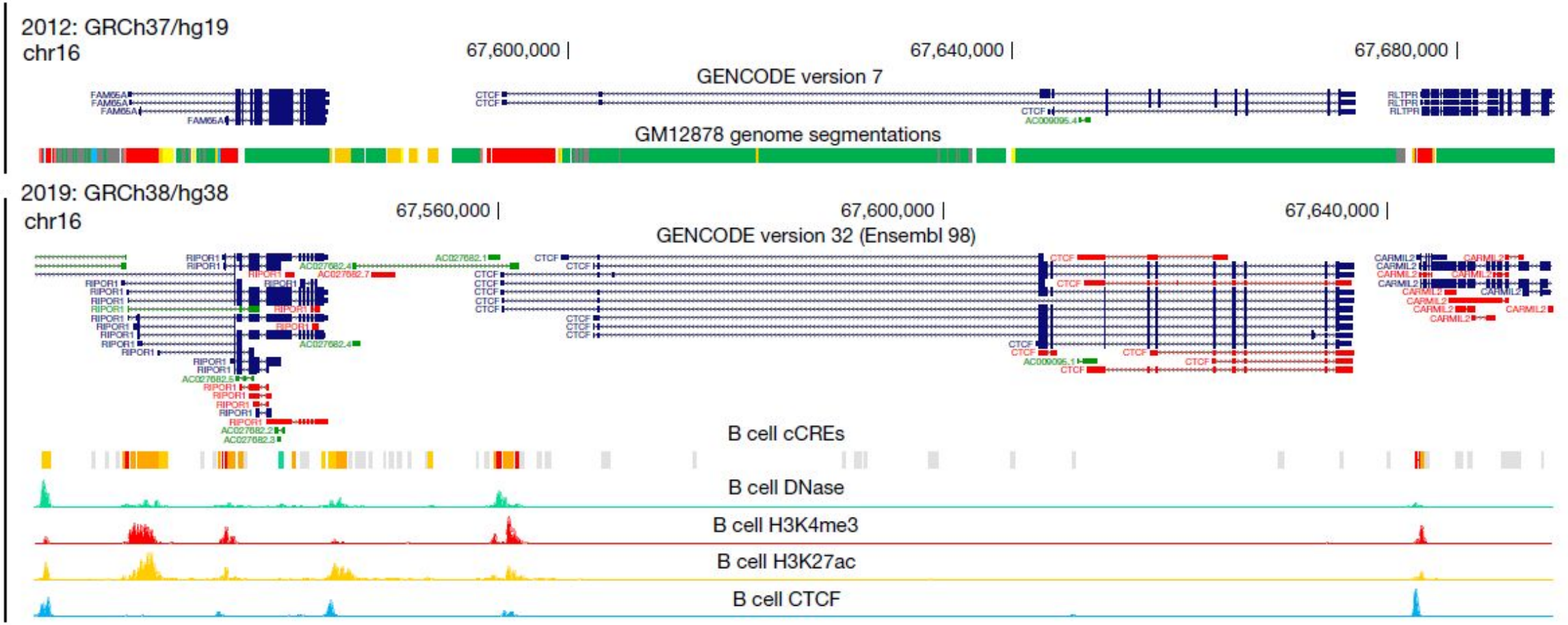
Lai da palpuogna,
Sunday

Davos,
Wednesday

Wrap-up of offline activities of Lecture 2

- 14 and 15 students replied to the anonymous survey and the offline activity form - **thank you!**
- A few points from the offline activity:
  - The efficacy, very broadly speaking, was tested *in vitro* (cells) and *in vivo* (animal models), and finally in clinical trials in human.
  - Quite a few participants replied correctly about the favourable scaling of pharmacokinetic properties as the reason for further development, but had difficulty understanding it. *No problem, it will be addressed later in the course.*
  - The exposure was measured with AUC. The mathematical operation is integration.
  - Where are mathematics and informatics used? *I found the results, which together draw a full picture, very interesting.*
  - Questions:
    - Is use of appropriate cell line critical? *It depends.*
    - How to understand such papers without biology background? *Great question, let's figure it out together.*
    - How drug finds the target in the body? Does the administration route make a difference? *Depends on the modality.*
    - What does structure-guided mean? *It's the topic of lecture 4 and 5.*

AMIDD Lecture 3: Biological Sequence Analysis



Perspectives on ENCODE, Nature 2020

Dr. Jitao David Zhang, Computational Biologist

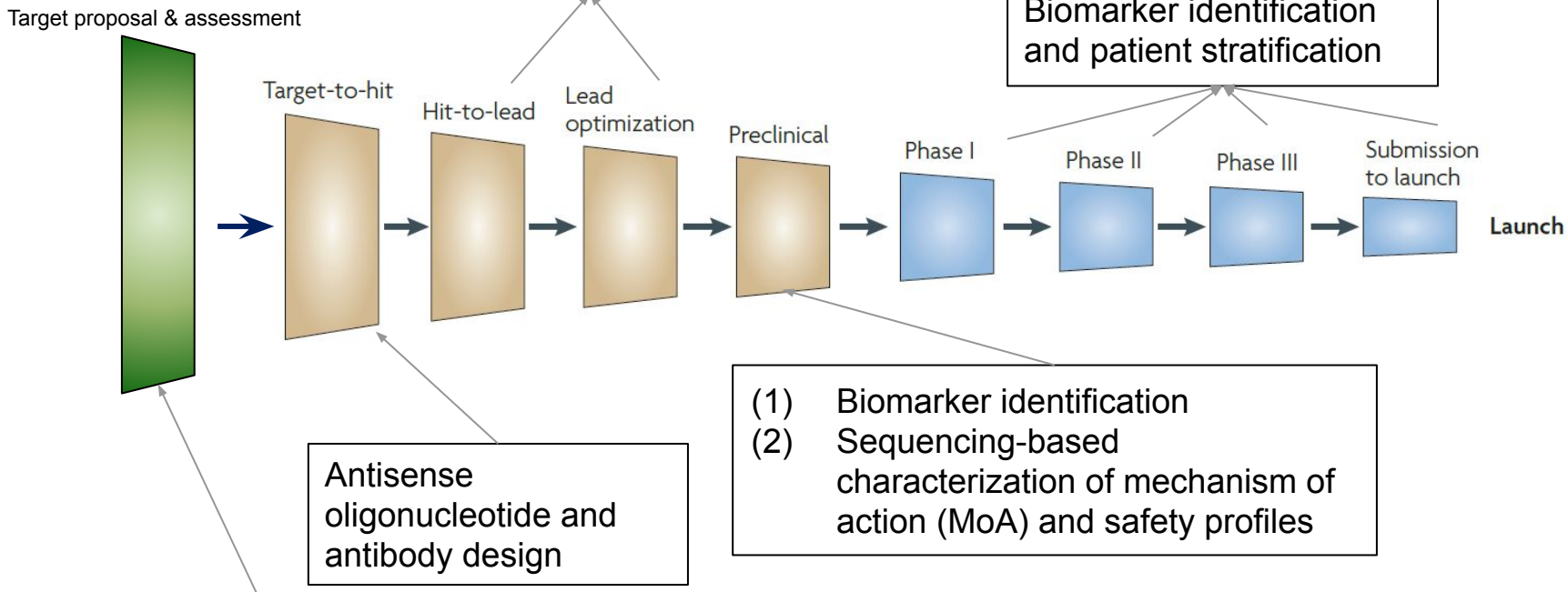
[1] Pharmaceutical Sciences, Pharma Research and Early Development, Roche Innovation Center Basel, F. Hoffmann-La Roche

[2] Department of Mathematics and Informatics, University of Basel

Today's goals

- Typical questions addressed by biological sequence analysis in drug discovery
- The deterministic view of sequence analysis: the edit distance and dynamic programming
- The probabilistic view of sequence analysis: Markov Chains and Hidden Markov Chains

Examples of typical questions addressed by sequence analysis in drug discovery



Sequencing-based characterization of drug candidates

Biomarker identification and patient stratification

Phase IV study with mutation monitoring for antiviral medicines

(1) Biomarker identification
(2) Sequencing-based characterization of mechanism of action (MoA) and safety profiles

Antisense oligonucleotide and antibody design

(1) Sequence similarity between the target human gene and its orthologues in animals
(2) Genetic variants in the general population and in patient (sub)populations
(3) Sequencing-based target expression

Adapted from Paul *et al.* "How to Improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge." Nature Reviews Drug Discovery, 2010

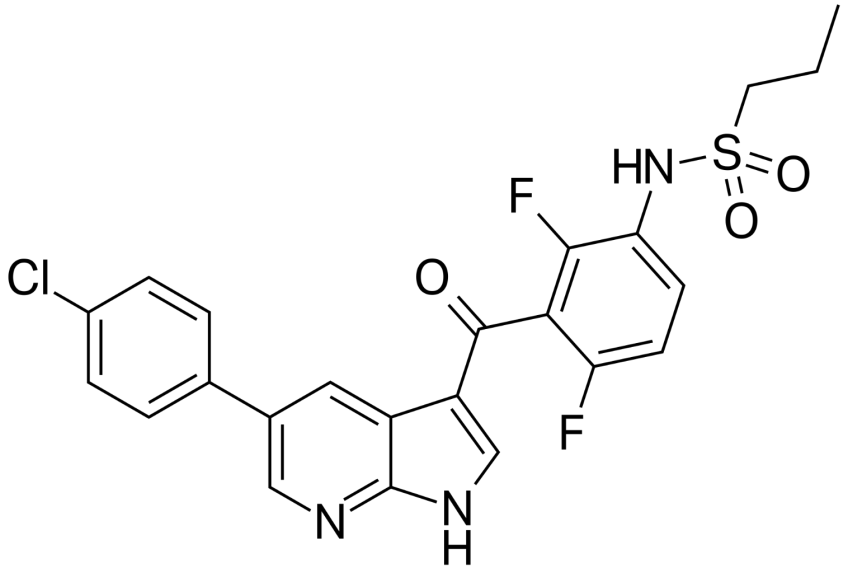
Vemurafenib (Zelboraf, PLX4032)
V600E mutated BRAF inhibition

- V600E: Valine (V) on the amino-acid position 600 is substituted by glutamic acid (E).

```
                560        570        580        590        600
IKLIDIARQT AQGMDYLHAK SIIHRDLKSN NIFLHEDLTV KIGDFGLATV
                610        620        630        640        650
```

Fragment of BRAF protein. Source: UniProtKB, P15056 (BRAF_HUMAN)

- View the 3D structure of the molecule at PDB ligand database
- View the X-ray structure of BRAF in complex with PLX4032 on PDB: accession number 3OG7.
- Find more information about the discovery and clinical efficacy of vemurafenib in the required read.



Source: https://commons.wikimedia.org/wiki/File:Vemurafenib_structure.svg

A single-amino-acid difference in BRAF gene may mean longer survival of melanoma patients given the correct treatment

McArthur, Grant A., Paul B. Chapman, Caroline Robert, James Larkin, John B. Haanen, Reinhard Dummer, Antoni Ribas, *et al.*

Safety and Efficacy of Vemurafenib in BRAFV600E and BRAFV600K Mutation-Positive Melanoma (BRIM-3): Extended Follow-up of a Phase 3, Randomised, Open-Label Study

The Lancet Oncology 15, Nr. 3 (1. März 2014): 323–32. https://doi.org/10.1016/S1470-2045(14)70012-9.

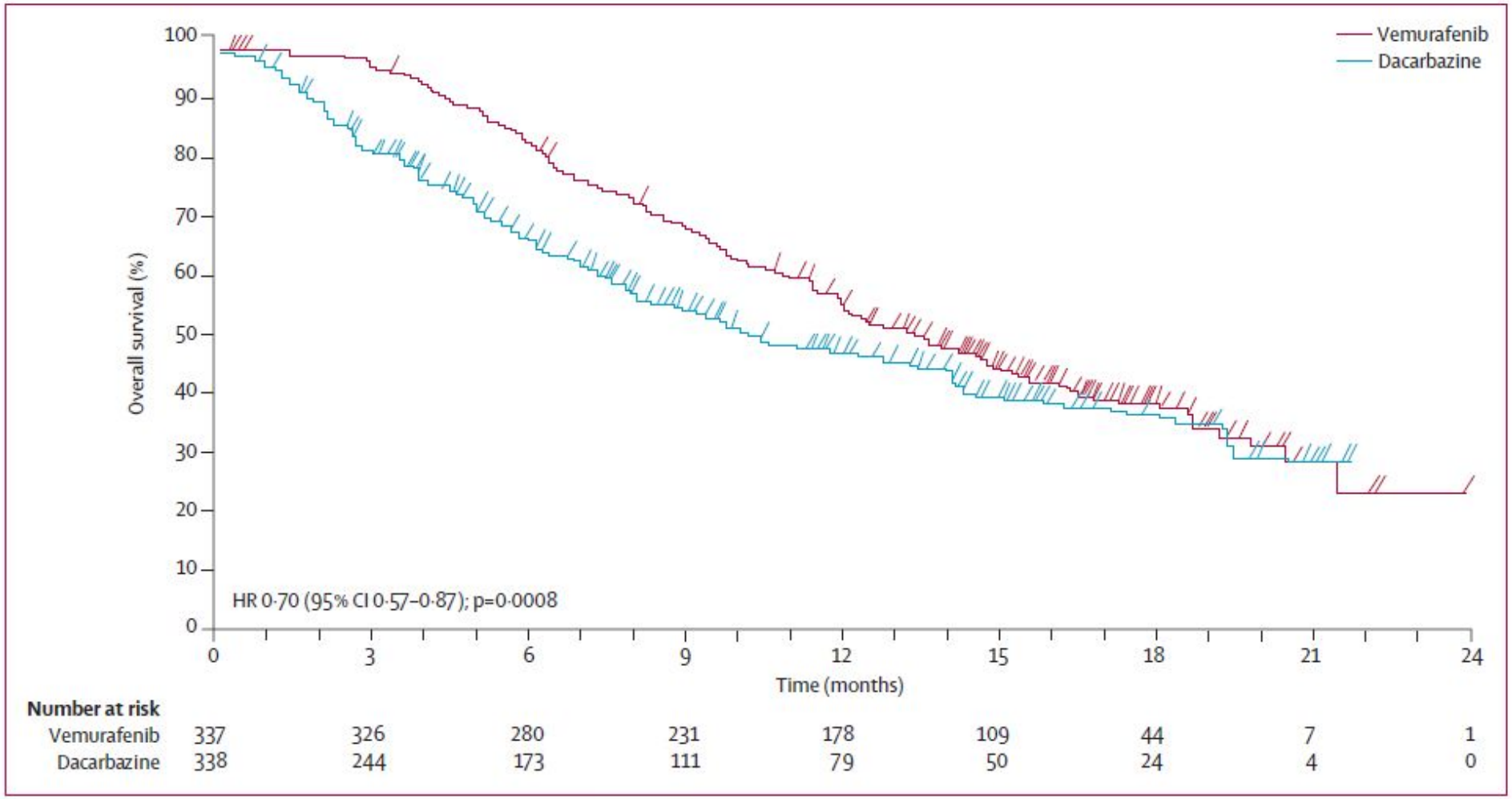
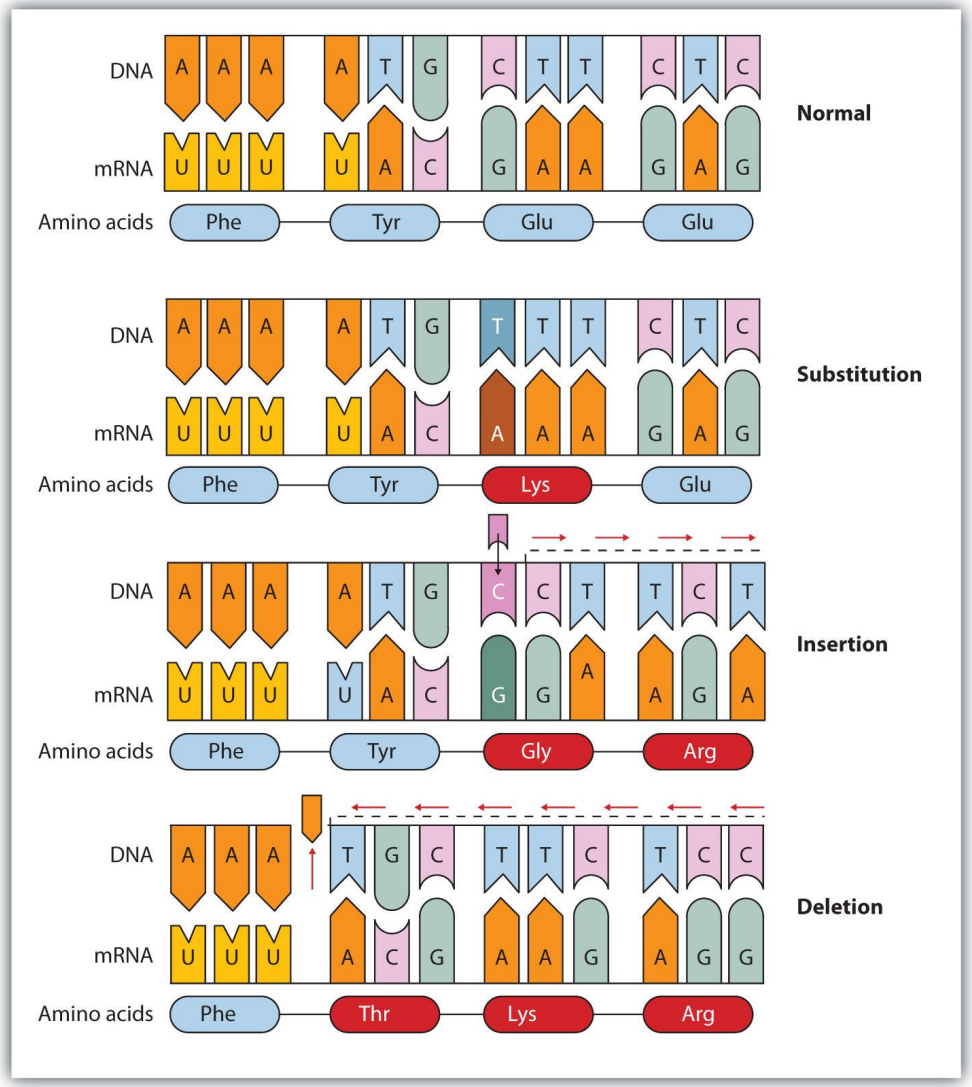


Figure 2: Overall survival (randomised population; censored at crossover) for patients randomly assigned to vemurafenib or to dacarbazine (cutoff Feb 1, 2012)

Chemistry and biology of point mutation



Disease	Responsible Protein or Enzyme
alkaptonuria	homogentisic acid oxidase
galactosemia	galactose 1-phosphate uridyl transferase, galactokinase, or UDP galactose epimerase
Gaucher disease	glucocerebrosidase
gout and Lesch-Nyhan syndrome	hypoxanthine-guanine phosphoribosyl transferase
hemophilia	antihemophilic factor (factor VIII) or Christmas factor (factor IX)
homocystinuria	cystathionine synthetase
maple syrup urine disease	branched chain α-keto acid dehydrogenase complex
McArdle syndrome	muscle phosphorylase
Niemann-Pick disease	sphingomyelinase
phenylketonuria (PKU)	phenylalanine hydroxylase
sickle cell anemia	hemoglobin
Tay-Sachs disease	hexosaminidase A
tyrosinemia	fumarylacetoacetate hydrolase or tyrosine aminotransferase
von Gierke disease	glucose 6-phosphatase
Wilson disease	Wilson disease protein

Three types of point mutations in DNA, The Basics of General, Organic, and Biological Chemistry

Different types of edit distance as a deterministic view of distance between two sequences

	Insertion	Deletion	Substitution	Transposition	Note
The Levenshtein distance	Allowed	Allowed	Allowed	Not allowed	
The longest common subsequence (LCS) distance	Allowed	Allowed	Not allowed	Not allowed	
The Hamming distance	Not allowed	Not allowed	Allowed	Not allowed	
The Damerau-Levenshtein distance	Allowed	Allowed	Allowed	Allowed (adjacent characters)	Not a distance metric, because triangle inequality is not satisfied
The Jaro-Winkler distance	Not allowed	Not allowed	Not allowed	Allowed	Not a distance metric

The Levenshtein distance models the biological process of point mutation most appropriately

The Levenshtein distance

Levenshtein distance: The minimum number of operations required to transform string a to string b with following operations:

- Insertion, for instance **bat** → **bait**
- Deletion, *e.g.* **boat** → **bot**
- Substitution, *e.g.* **pig** → **big**

The Levenshtein distance between two strings a, b of length $|a|$ and $|b|$ respectively is given by $\mathrm{lev}_{a,b}(|a|, |b|)$ where

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise, and $\mathrm{lev}_{a,b}(i,j)$ is the distance between the first i characters of a and the first j characters of b.

Calculate the Levenshtein distance with dynamic programming

- What is the Levenshtein distance between `ATGC` and `AGC`?

		A	T	G	C
A					
G					
C					

		A	T	G	C
	0	1	2	3	4
A	1	0	*1*	2	3
G	2	1	1	1	2
C	3	2	2	2	1

- Solution: 1

```
ATGC
A-GC
```

Calculate the Levenshtein distance with dynamic programming

- What is the Levenshtein distance between `ACTGCTT` and `ACATT`?

		A	C	T	G	C	T	T
	0	1	2	3	4	5	6	7
A	1	0	1	2	3	4	5	6
C	2	1	0	1	2	3	4	5
A	3	2	1	1	2	3	4	5
T	4	3	2	1	2	3	3	3
T	5	4	3	2	2	3	3	3

```
ACTGCTT          ACTGCTT          ACTGCTT
AC--ATT          ACA--TT          AC-A-TT
```

The Needleman-Wunsch algorithm uses dynamic programming for *global alignment* of two sequences

Compared with the Levenshtein distance, the Needleman-Wunsch algorithm uses biologically meaningful parameters to score insertion or deletion (gap penalty *d*), and substitution or mutation events (a substitution matrix *M*). The dynamic programming technique is used in a similar way.

Task: align two sequences `ATCGAC` and `CATAC`.

Parameters: d=-4, $M = \begin{pmatrix} & A & C & T & G \\ A & 5 & -3 & -3 & -3 \\ C & -3 & 5 & -3 & -3 \\ T & -3 & -3 & 5 & -3 \\ G & -3 & -3 & -3 & 5 \end{pmatrix}$

Solution:

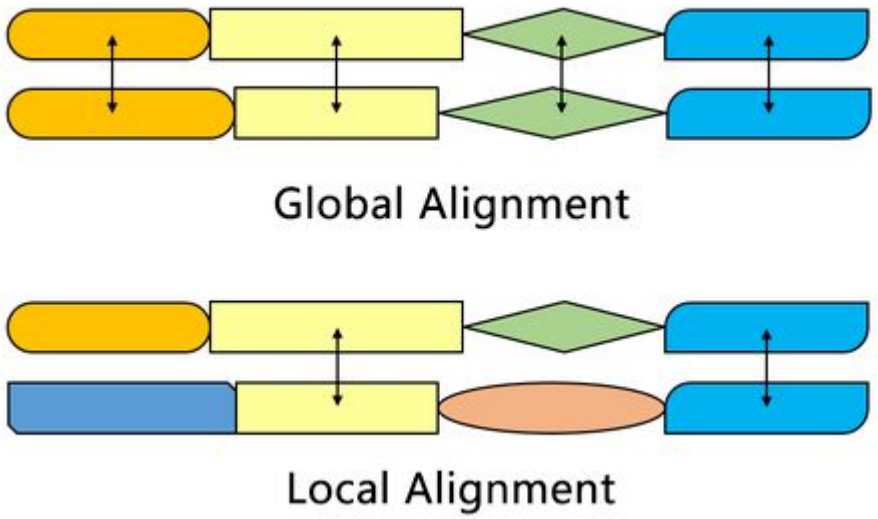
```
ATCGAC
||  ||
CAT--AC
```

-	-	A	T	C	G	A	C
-	0	-4	-8	-12	-16	-20	-24
C	-4	-3	-7	-3	-7	-11	-15
A	-8	1	-3	-7	-6	-2	-6
T	-12	-3	6	2	-2	-6	-5
A	-16	-7	2	3	-1	3	-1
C	-20	-11	-2	-1	0	-1	8

Source: https://commons.wikimedia.org/wiki/File:Needleman-wunsch.jpg

Check this video out if a step-by-step tutorial may help you: Link to YouTube (thanks to the contribution of Robert Do)

The Smith-Waterman algorithm uses dynamic programming for *local alignment* of two sequences



Yz CS5160, CC-BY SA 4.0

	Needleman-Wunsch	**Smith-Waterman**
Initialization	Gap penalty in first column and first row	0 in first column and first row
Scoring	Scores can be negative	Negative scores are set to 0
Traceback	Begin at the bottom right, and end at the top left cell.	Begin at the cell with the highest score, end when 0 is met.

Major differences between the Needleman-Wunsch and the Smith-Waterman algorithm. See this animation for an example.

Sequence alignment is fundamental for many bioinformatics tasks and tools

Examples:

- BLAST (Basic Local Alignment Search Tool) is used for almost all biological sequence analysis tasks. At its core, a heuristic algorithm approximates the Smith-Waterman algorithm for local alignment.
- Software tools such as Bowtie/Bowtie2 (Langmead *et al.*, Genome Biology, 2009; source code on GitHub), STAR (Dobin *et al.*, Bioinformatics, 2013; source code on GitHub) and GSNAP (web link) use more sophisticated methods to map sequencing reads (usually ~30-200 nucleotides) to large genomes (*e.g.* ~10^9 base pairs of mouse or human).

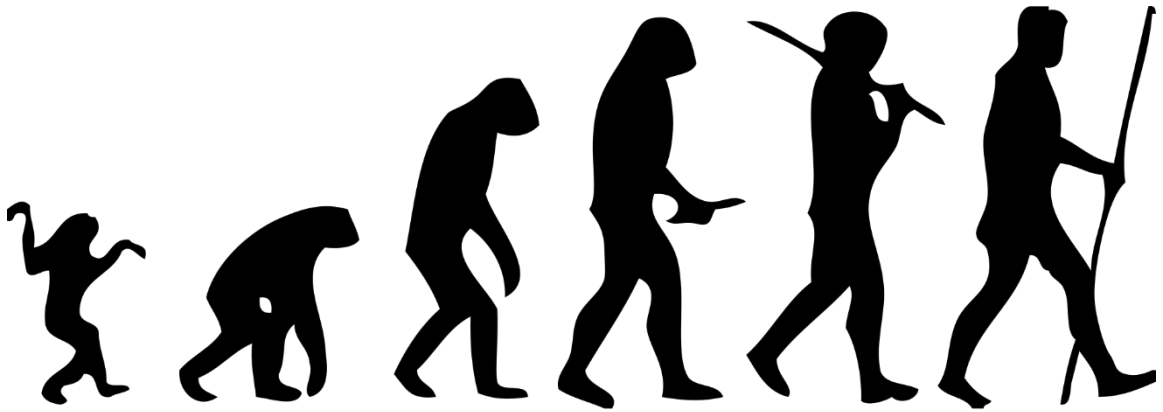
A typical case for Blast: we have a RNA sequence (see below). How can we know the original genome of the sequence, and ideally the gene encoding the sequences?

```
ATGTTTGTTTTTCTTGTTTTATTGCCACTAGTCT
CTAGTCAGTGTGTTAATCTTACAACCAGAACTCA
ATTACCCCCTGCATACACTAATTCTTTCACACGT
GGTGTTTATTACCCTGACAAAGTTTTCAGATCCT
CAGT
```

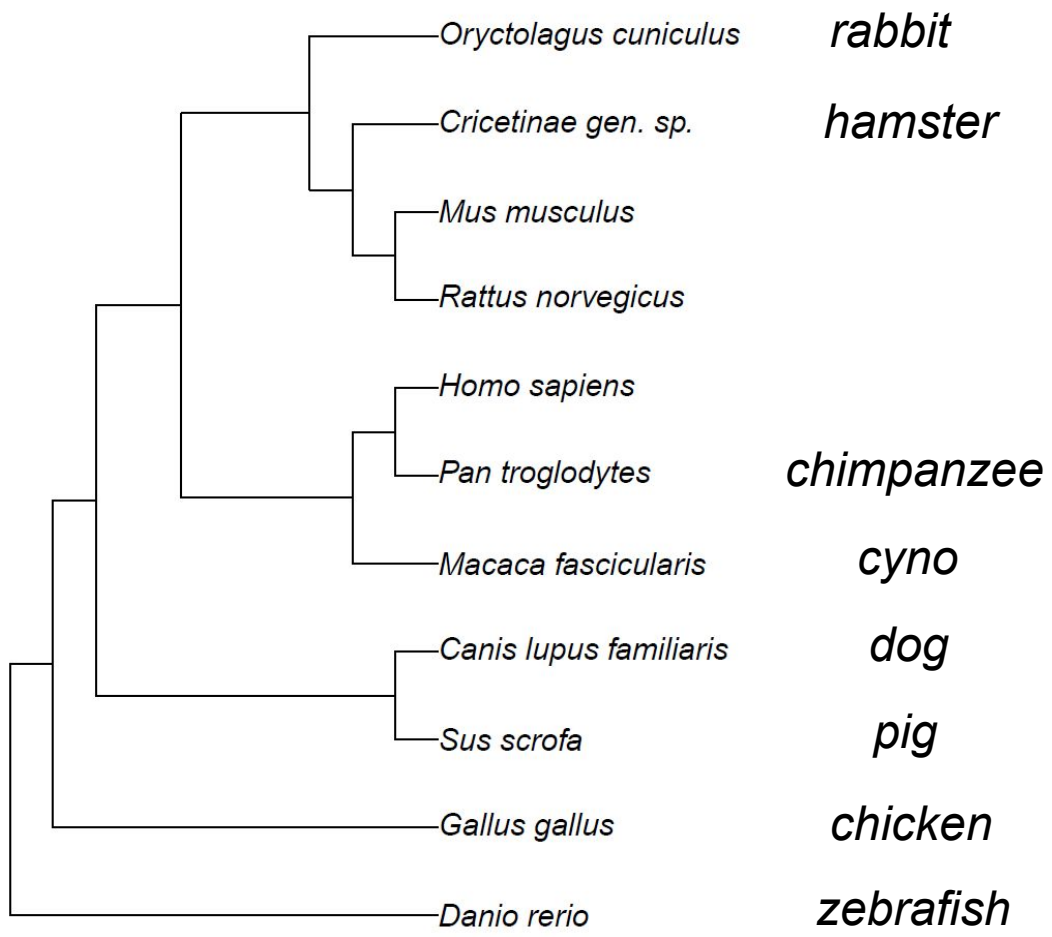
Tip: go to the NCBI BLAST tool (https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastn&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome), copy and paste the sequence as the query sequence, and try your luck. The default parameter would do.

The Wiki page of BLAST is a good start to understand how it works

Evolution: what is wrong with this figure?



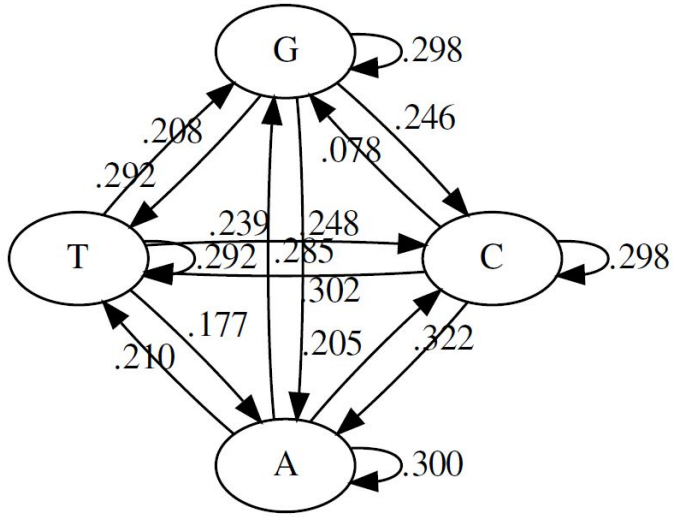
Phylogeny of commonly used species for animal studies



Tree structure retrieved from https://itol.embl.de/ (iTOL, Interactive Tree of Life), visualized with the *FigTree* software developed by Andrew Rambaut

A probabilistic view of biological sequence analysis with Markov chains

- A **discrete-time** Markov chain is a sequence of random variables with the Markov property, namely that the probability of moving to the next state depends only on the present state and not on the previous states.
- A Markov chain is often represented by either **a directed graph** or **a transition matrix**.
- Two sides of application
  - Given a string, assuming that the Markov chain model is suitable, we can easily construct a Markov chain, for instance by counting transitions and normalize the count matrix (variants possible).
  - Given a Markov chain model and a string, we can calculate the probability that the string is generated by the specific Markov chain model with the **chain rule of conditional probability.**

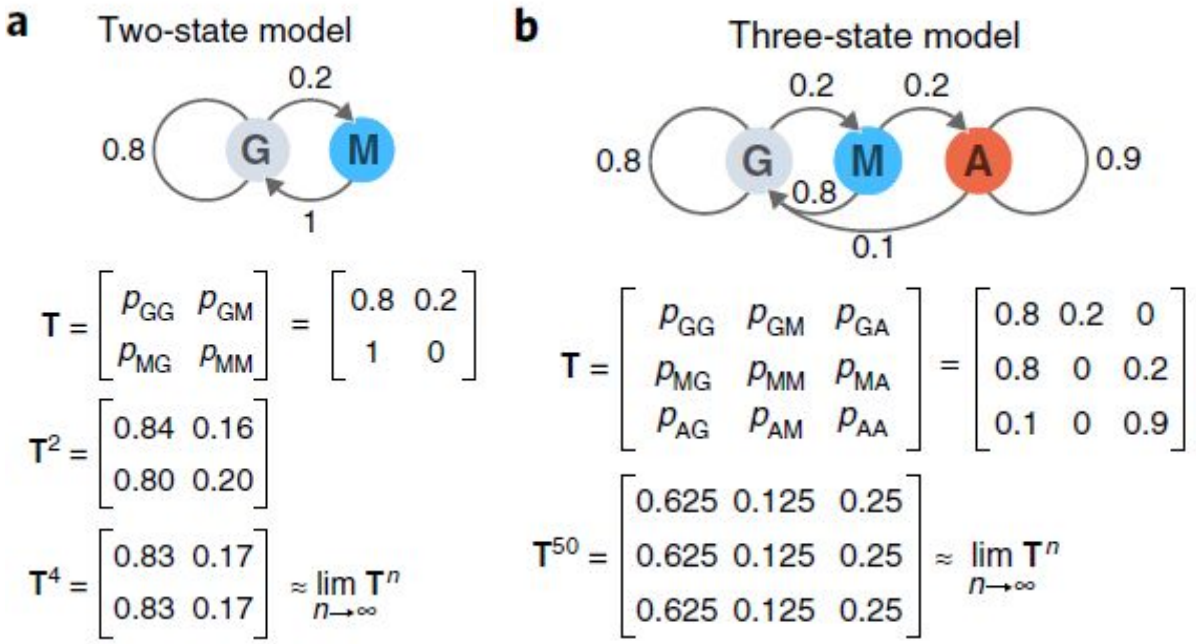


	A	C	G	T
A	.300	.205	.285	.210
C	.322	.298	.078	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292

Data: https://web.stanford.edu/class/stats366/exs/HMM1.html

Stationary distribution exist for ergodic (irreducible and aperiodic) Markov Chains

- A Markov Chain has stationary *n*-step transition probabilities, which are the *n*th power of the one-step transition probabilities. Namely, $P_n=P^n$.
- A stationary distribution π is a row vector whose entries are non-negative and sum to 1. It is unchanged by the operation matrix P on it, and is defined by $\pi P=\pi$.
  - In another word, it is the limit of the transition matrix multiplying itself.
  - Note that is has the form of the left eigenvector equation, $uA=\kappa u$, where κ is a scalar and u is a row vector. In fact, π is a normalised (sum to 1) multiple of a left eigenvector e of the transition matrix P with an eigenvalue of 1.
- Markov chains capture dependencies within a system and reveal interesting long-term behavior. They are subjects of the study of **stochastic processes.**

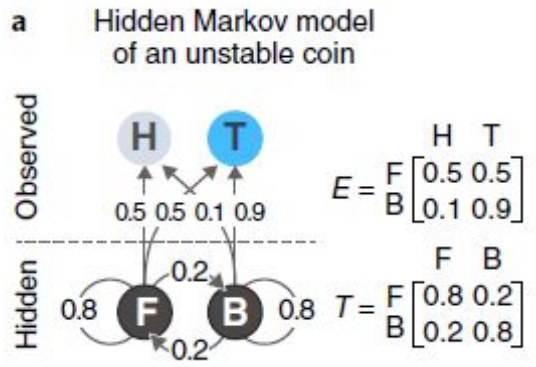


G=Growth, M=Mitosis, A=Arrest

Grewal, Jasleen K., Martin Krzywinski, and Naomi Altman. 2019. "Markov Models—Markov Chains." Nature Methods 16 (8): 663–64. https://doi.org/10.1038/s41592-019-0476-x.

Hidden Markov Chains

A Hidden Markov Model consists of two graphs (matrices): one of hidden states (corresponding to the transition matrix), and one of observed states (emitted by the hidden states according to the emission matrix). The Viterbi algorithm (based on dynamic programming), or the Baum-Welch algorithm (a special case of EM algorithms) is used to estimate its parameters.



A Hidden Markov Model of an unstable coin that has a 20% chance of switching between a fair state (F) and a biased state (B). Source: Grewal, Jasleen K., Martin Krzywinski, and Naomi Altman. 2019. "Markov Models — Hidden Markov Models." Nature Methods 16 (9): 795–96.

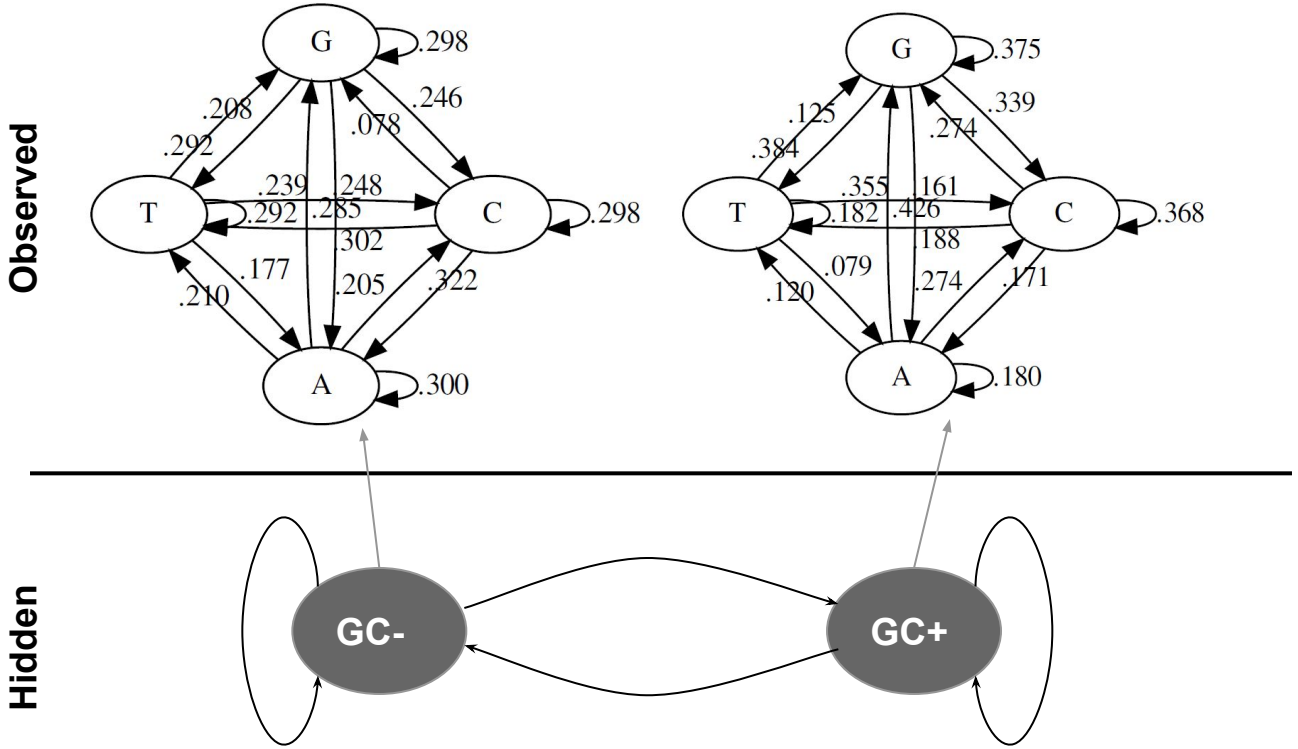
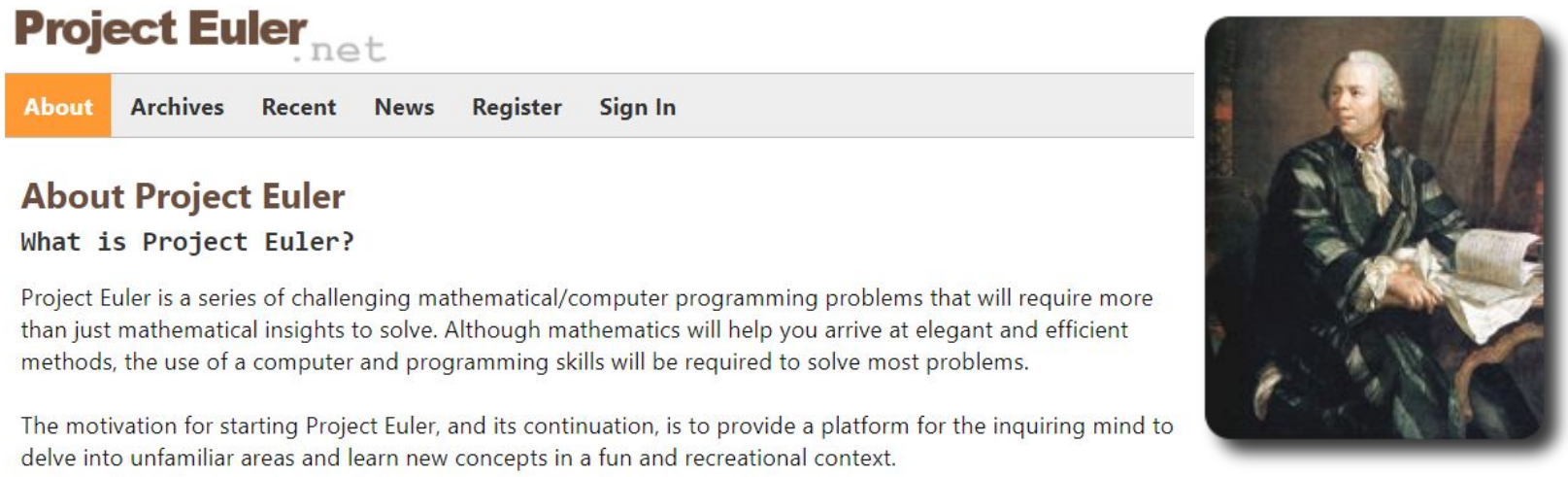


Illustration of a Hidden Markov Model predicting CpG islands in genomic sequences

Software tools

- **General biological sequence analysis**
  - EMBOSS software suite: http://emboss.sourceforge.net/, also available online at European Bioinformatics Institute (EBI): https://www.ebi.ac.uk/services
  - BLAST (=Basic Local Alignment Search Tool) can be run at many places, for instances from EBI and National Center for Biotechnology Information (NCBI): https://blast.ncbi.nlm.nih.gov/Blast.cgi
  - Programming access, for instance the Biopython project: https://biopython.org
- **RNA biology**
  - ViennaRNA package (https://www.tbi.univie.ac.at/RNA/)
  - RNA processing tools available at U Bielefeld, for instance RNAhybrid, which finds minimum free energy hybridization using dynamic programming (https://bibiserv.cebitec.uni-bielefeld.de/rnahybrid)
- **Profile Hidden Markov Models (HMMs)**
  - The HMMER package: http://hmmer.org/

The Euler Project



https://projecteuler.net/

- **Learning by problem-solving**
- **Free**
- **Math + CS**

Problem 1: Multiples of 3 and 5

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

Rosalind: a great scientist, and a platform for learning bioinformatics and programming through problem solving

Rosalind Elsie Franklin

1920-1958

ROSALIND http://rosalind.info/problems/locations/

→

A Rapid Introduction to Molecular Biology click to expand

Problem

A **string** is simply an ordered collection of symbols selected from some **alphabet** and formed into a word; the **length** of a string is the number of symbols that it contains.

An example of a length 21 **DNA string** (whose alphabet contains the symbols 'A', 'C', 'G', and 'T') is "ATGCTTCAGAAAGGTCTTACG."

Given: A DNA string s of length at most 1000 nt.

Return: Four integers (separated by spaces) counting the respective number of times that the symbols 'A', 'C', 'G', and 'T' occur in s.

Sample Dataset

```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
```

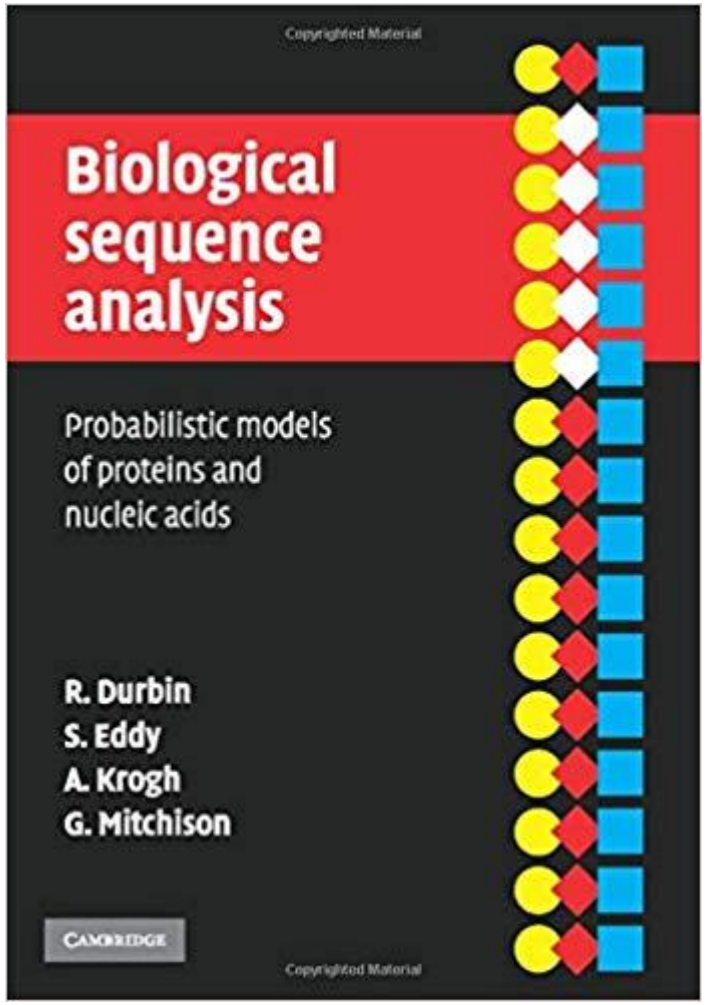
Sample Output

```
20 12 17 21
```

Please login to solve this problem.

Further resources

***Biological Sequence Analysis* by Durbin, Eddy, Krogh, and Mitchison**



Teaching RNA algorithms by the Backofen Lab at U Freiburg, with source codes available on GitHub.

The website hosts among others an interactive tool to visualize how dynamic programming (DP) helps to predict RNA secondary structure.

For a gentle introduction, see also *How Do RNA Folding Algorithms Work?* by Eddy, Sean R, *Nature Biotechnology* 22, Nr. 11 (November 2004): 1457–58. https://doi.org/10.1038/nbt1104-1457.

An Introduction to Applied Bioinformatics by Greg Caporaso (NAU)

The tutorial is written in Python using Jupyter. It introduces concepts in (a) pairwise sequence alignment, (b) sequence homology searching, (c) generalized dynamic programming for multiple sequence alignment, (d) phylogenetic reconstruction, (e) sequence mapping and clustering, as well as (f) machine learning in bioinformatics. Applications and exercises are also available.

The Coursera course ***Algorithms on Strings*** **by UC San Diego** may be interesting if you are interested in algorithms on strings**.**

Summary and Q&A

- Biological sequence analysis is used throughout the drug discovery process, and is essential for molecular modelling in the multiscale-modelling view of drug discovery.
- We explored both deterministic views of sequence analysis, with the example of Levenshtein distance, and probabilistic views, with the example of Markov chains and hidden Markov chains.
- Mathematical techniques such as dynamic programming, when implemented as algorithms software tools, are important for many tasks. We discussed in particular the Needleman-Wunsch algorithm, the Smith-Waterman algorithm, the BLAST software, and sequencing read alignment tools such as Bowtie2, STAR, and GSNAP.
- We provided further resources for further study and exploration.

Offline activities

- Read Tsai et al. (2008) ("Discovery of a Selective Inhibitor of Oncogenic B-Raf Kinase with Potent Antimelanoma Activity." PNAS 105 (8): 3041–46, which can be downloaded here) and answer questions (see the next slide). Please submit your results to the Google Form, the link of which will be sent via a separate email.
- Optional and recommended:
  - Exercises in the Handout of the Lecture 2.
  - Fill the anonymous survey #3 (link will be sent via a separate email).
  - Beyond bioinformatics, edit distance is often used in computational linguistics and natural language processing. For instance, check out *How to Write a Spelling Corrector* by Peter Norvig.

Questions on the PNAS paper by Tsai *et al.*

1. **(a) How many** compounds were screened? (b) What information is available about their **properties**?
2. **How** were the compounds **screened**?
3. What was the **initial chemical structure** that was found to bind to the ATP-binding site?
4. By overlapping structures, the team aimed to optimizing what **two properties of the compounds**?
5. What types of compounds were tested in the **subsequent screening**?
6. What properties does the PLX4720 compound have that make it **particularly attractive** as a drug?

Backup slides

Continuous-time Markov Chains

- Continuous-time Markov Chains are used for **phylogenetic analysis**, for instance of orthologous genes and of bacterial/viral genomes. They satisfy the Markovian property: $P(t+\tau)=P(t)P(\tau)$.
- The process makes a transition from the current state *i* after an amount of time modelled by an *exponential random variable* E_i, known as the **holding time**. Random variables of each state is independent.
- When a transition is made, the process moves according to the **jump chain**, a discrete-time Markov chain with a transition matrix.
- If there are *n* states, then at the time of transition, there are *n-1* competing exponentials. Since the distribution of the minimum of exponential random variables is also exponential, the continuous-time Markov chain changes its state from *i* according to a parameter $E_{i,j} \sim \text{Exp}(q_{i,j})$ ($i \neq j$). The parameters are known as the Q-matrix, or the **rate matrix**. The transition rate *E* is the product of holding time and the transition probability.
- Whereas the row sums of a transition matrix are always 1, the row sums of a rate matrix are always zero.

Given the transition matrix

$$P(t) = \begin{pmatrix} p_{AA}(t) & p_{AG}(t) & p_{AC}(t) & p_{AT}(t) \\ p_{GA}(t) & p_{GG}(t) & p_{GC}(t) & p_{GT}(t) \\ p_{CA}(t) & p_{CG}(t) & p_{CC}(t) & p_{CT}(t) \\ p_{TA}(t) & p_{TG}(t) & p_{TC}(t) & p_{TT}(t) \end{pmatrix}$$

We model the probability of seeing the same alphabet after a small increment of time as the sum of the starting probability, minus its loss, and plus its gain

$$\mu_x = \sum_{y \neq x} \mu_{xy}$$

$$p_A(t + \Delta t) = p_A(t) - p_A(t)\mu_A \Delta t + \sum_{x \neq A} p_x(t)\mu_{xA}\Delta t\,.$$

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \mathbf{p}(t)Q\Delta t\,,$$

where

$$Q = \begin{pmatrix} -\mu_A & \mu_{AG} & \mu_{AC} & \mu_{AT} \\ \mu_{GA} & -\mu_G & \mu_{GC} & \mu_{GT} \\ \mu_{CA} & \mu_{CG} & -\mu_C & \mu_{CT} \\ \mu_{TA} & \mu_{TG} & \mu_{TC} & -\mu_T \end{pmatrix}$$

Source: https://en.wikipedia.org/wiki/Models_of_DNA_evolution