

Opsporen fecale verontreiniging op zwemwater- locaties



Opsporen fecale verontreiniging op zwemwaterlocaties

Vraag

Hoe kunnen we beter achterhalen wat de bron is van ziekteverwekkers (fecale verontreiniging) in zwemwater, vooral bij menselijke bronnen zoals recreatievaart, rioolwater en overstorten?

Oplossing

We gebruiken metagenomics, een geavanceerde DNA-techniek, in combinatie met AI om precies te bepalen welke bacteriën de vervuiling veroorzaken.

Impact

We brengen de verschillende bronnen in kaart, zodat we gericht en effectiever maatregelen kunnen nemen



Wat is dit?



*Is dit afkomstig van een hond?
Of van watervogels?*



**Afbeelding gegenereerd met AI*

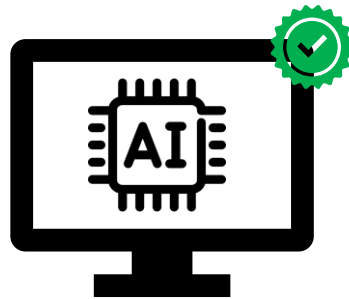
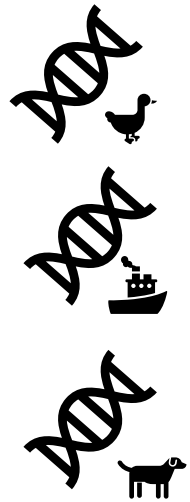


Stap 1: We nemen monsters van bekende verontreinigingen en halen het DNA eruit.

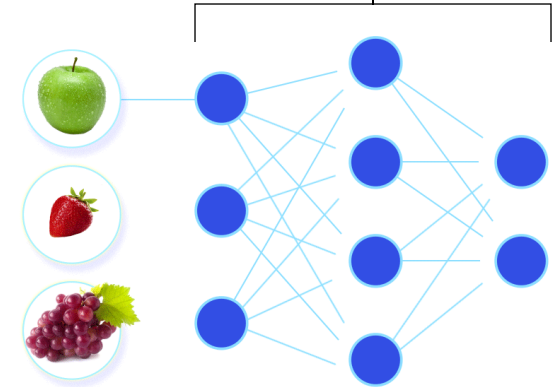




Stap 2: We leren een computer (AI model) om de verschillende verontreinigingen te herkennen.

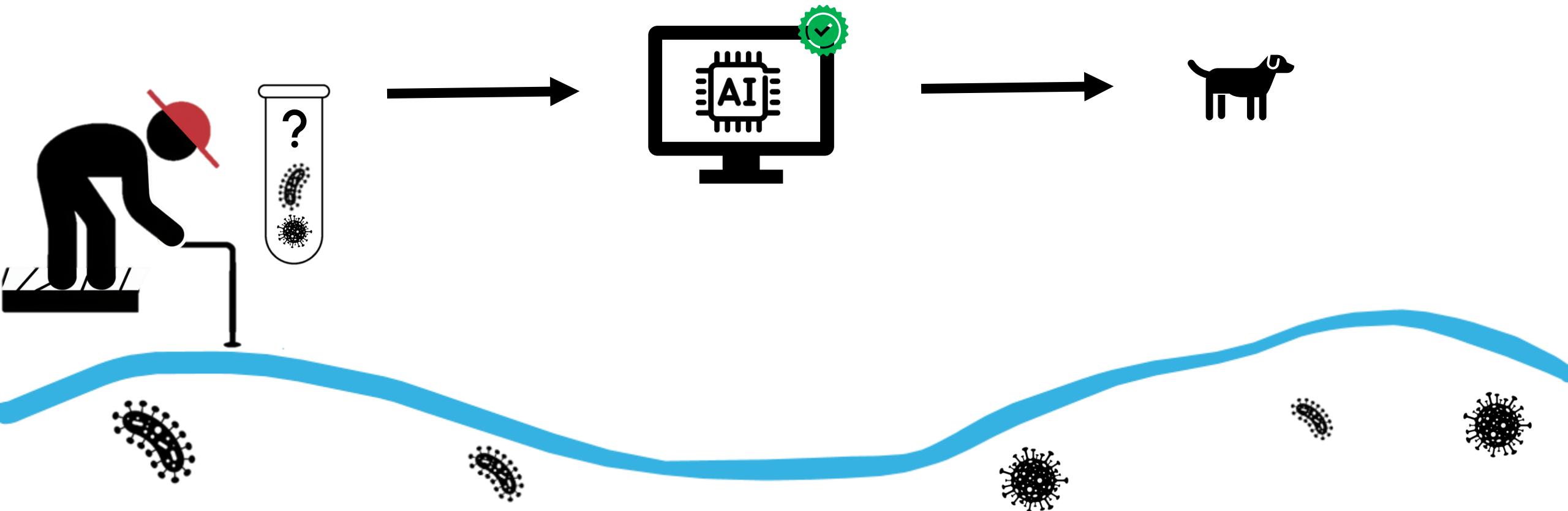


AI model





Stap 3: Het slimme AI model wordt toegepast op een nieuw (onbekend) watermonster en voorspelt wat voor verontreiniging het bevat.



FOR
EXPERTS





1. Je haalt het DNA uit het materiaal dat je hebt verzameld (DNA extractie)
2. Je ‘leest’ het DNA uit, dus je bepaalt de volgorde van de bouwstenen (A, T, C, G) in het DNA dat je hebt geëxtraheerd (DNA sequencing)

Doel: een digitale representatie van DNA materiaal maken wat aanwezig is in het monster.

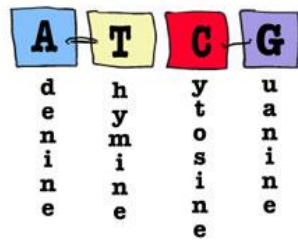
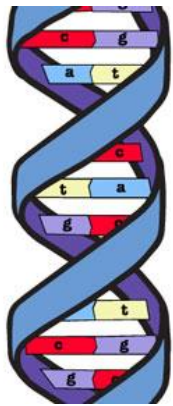
Stap 1: We nemen monsters van bekende verontreinigingen en halen het DNA eruit.





Resultaat:

- Een FASTQ/FASTA bestand met daarin miljoenen sequences/reads.
- Een sequence/read één kort stukje DNA dat tijdens sequencing is gelezen.



```
>188169f6-ca8c-476e-8d88-b29110c69a96 runid=c20bbefdba72dcc3ce403902378885781c0ce858
ATAAAGATCCCATATCATATAAATCGCTTCTCTAAGCGATTTAATAACTCAAATCGTTG
TTCCTATCTTTTGCAAATAAGAATCGTATCCACCATATCTCACACAACCATAGTGTGATA
TTACTAAACTAAAACGGAAGATATCTTTCATATTAGTAAATAATGTGTGCATTGAGTTA

>927e59ba-46d1-49bb-be0f-32691148eee3 runid=c20bbefdba72dcc3ce403902378885781c0ce858
GGAACGGCGCGCAACAACGAGTACTCCGCGCGCAAGTTGAGAAAACGGGAATCAACGGT
TAACAGCGGGAAACAATGGGAATTGCGGGGCTTCCGCTAAGAATCGGCCAAAACCGACC
```

Stap 1: We nemen monsters van bekende verontreinigingen en halen het DNA eruit.

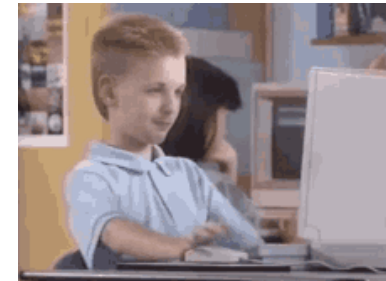




- Een FASTQ bestand is enorm (10+GB per monster)
- Een computer kan niet zo veel met DNA letters (ATCG)



- DNA-sequenties moeten dus worden omgezet naar getallen. **Maarr** biologische context en patronen moeten behouden blijven
- **Doel:** Alle DNA data van één monster samenvatten in rijen van getallen
-> een vingerafdruk



Stap 2: We leren een computer (AI model) om de verschillende verontreinigingen te herkennen.





- DNA sequence: **TCTAGGCGCTTCGGCTGATCCAGCAGCCGGC**..... (soms wel 500k lang)



- Samenvatting van het DNA (embedding) in getallen: **[0.3, -0.1, 0.7, 0.4, -0.2, 0.8, 0.9, -0.1**] (768 nummers)





- DNA sequence: **TCTAGGCGCTTCGGCTGATCCAGCAGCCGGC**..... (soms wel 100k lang)

1) Tokeniseren

- Splits DNA in veelvoorkomende patronen (tokens):
- [CLS] TC | TAG | GCG | CTT | CGG | CT | ... [SEP]
- | | | | | | | | |
|---|----|----|----|----|----|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| 0 | 18 | 45 | 37 | 53 | 39 | 4 | 1 |
- Speciale tokens- [CLS] aan het begin, [SEP] aan het eind- [CLS] wordt gebruikt als “samenvattingsvector” van het hele stuk.
- Waarom: compressie + patroonherkenning + **noodzakelijk voor het model**


2) Omzetten tokens naar *embeddings*





- DNA sequence: **TCTAGGCGCTTCGGCTGATCCAGCAGCCGGC**..... (soms wel 100k lang)

2) Omzetten tokens naar *embeddings*

- We hebben een model nodig dat tokens om kan zetten in betekenisvolle getallen (*embeddings*)
- **DNABERT-S**: AI-model gespecialiseerd in DNA-sequenties 
- Vergelijkbaar met ChatGPT, maar dan voor genetische code ipv menselijke taal
- Voorgetraind op enorme hoeveelheden genomische data
- Begrijpt patronen en context in DNA-sequenties, zoals ChatGPT bijv. grammatica begrijpt.
- State-of-the art, 2024 gepubliceerd.





- DNA sequence: **TCTAGGCGCTTCGGCTGATCCAGCAGCCGGC**..... (soms wel 100k lang)

1) Tokeniseren

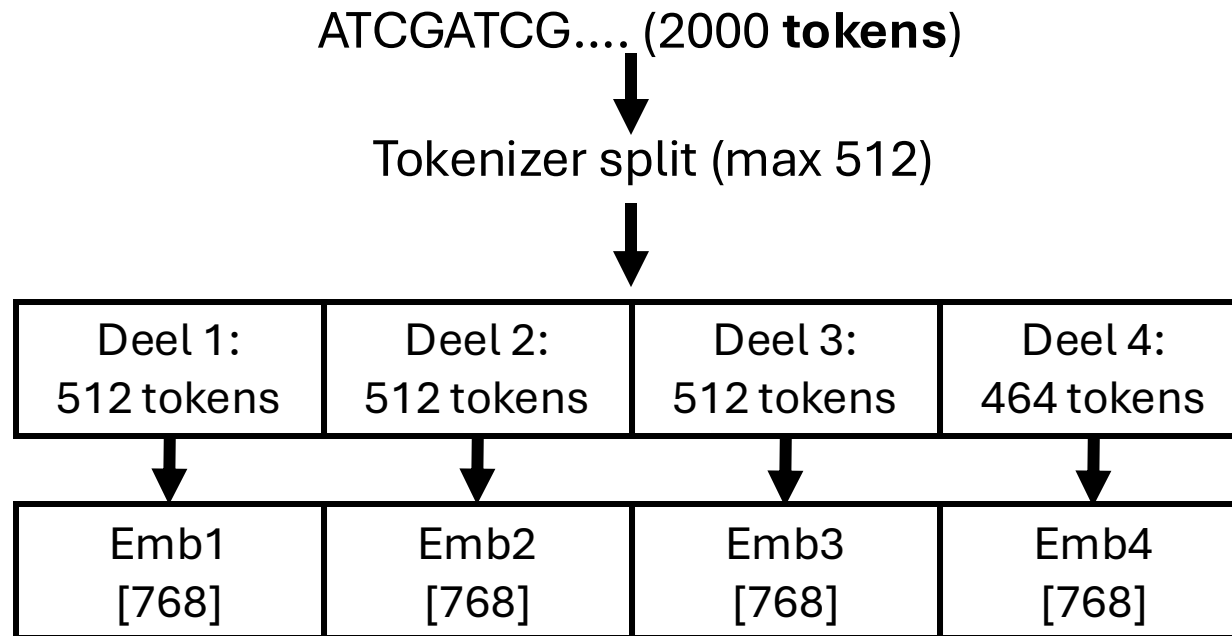
2) Omzetten tokens naar *embeddings*

- Samenvatting van het DNA (embedding) in getallen: **[0.3, -0.1, 0.7, 0.4, -0.2, 0.8, 0.9, -0.1**] (768 nummers)






- Dit doen we niet voor 1 sequence: **TCTAGGCGCTTCGGCTGATCCAGCAGCCGGC**.....
- 1 monster bevat **miljoenen** sequences
- DNABERT-S kan maximaal 512 tokens (context window) per keer verwerken:



4 embeddings voor 1 sequence



- **Resultaat:**

- De 24 monsters hebben we nu ‘samengevat’ in **57 miljoen** rijen met getallen
- Nog steeds best veel :) en omzetting duurt bijna een **week** 

- **Ronde 2** ‘samenvatten’:

- We nemen het gemiddelde per sequence
- We nemen daarna het gemiddelde over 500 sequences
- Resultaat: **100 duizend** rijen met getallen
- Aan dit resultaat voegen we nog de nummers toe bij welke klasse ze horen (4=paard, 5=varken etc.)



Stap 2: We leren een computer (AI model) om de verschillende verontreinigingen te herkennen.

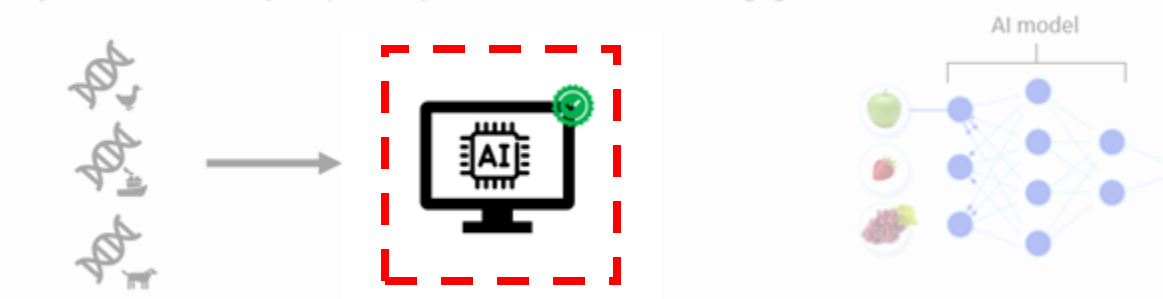




Model trainen:

- **Uitgangspunt:** we trainen een model op *pure klassen* (bijv. alleen hond, alleen vogel, alleen mens).
Probleem: bij toepassing op een *mengmonster* (water met meerdere bronnen tegelijk) past dit niet helemaal.
- Simpel model (baseline)
 - Eenvoudig lineair model
 - Voorspelt één meest waarschijnlijke verontreinigingsbron
 - Uitkomst: “dit monster lijkt het meest op hond” Geen percentages of verhoudingen
- Voor elke verontreinigingsbron een aparte “ja/nee”voorspeller
 - Model kan meerdere bronnen tegelijk detecteren
 - Uitkomst bij mengmonster: “80% kans vogel, 20% kans hond, 80% kans effluent, 0% kans varken”
 - Geeft meer realistisch beeld van vervuilingsmix

Stap 2: We leren een computer (AI model) om de verschillende verontreinigingen te herkennen.





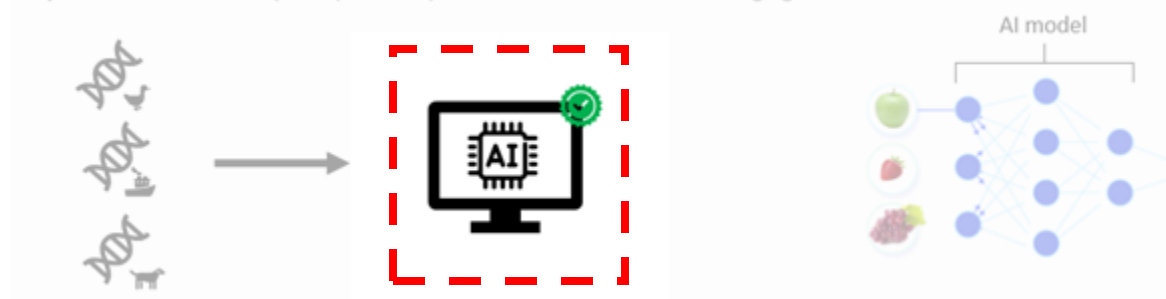
Eerste resultaten simpel model:

Doel: concept testen voordat we opschalen en uitbreiden



Maar 24 monsters met 5 klassen (watervogel, paard, humaan - recreatieve vaart, herkauwer, varken)

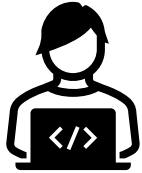
Stap 2: We leren een computer (AI model) om de verschillende verontreinigingen te herkennen.





Validatie van model:

- Getraind op pure samples, toegepast op complexe, gemengde watermonsters met ruis en andere organismen.
- Meerdere fecale bronnen tegelijk aanwezig, model geeft aanwezigheidsscores maar geen zuivere scheiding.
- Afwezige ground truth: geen exacte, onafhankelijke referentiemetingen per bron in het gemengde monster → echte labels onzeker.
- Uiteindelijk model is dus moeilijk kwantitatief te valideren.
- Model output kan zoiets zijn: “80% kans vogel, 20% kans hond, 80% kans effluent, 0% kans varken”
- Het model kan **niet** aangeven: “het monster bevat 70% vogel, 20% varken en 10% overig”.



dev fecale-verontreiniging-classificatie

fecale-verontreiniging-classificatie

+ Find file Code

Merge branch '3-fixes-add-ymconfig' into 'dev' Stefan de Jong authored 17 hours ago

90ec67f1 History

Name	Last commit	Last update
data	Resolve "Improve data loading and preprocessing"	2 days ago
notebooks	Add DNABERT-S embedding processing pipeline for fec...	1 week ago
output	Resolve "Improve data loading and preprocessing"	2 days ago
src	Refactor pipeline and embedder to use configuration fil...	17 hours ago
.gitignore	Resolve "Improve data loading and preprocessing"	2 days ago
README.md	Refactor pipeline and embedder to use configuration fil...	17 hours ago
config.yml	Refactor pipeline and embedder to use configuration fil...	17 hours ago
requirements.txt	Refactor pipeline and embedder to use configuration fil...	17 hours ago

README.md

Fecale Verontreiniging Classificatie - DNA Sequence Embeddings Pipeline

Een schaalbare pipeline voor het omzetten van DNA-sequenties uit zwemwatermonsters naar contextual embeddings met behulp van DNABERT-S. **Alleen inference - geen training.**

Overzicht

Dit project gebruikt het pre-trained DNABERT-S model om microbiële DNA-sequenties uit zwemwatermonsters om te zetten naar 768-dimensionale embeddings. De pipeline is geoptimaliseerd voor grote datasets met chunking en batch processing capabilities.

Kernfunctionaliteit:

- ✓ FASTA/FASTQ bestanden laden en verwerken
- ✓ DNA sequenties tokeniseren met DNABERT-S BPE (Byte Pair Encoding)
- ✓ Automatische chunking van lange sequenties (max 512 tokens per chunk)
- ✓ Batch processing voor memory- efficiënte verwerking
- ✓ GPU/CPU support met automatische detectie
- ✓ Configureerbare chunk sizes en batch sizes
- ✗ Geen model training (alleen inference)

FOR
EXPERTS



Vragen?

