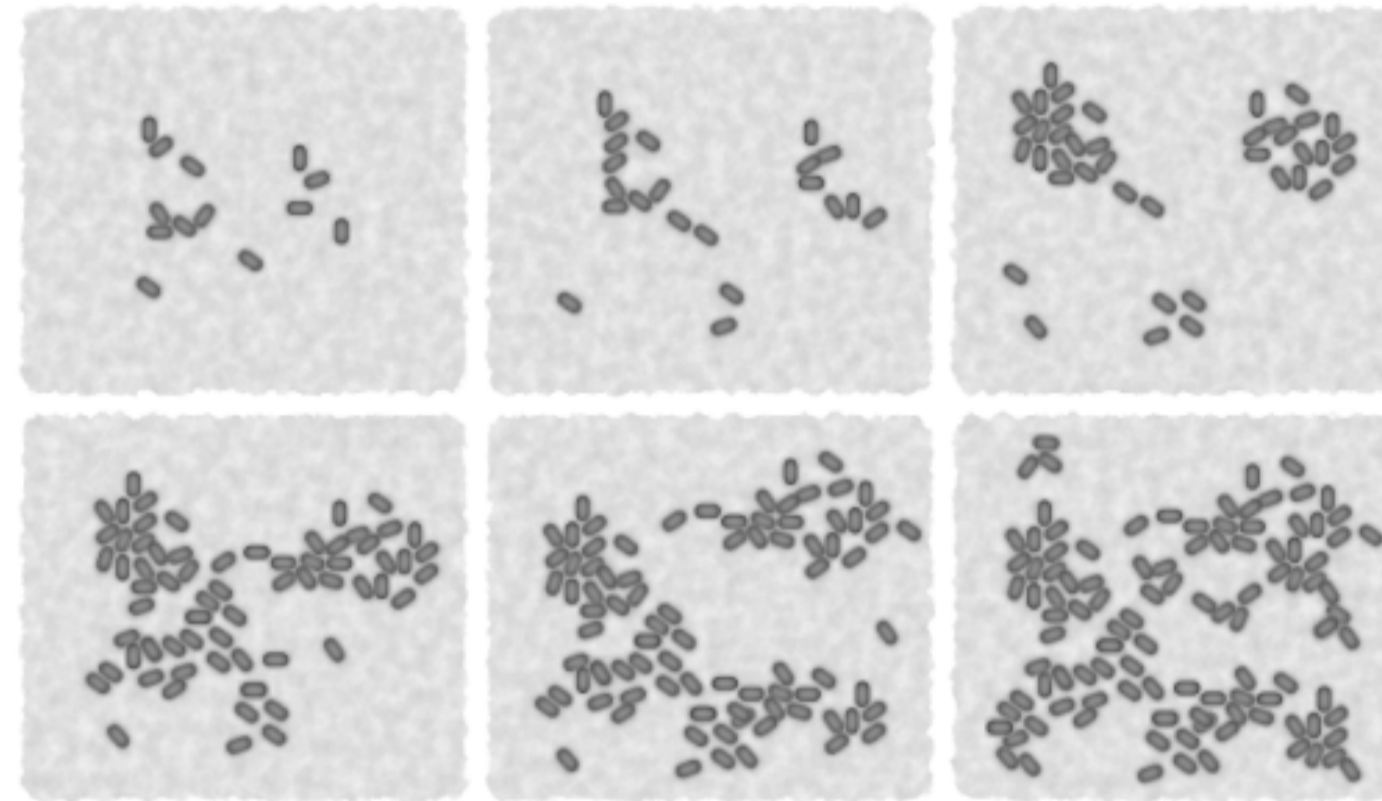


BIOS1100 H17 uke 3

Lex Nederbragt



Ukens forelesning

Ukens forelesning

- noen praktiske ting

Ukens forelesning

- noen praktiske ting
- begrepsforståelse kapittel 2

Ukens forelesning

- noen praktiske ting
- begrepsforståelse kapittel 2
- utvalgte øvelser

Ukens forelesning

- noen praktiske ting
- begrepsforståelse kapittel 2
- utvalgte øvelser
- nytt stoff uke 3

Læringsmål uke 2

- kunne forklare hvordan bakterier formerer seg
- kunne de forskjellige faser i bakteriell vekst

Programmering

- kunne visualisere et enkelt datasett
- kunne lagre visualiseringen i en fil
- kunne last inn et enkelt datasett fra fil
- kunne bruke lister i python

Matematikk

- kunne beskrive en lineær figur i rommet matematisk
- være i stand til å utlede vekstrate og doblingstid for eksponensiell bakterievekst

En del praktiske ting

- obliger: tidspunkter
- celler i notebook
- hva er en python pakke?

Exercise 1: List slicing (1)

Fill in the blank to achieve the desired output

```
bases = ["A", "C", "G", "T"]  
AT_basepair = [bases[_], bases[_]]  
print(AT_basepair)
```

[A, T]

1. A, T
2. 0, 3
3. 1, 4
4. 0, -1

Exercise 2: List slicing (2)

```
list = [0, 2, 4, 6, 8, 10]
```

Which of the following would give the list `[2, 4, 6]`?

1. `list[1:3]`
2. `list[1:4]`
3. `list[2, 4, 6]`
4. `list[0:3]`

Exercise 3: List slicing (3)

What is the output of following program?

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = list1[-2:]
print(list2)
```

1. [5, 6]
2. [1, 2, 3, 4, 5]
3. [5]
4. [6]

Exercise 4: List slicing (4)

What is the output of following code?

```
list1 = [10, 12, 16, 23, 54, 3]
print(len(list1[1:5]))
```

1. 4
2. 5
3. 3
4. 1

Exercise 5: Logarithms

a) What is $\log_2(2^3)$

1. 1

2. 2

3. 3

4. 8

Utvalgte øvelser

- Exercise 3: Death phase: decline rate
- Exercise 7: Oldest people in the world
- Exercise 8: Volume of a sphere and Bergmann's rule

Læringsmål uke 3

Matematikk

- kunne lage og implementer et enkelt eksponensiell vekst modell
- kunne sammenligne resulater fra modellen med eksperimentelle data
- kunne forklare begrensninger med eksponensiell vekst modellen

Programmering

- Numpy arrays
- `for` løkker

Parring

Vi begynner med 1 student som står mens resten sitter. Vi gjør følgende:

1. hver runde finner hver student som står en partner
2. alle nye partnere skal stå opp også

Etter hvor mange runder tror du at alle står?

Parring

Vi begynner med 1 student som står mens resten sitter. Vi gjør følgende:

1. hver runde finner hver student som står en partner
2. alle nye partnere skal stå opp også

Etter hvor mange runder tror du at alle står?

--> Mentimeter

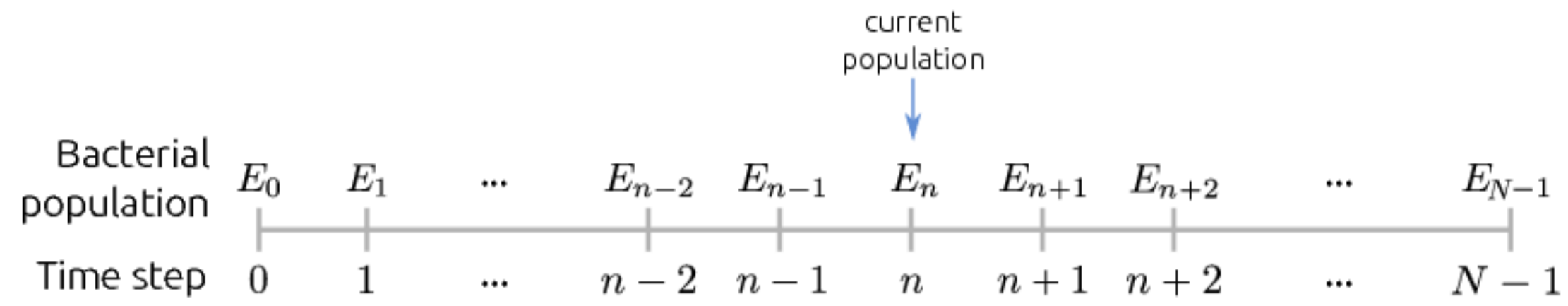
Hvorfor modellering

- simulering
- predikering
- billig og trygt eksperiment
- eksempler
 - klimamodeller
 - sygdomsspredning
 - torskebestander

Hvordan modellering

- matematisk modell
 - sett med regler
- implementering
 - beregningsmodell
 - algoritme

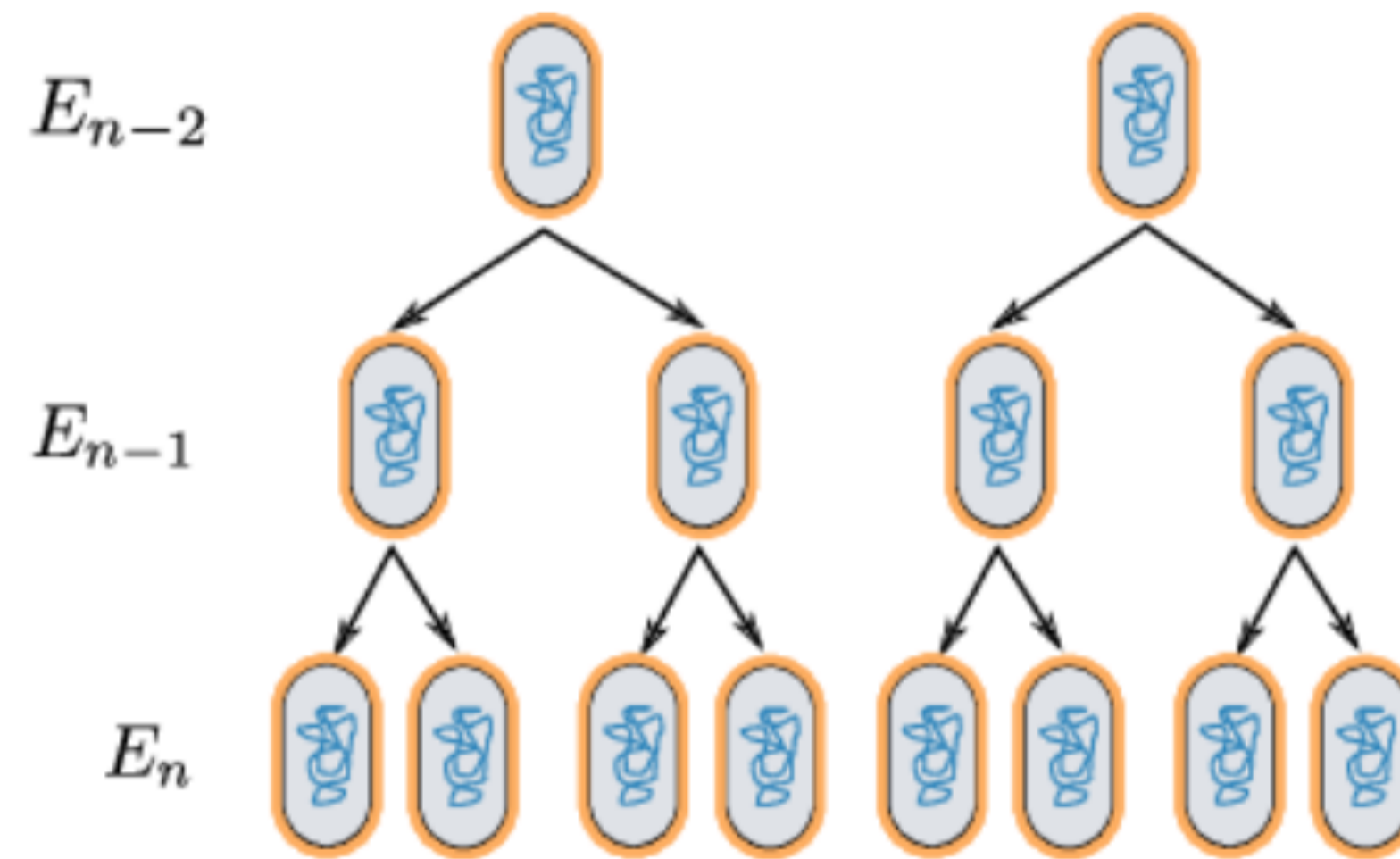
Matematisk modell



Matematisk modell

After each time step, we have twice as many bacteria.

$$E_n = 2E_{n-1}. \quad (1)$$



Implementering

Python

- Numpy arrays
 - som lister, men mer egnet for matematikk
- `for` løkker
 - jobbe effektivt med lister, arrays mm

Repetisjon

Rule of three.

In programming, there is a rule of thumb called "the rule of three". It states that if a task has to be repeated three times or more, we should automate it.

Repetisjon

Rule of three.

In programming, there is a rule of thumb called "the rule of three". It states that if a task has to be repeated three times or more, we should automate it.

- manuell repetisjon er tidskrevende

Repetisjon

Rule of three.

In programming, there is a rule of thumb called "the rule of three". It states that if a task has to be repeated three times or more, we should automate it.

- manuell repetisjon er tidskrevende
- manuell repetisjon kan lett bli en feilkilde

For loops

```
numbers_list = [7, 2, 4, 3]
```

```
print(numbers[0])
```

```
print(numbers[1])
```

```
print(numbers[2])
```

```
print(numbers[3])
```

Eller

For loops

```
numbers_list = [7, 2, 4, 3]

print(numbers[0])
print(numbers[1])
print(numbers[2])
print(numbers[3])
```

Eller

```
numbers_list = [7, 2, 4, 3]

for number in numbers_list:
    print(number)
```

Volume of a sphere and Bergmann's rule

Calculate the ratio $\frac{SurfaceArea}{Volume}$, for the following radiuses:

0.5, 1, 1.5, 2, 2.5, 3, 3.5 and 4 meter.

Store the results in two different lists called `radiuses` and `ratios`.

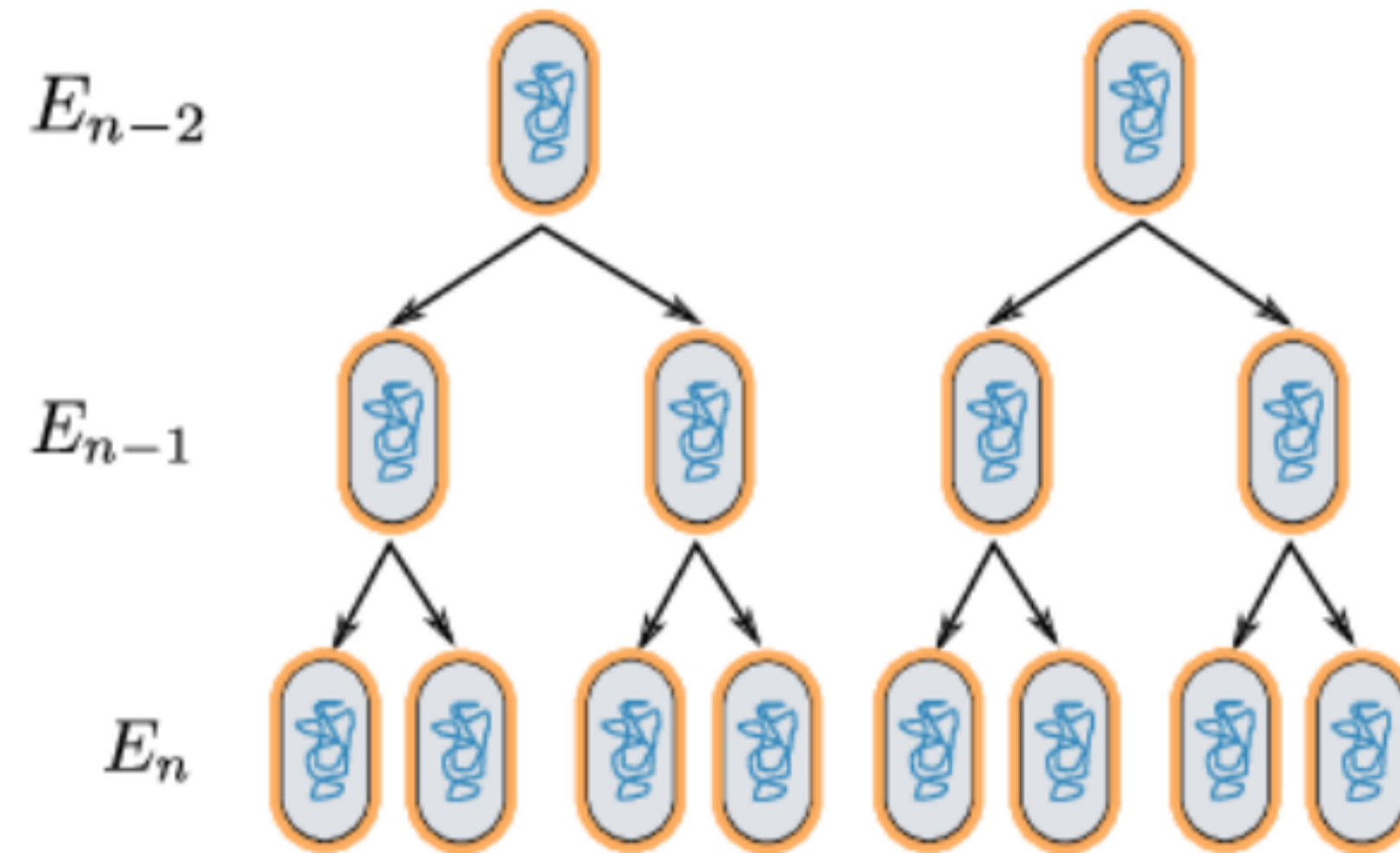
Start the `ratios` list as an empty list:

```
ratios = []
```

Then add the ratios each time you calculate them using the `ratios.append()` function.

Growth model for *E. coli*

$$E_n = 2E_{n-1}. \quad (2)$$



Growth model for *E. coli*

$$E_n = 2E_{n-1}, \tag{3}$$

$$t_n = t_{n-1} + \Delta t, \tag{4}$$

Growth model for *E. coli*

$$E_n = 2E_{n-1}, \quad (3)$$

$$t_n = t_{n-1} + \Delta t, \quad (4)$$

```
for n in range(1, N):  
    E[n] = 2*E[n-1]  
    t[n] = t[n-1] + delta_t
```

Python versus matte

$$x = x + 1.$$

(5)

Python versus matte

$$x = x + 1. \tag{5}$$

```
x = 12  
x = x + 1  
print(x)
```

Python versus matte

$$x = x + 1.$$

(5)

```
x = 12  
x = x + 1  
print(x)
```

13