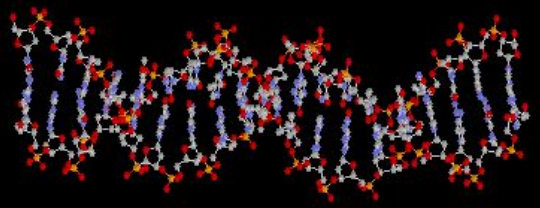
Biotech is the *future*.

constructive propaganda



Desoxyribo-Nucleic Acid

ATGCCGTACGTACTGACGTGTCGATCGTAG

blueprint instructions

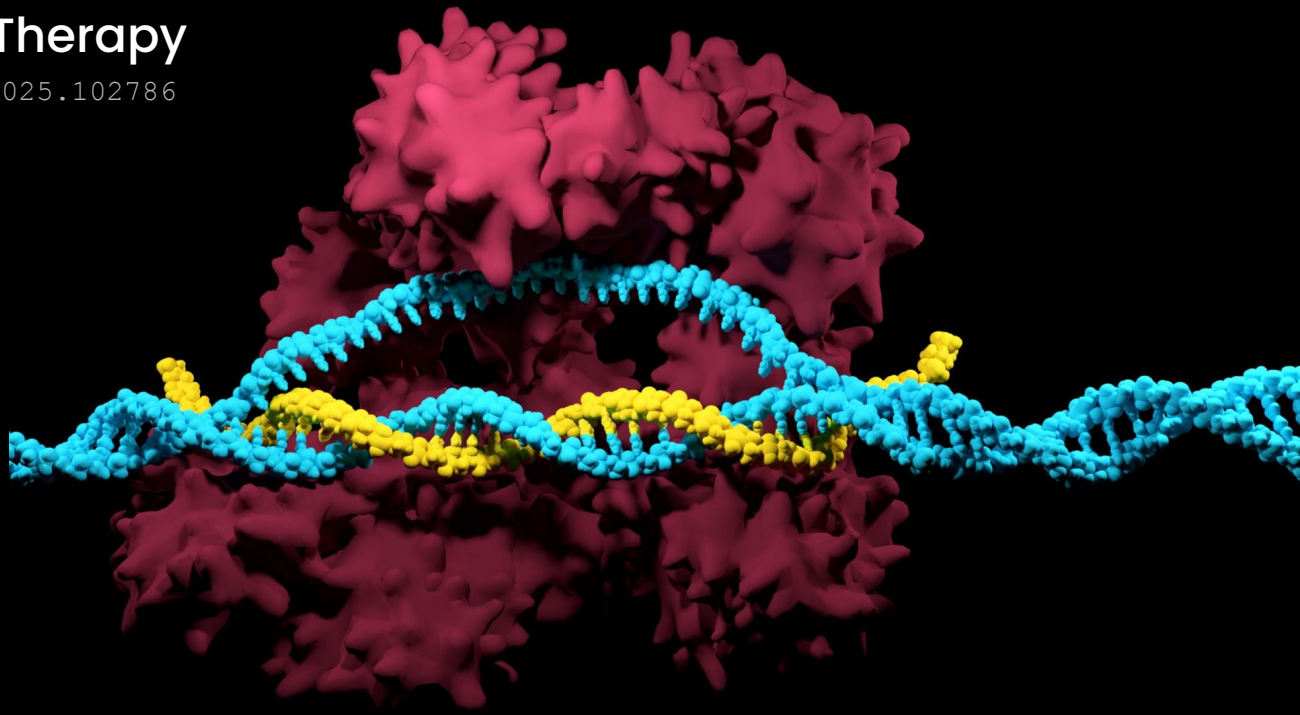
3-Dimensional Protein

MEEPQSDPSVEPPLSQETFSDLWKLLPE

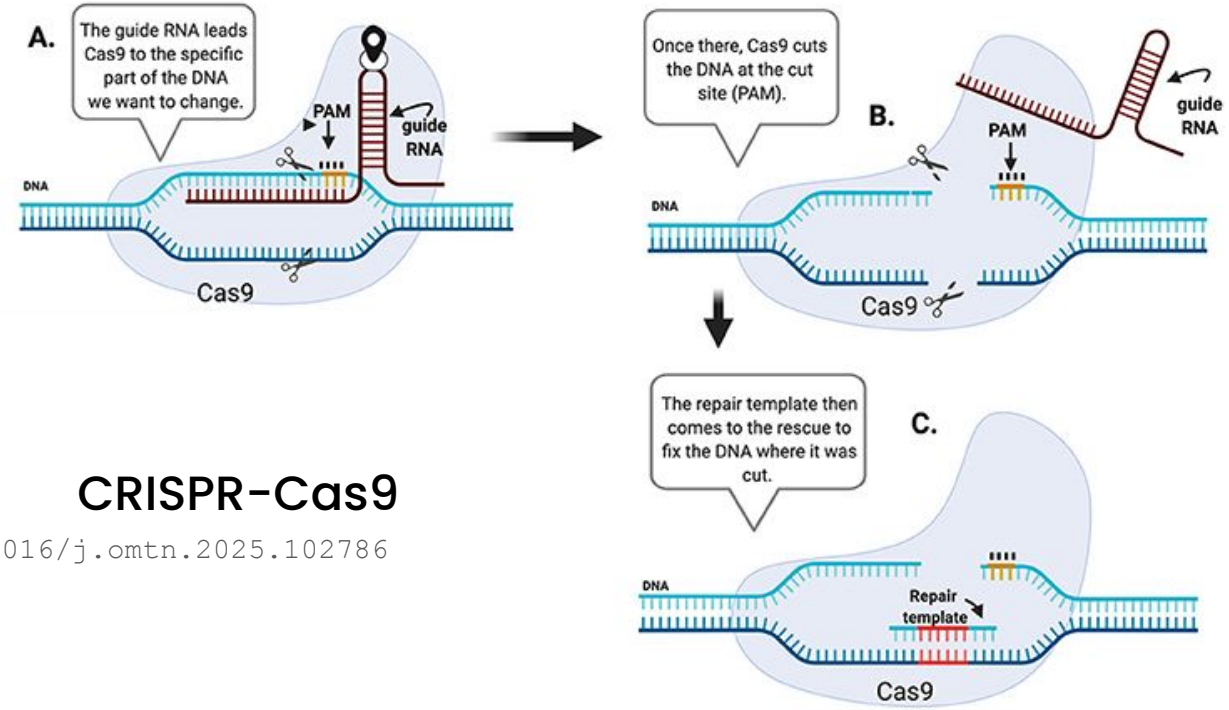
workers and structure

CRISPR-Cas9 Therapy

10.1016/j.omtn.2025.102786

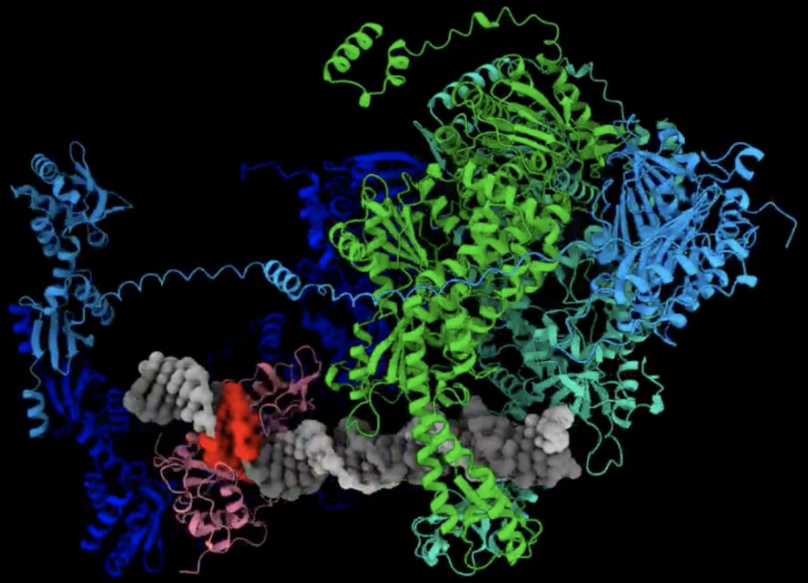


8.5Md USD market (2026)
used for SCD treatment



CRISPR-Cas9

10.1016/j.omtn.2025.102786

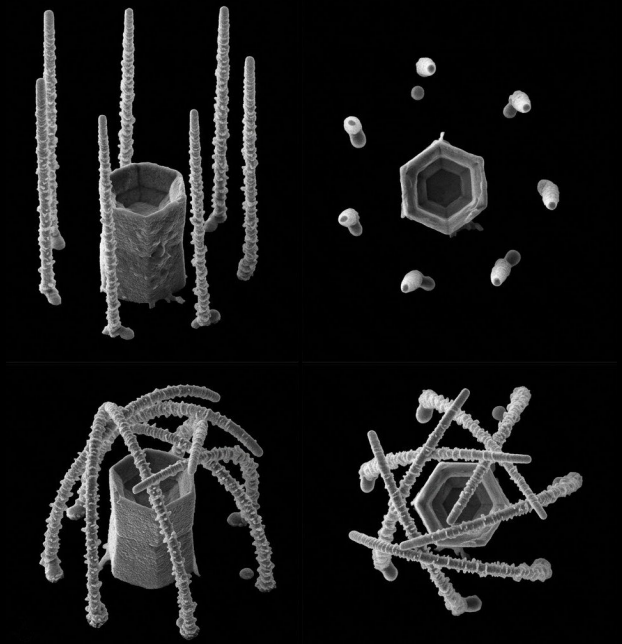


AlphaFold Version 3

10.1038/s41586-024-07487-w

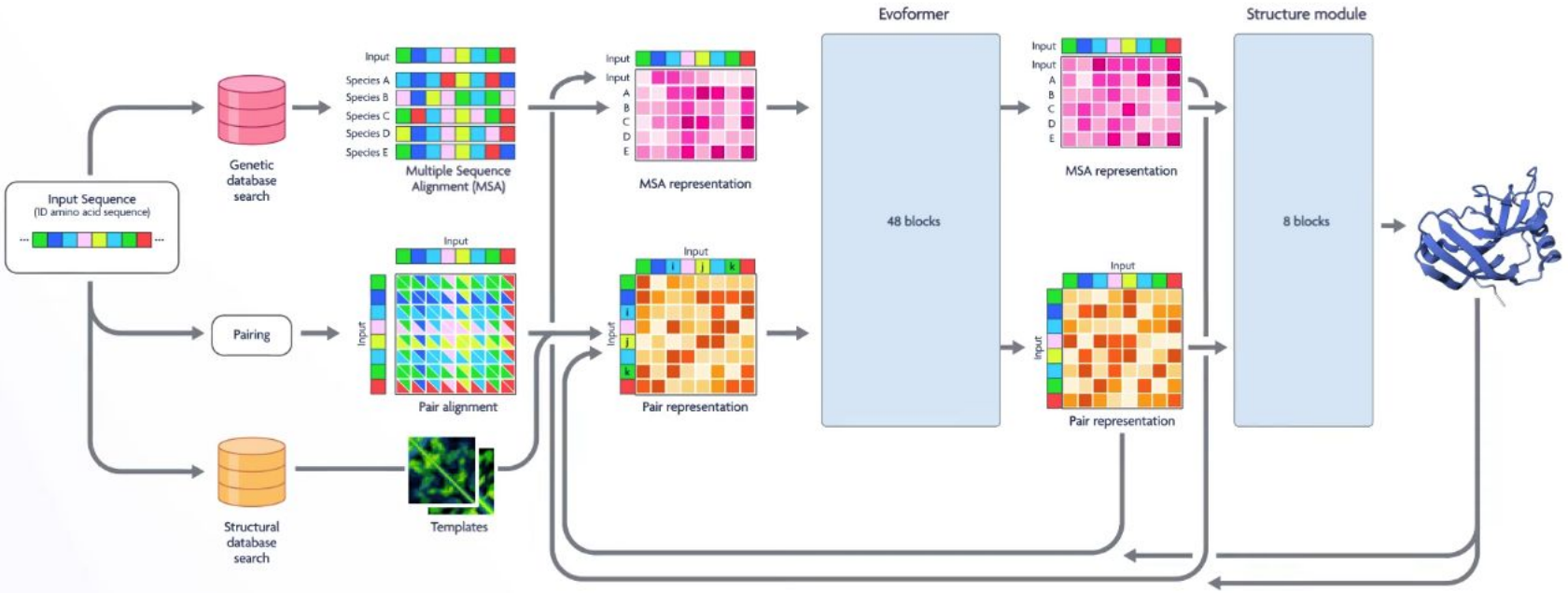
RFdiffusion Nanocages

10.64898/2026.03.04.709581

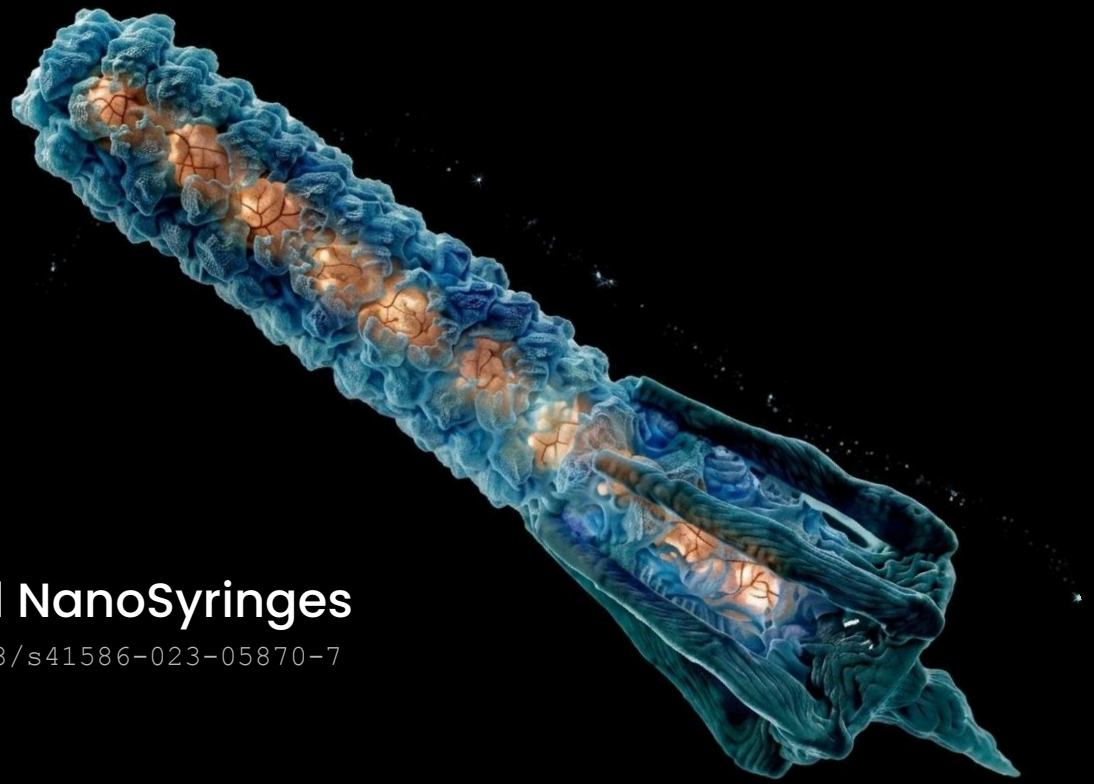


AlphaFold Version 2

10.1038/s41586-021-03819-2



destroy human tumors
100% specificity
can deliver CRISPR

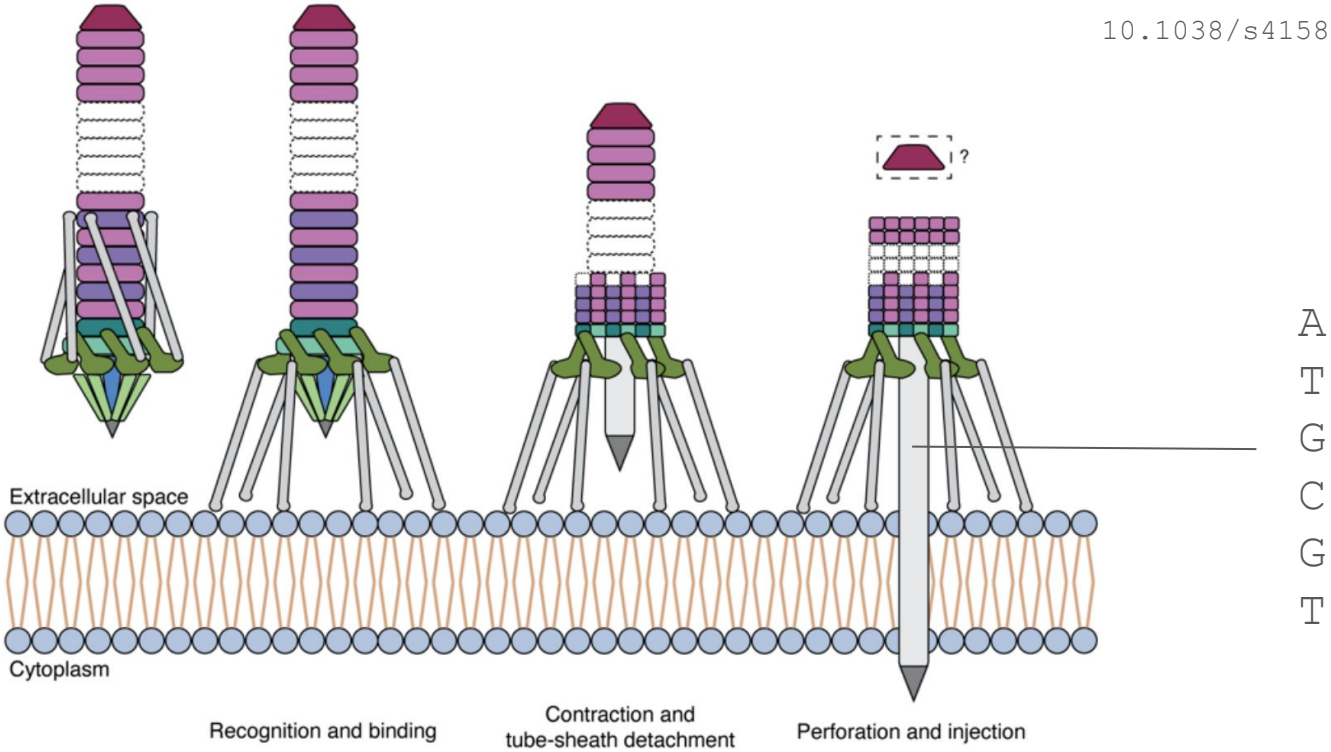


Bacterial NanoSyringes

10.1038/s41586-023-05870-7

NanoSyringe

10.1038/s41586-023-05870-7



Questions ?

synthetic biology

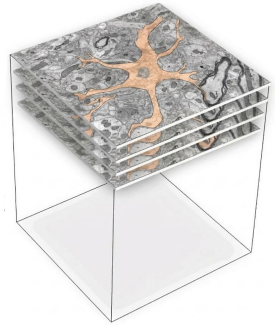
Worm Full Simulation

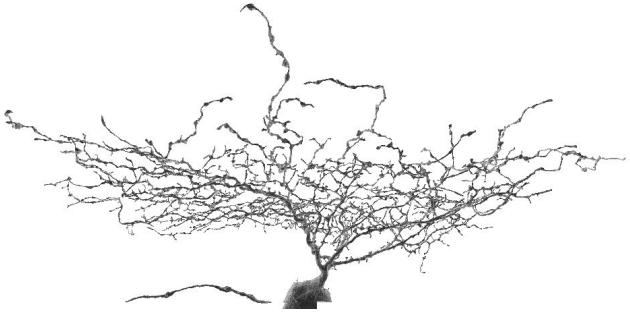
https://openworm.org/

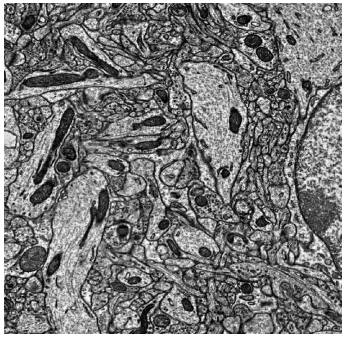
```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

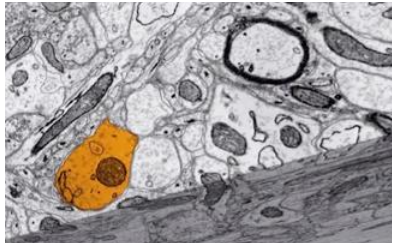
Eyewire Pipeline

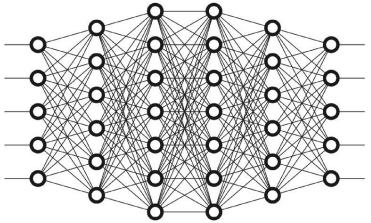
https://seunglab.org









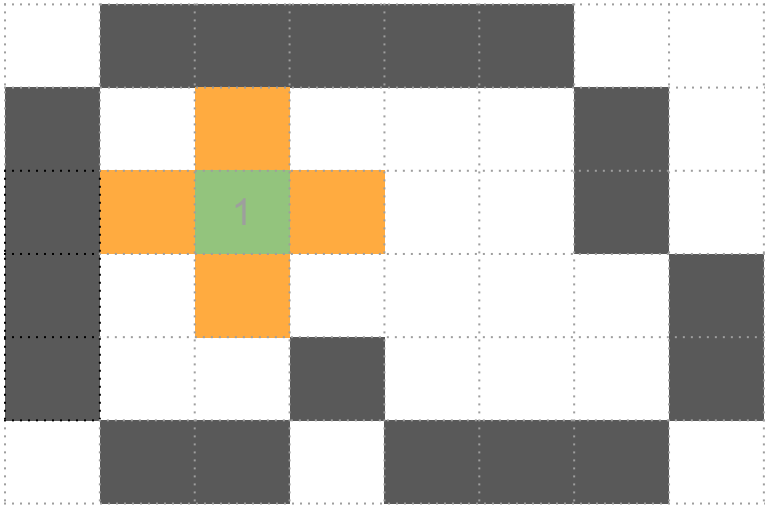


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

step 1

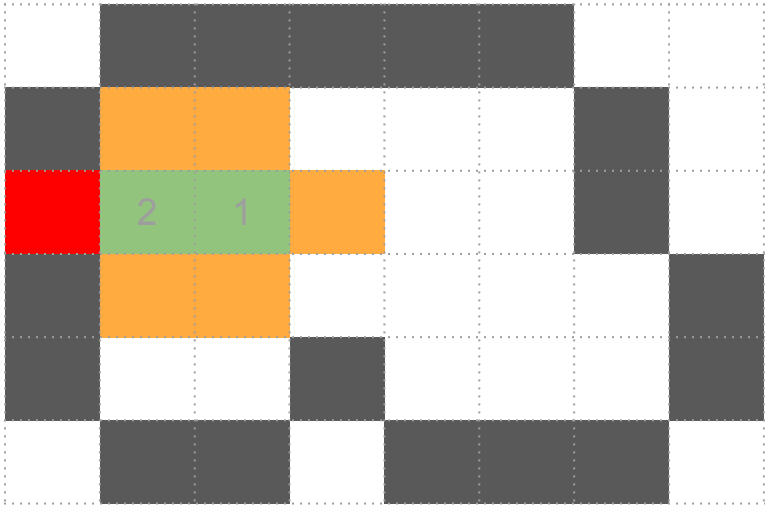


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

step 2

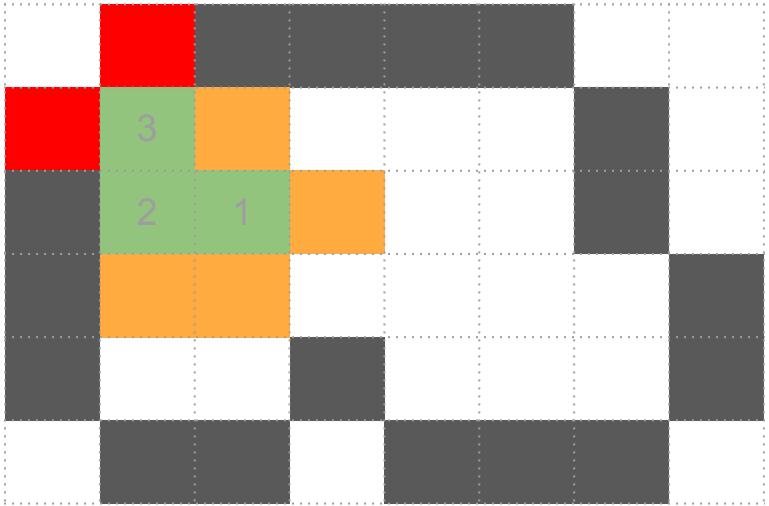


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

step 3

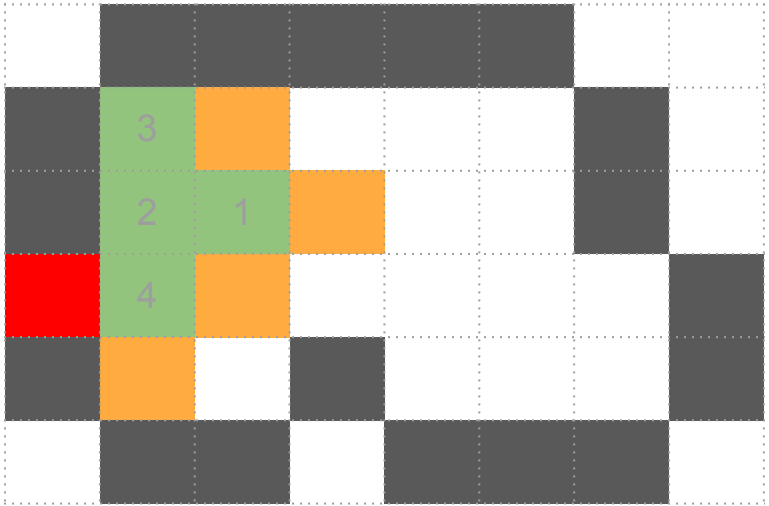


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

step 4

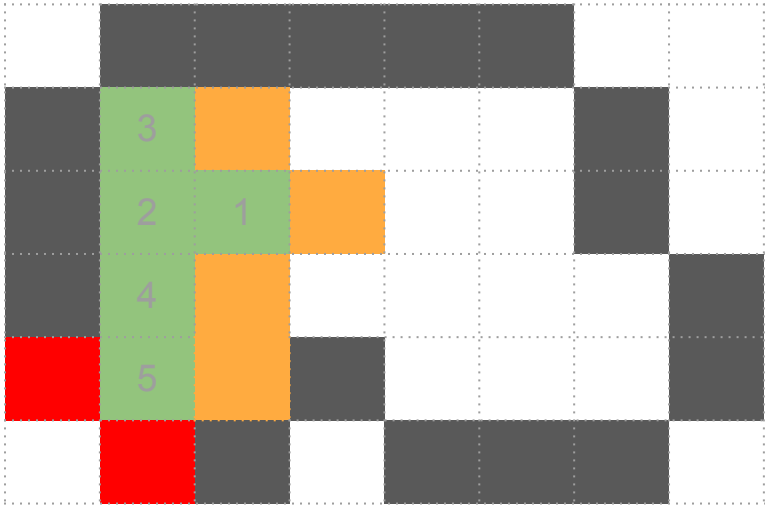


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

step 5

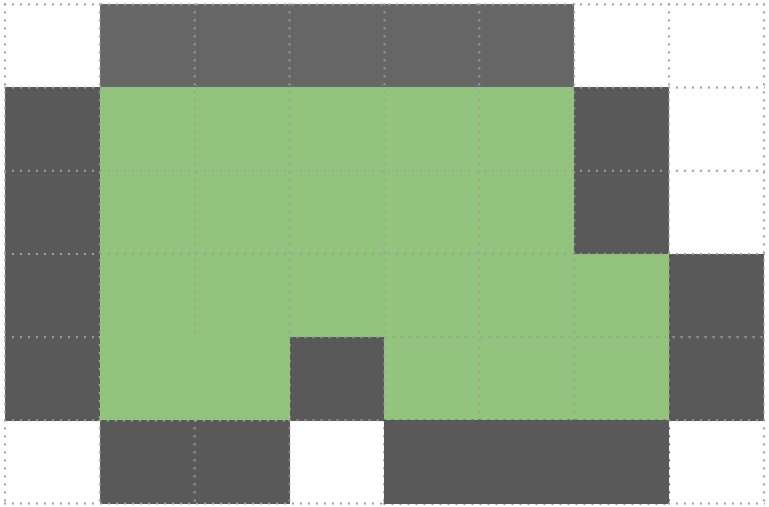


FloodFill Algorithm

https://en.wikipedia.org/wiki/Flood_fill

```
void floodFill(int x, int y, int fill, int old)
{
    if ((x < 0) || (x >= width)) return;
    if ((y < 0) || (y >= height)) return;
    if (getPixel(x, y) == old) {
            setPixel(fill, x, y);
            floodFill(x+1, y, fill, old);
            floodFill(x, y+1, fill, old);
            floodFill(x-1, y, fill, old);
            floodFill(x, y-1, fill, old);
    }
}
```

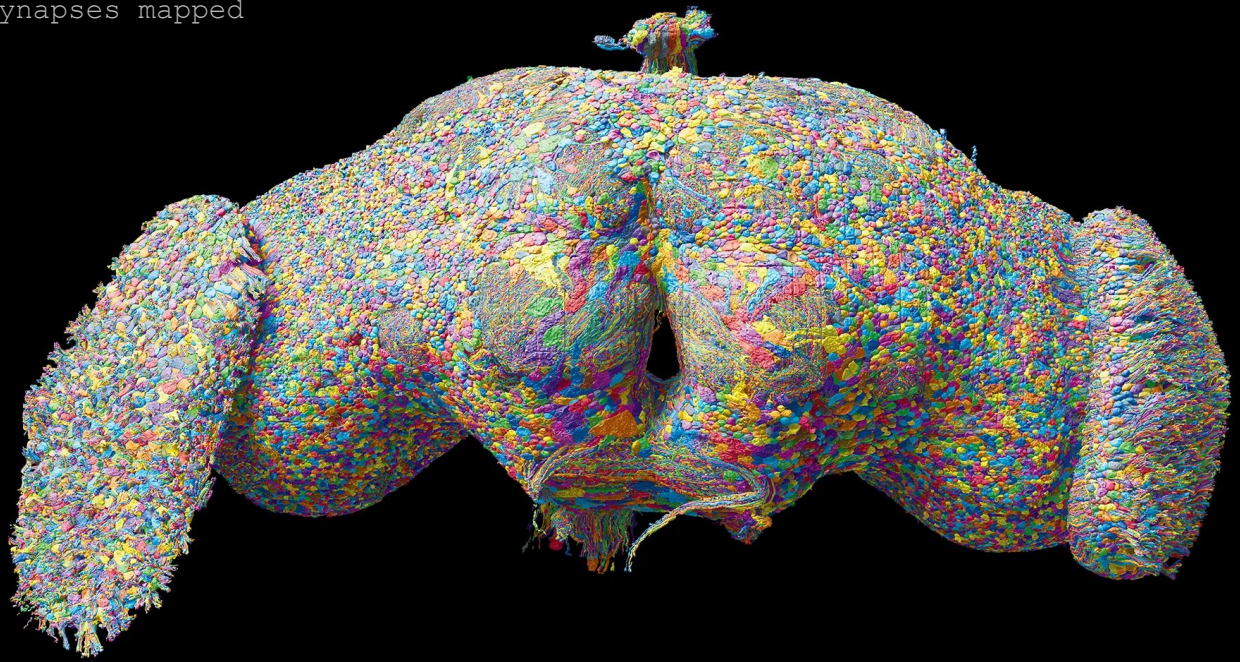
step 6

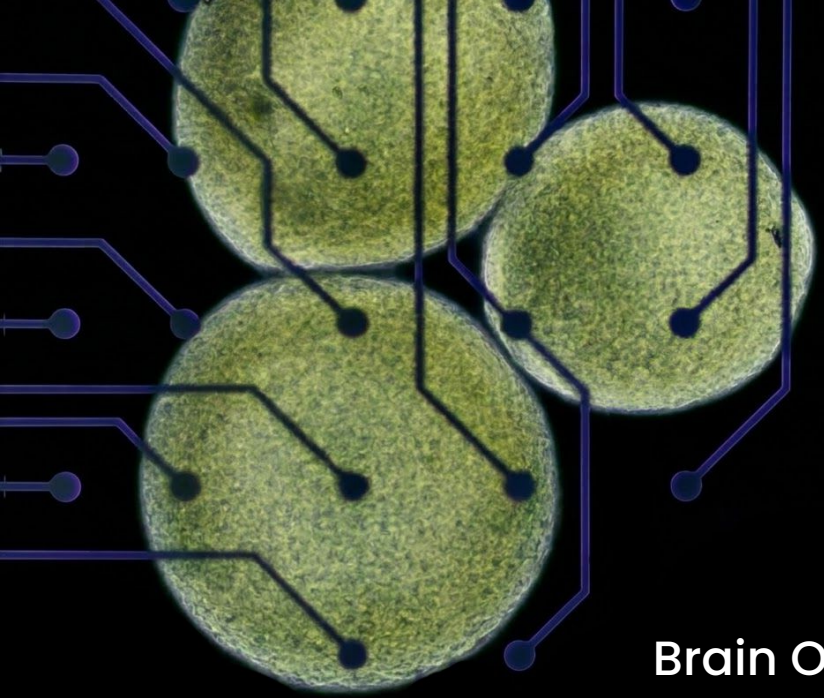


Drosophila Connectome

https://flywire.ai

139255 neurons
>50M synapses mapped





Brain Organoids

10.1016/j.neuron.2022.09.001

can learn pong <5mins
more plasticity than AI

Running Butterfly

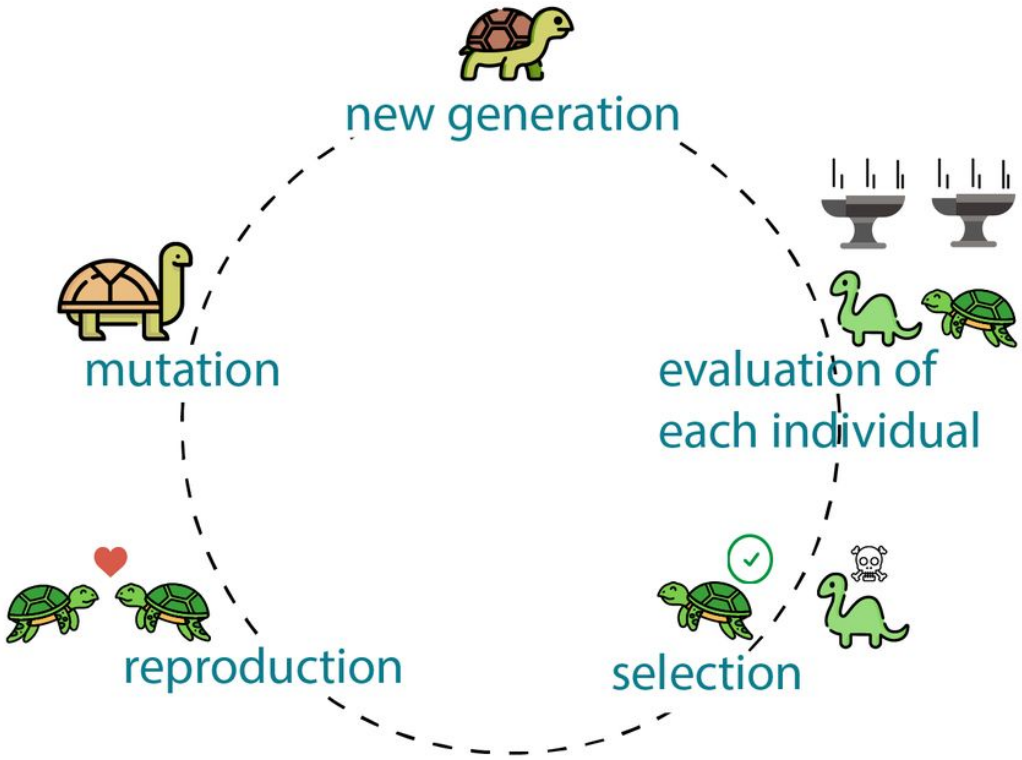
https://finalspark.com/

Questions ?

computational neuroscience

Genetic Algorithm

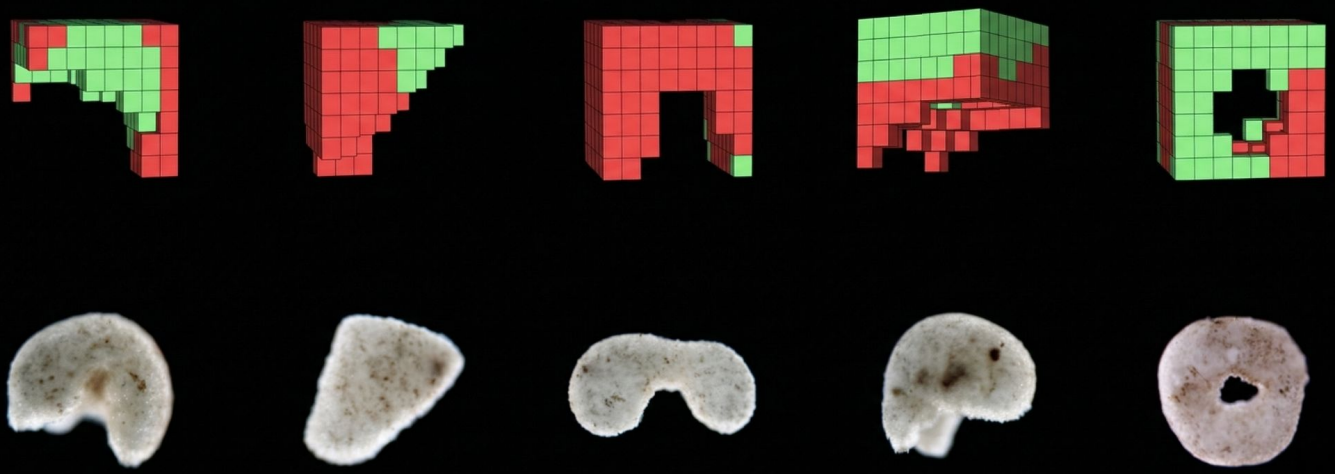
https://wikipedia.org



genetic algorithm search
kinematic auto-movement

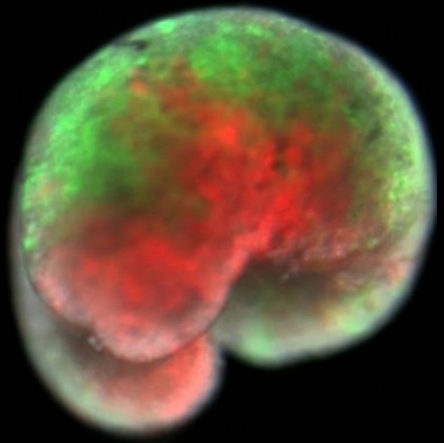
Xenobots Selection

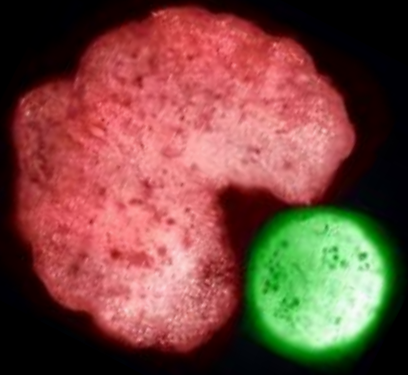
10.1073/pnas.1910837117



Programmable Xenobots

10.1073/pnas.1910837117





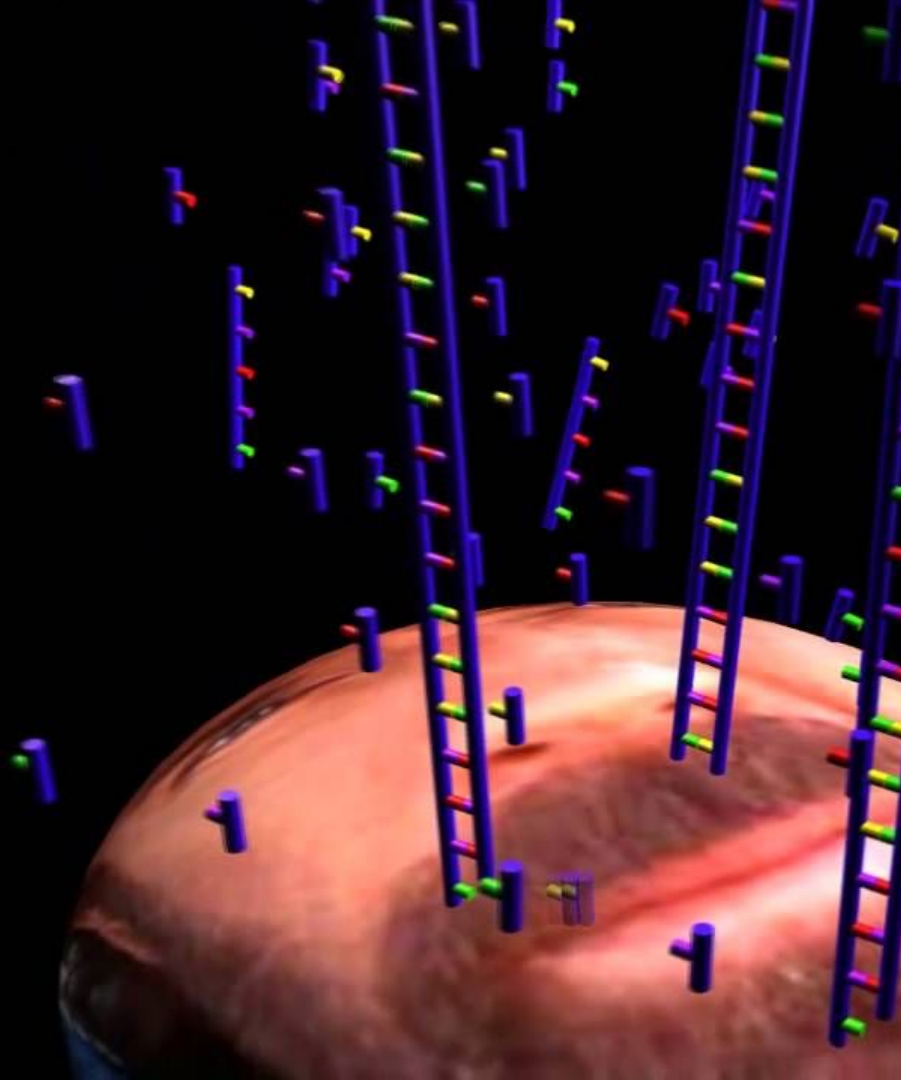
kinematic self-replication
can repair human neurons

Questions ?

developmental engineering

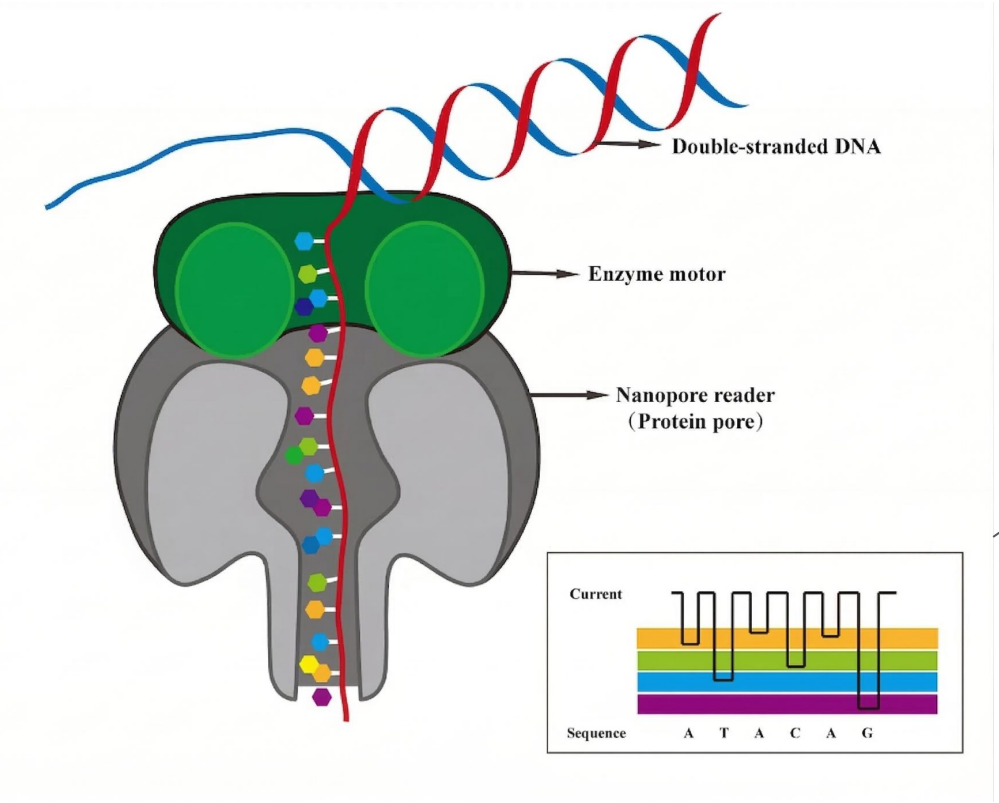
Next Gen Sequencing

10.1038/nature07517



Nanopore Sequencing

https://nanoporetech.com



A
T
G
C
G
T

TAGCTATCGTAGTGAT

TAGCTAT___AGTGAT

TAGCTATCGTAGTGAT

TAGACTATCGTAGTGAT

TAGCTATCGTAGTGAT

TAGCTATCGTAGGTAT

Sequence Alignment

10.1016/0022-2836(70)90057-4

find twin sequences
billion DNA entries

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

Needleman–Wunsch

https://wikipedia.org

x compare “G” et “GCA”

x compare “GATT” et “GC”

x le nombre que l’on cherche

Le nombre affiché est le nombre minimal de modifications à faire pour passer de l’un à l’autre

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

Needleman–Wunsch

https://wikipedia.org

Comparaison entre G et G :

- Score de "G" et "" + 1 car il suffit de rajouter G
- Score de "" et "G" + 1 car il suffit de supprimer G
- Score de "" et "" +
  - 0 si les derniers caractères sont égaux
  - 1 sinon

=> **3ème option dans notre cas**

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

Needleman–Wunsch

https://wikipedia.org

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

Needleman–Wunsch

https://wikipedia.org

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

Needleman–Wunsch

https://wikipedia.org

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

Needleman–Wunsch

https://wikipedia.org

GCA_TGCG

G_ATTACA

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

Needleman–Wunsch

https://wikipedia.org

```
GCAT_GCG
G_ATTACA
```

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

Needleman–Wunsch

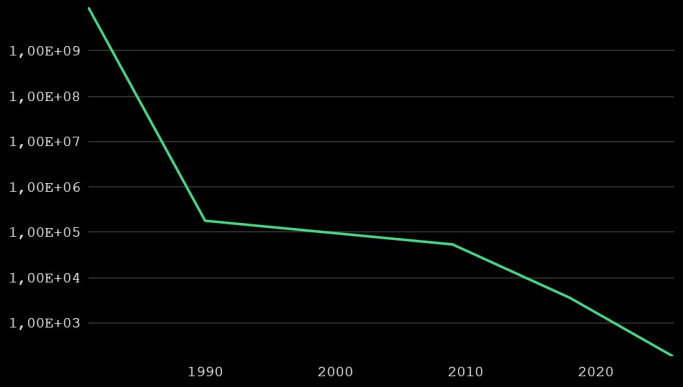
https://wikipedia.org

GCATG_CG

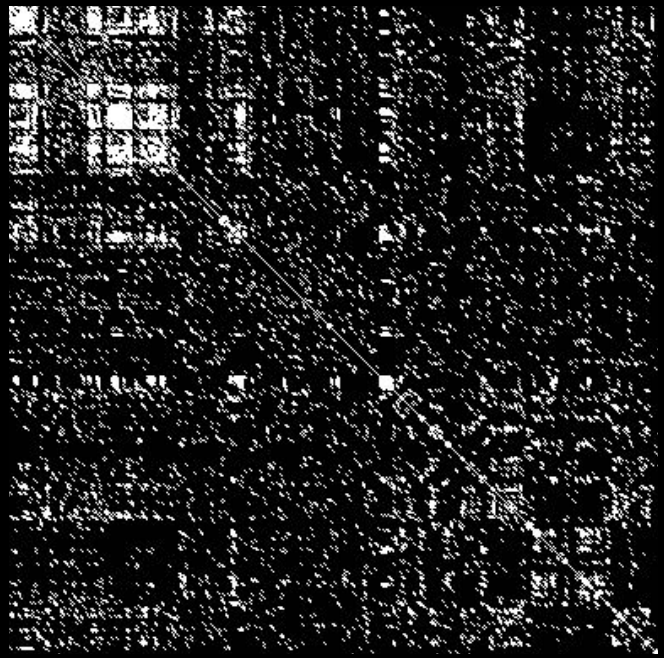
G_ATTACA

Instant DNA Alignment

10.1016/0022-2836(70)90057-4



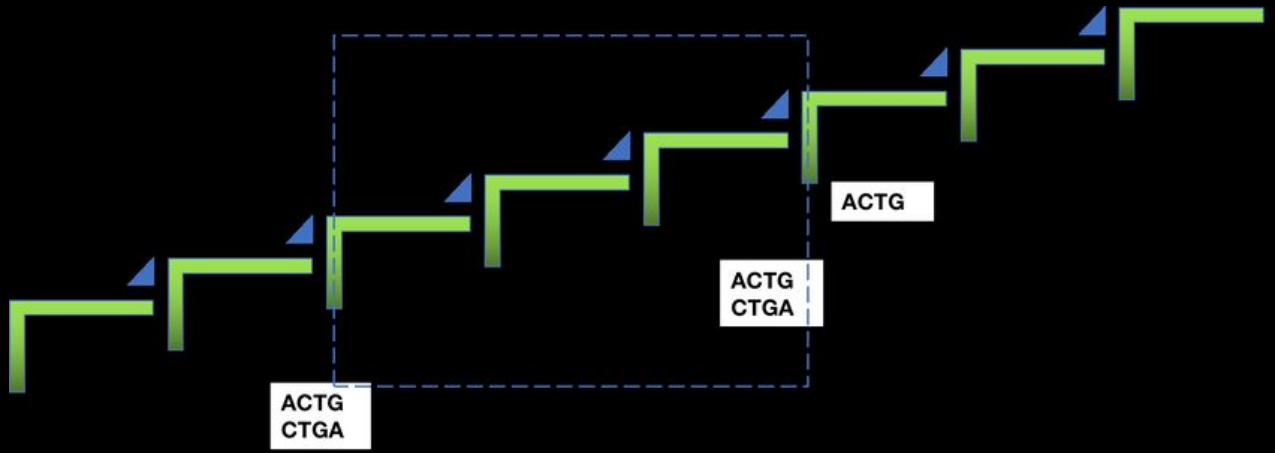
genome alignment time (seconds)



215000 To/g (x484000)
1000 years, no cooling
283M USD market (2026)

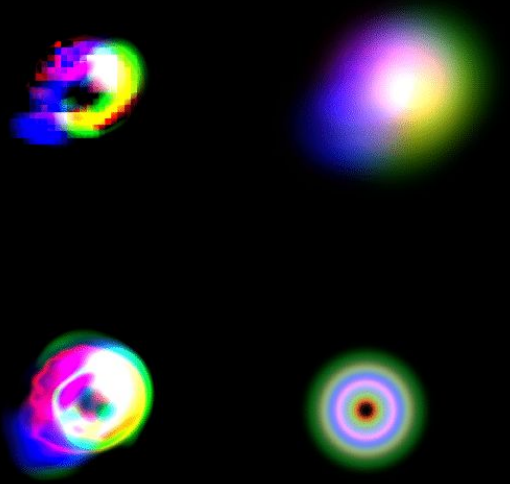
DNA Data Storage

10.1126/science.1226355



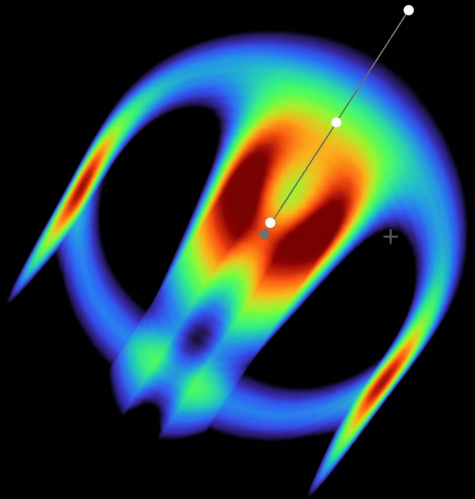
Questions ?

applied bioinformatics



Continuous Automata

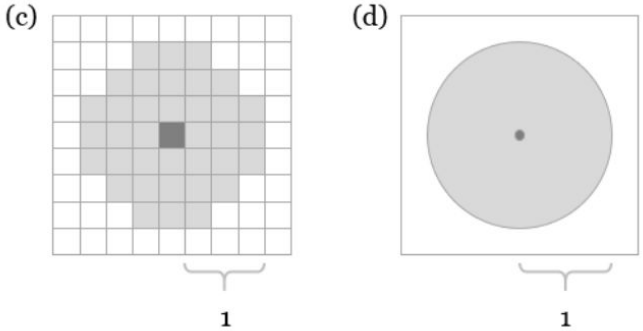
10.48550/arXiv.1812.05433

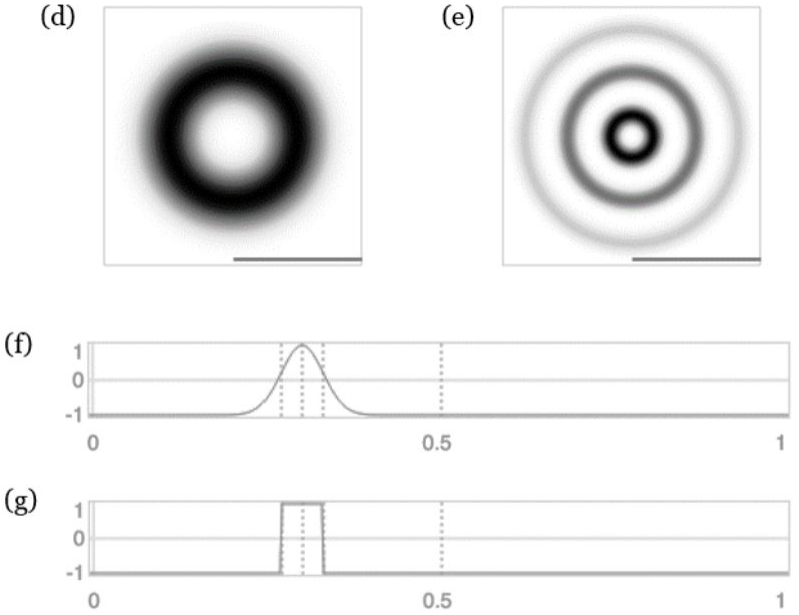


emerging self-reproduction
solves obstacles, mazes
coded as matrix products

Lenia Cell Automata

10.48550/arXiv.1812.05433





$$A^{t+1} = \sigma\left(A^t + \Delta t \cdot G * K\right)$$

Questions ?

computational life

Bee-Free Honey

iGEM Technion 2019

Lake P-Pumps

iGEM Uppsala 2018

Real Vegan Cheese

https://realvegancheese.org

Spider-Free Silk

iGEM Uppsala 2014

Mycelium Bricks

iGEM Design League

Moon Biostimulant

iGEM Toulouse 2024

Open Insuline Project

https://openinsulin.org

Magnetic Gene Control

iGEM Cambridge 2024

Foldit Protein Game

https://fold.it/

DNA Fountain Healing

10.1126/science.aaj2038

Cell Code Compiler

https://cidarlab.org/cello

Eyewire Neuron Game

https://eyewire.org/

Phylogenetic Inference

https://nextstrain.org/

DNA Steganography

10.1038/21092

Universal DNA Registry

https://bricks.bio

ViennaRNA Package

github.com/ViennaRNA

English Playlist

https://youtube.com/playlist?list=PLwiRVNw-iD2wYNyCPUcZ3sR3YtI91hdsw&si=ONzNQjCYxZidJTLO

French Playlist

https://youtube.com/playlist?list=PLwiRVNw-iD2xOaWHnrwUZ7nY_-6D7fE-z&si=t2KsuOeGvXMzR4sv