Introduction
All correspondence via piazza
All emails will be silently ignored
DO NOT WRITE EMAILS TO DR. KRUSCHE, PROF. BRUEGGE, OR THE PRESIDENT OF THE UNIVERSITY
Standard exercise group will be deleted soon
Register for one of the four available groups
Material is uploaded to rostlab.org
Watch rostlab for changes
Communication mainly on Piazza

piazza.com/tum.de/summer2019/pp1cs

Get help by instructors and students

Not for sharing code! Mainly technical issues or task clarification
Exercise Structure

- Rehearsing and explaining lecture material
- Discussion of previous homework
- Q&A session
- Introduction to new homework
Coding Exercise

- Python 3
- Strict compliance with the provided template
- Homework worksheets are published on www.rostlab.org each Thursday
- Submit homework before Tuesday 10am, 2 weeks after publishing (you have 1.5 weeks)
- We will check for plagiarism in your code
- Do not include personal data in your code as it might be uploaded to non-TUM servers (plagiarism check)
We publish test cases. Use them to make sure your code runs and adheres to the specification.

The global tests use different data with the same tests and might be slightly more extensive.

Don’t try modifying the local test code! It will not be used anyway.

We won’t answer questions regarding the content of the server.
Testing using pytest

$ pip install pytest

Run pytest in code directory

$ pytest
There will be short quizzes at the end of the lecture that give additional points

The quizzes will be held and completed in the lecture time slot

You can achieve a grade weighted with 50% on a passed final exam

Exercise grading will be announced later in the semester

Points will consist of 80% programming and 20% quizzes

You need to achieve at least 75% of the total points to be eligible for the bonus
Exercise
If you are new to python, complete the tutorial at https://docs.python.org/3/tutorial/
In order to participate in the exercises you need to know basic git commands

If you are new to Git, complete the tutorial at https://try.github.io
Artemis

- AuTomated assEssment Management System for Interactive Learning
- https://artemis.ase.in.tum.de
- Log in using your TUM online credentials
- You need to register yourselves
- Do not rely on ArTEMiS submission time as that relies on your system time and thus might be wrong
Your current courses
Your current courses

Protein Prediction 1 (Summer 2019)

Sign up for selected course  Cancel
Your current courses

Protein Prediction 1 (Summer 2019)

Your current score:
0%

No exercise planned
Exercise 0

- Clone Repository
- Easy

0%, Build Failed

14/05/19 (in 13 days)
Mainly aimed at testing that you have a working setup
Start the exercise in ArTEMiS & clone the repository
Modify the method complementary in the main.py file, s.t. it returns a string of complementary DNA nucleobases for a given string
Test your method by running pytest in the directory of the main file.
For example, it should return T for A, C for G, and ATGC for TACG
Commit your changes and push to run server tests
Start and Clone

Exercise 0

Clone your personal repository for this exercise:

https://ab59ycgpe3ebrvepui.x.tum.de/xm/PP15518EXERCISE8/pe1518exercise8-exercise-ab59xyz.git

Copy URL Clone in SourceTree Albatross SourceTree is the free Git client for Windows or Mac.
## Exercise Test

<table>
<thead>
<tr>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>🟢</td>
<td>100%, 7 passed</td>
</tr>
<tr>
<td></td>
<td>07/00/19 (in 4 months)</td>
</tr>
<tr>
<td>🔴</td>
<td>Not released</td>
</tr>
</tbody>
</table>

---

**Exercise Test**

**Status:** 100%, 7 passed

**Details:** 07/00/19 (in 4 months)

**Status:** Not released

collected 5 items

tests/test_main.py  FFFFF

================================= FAILURES ==========

def test_a():
    func_res = str(main.complementary('A')).upper()
    test_res = func_res == 'T'
    > assert test_res
    E    assert False

tests/test_main.py:7: AssertionError

def test_t():
    func_res = str(main.complementary('T')).upper()
    test_res = func_res == 'A'
    > assert test_res
    E    assert False

tests/test_main.py:13: AssertionError

def test_g():
    func_res = str(main.complementary('G')).upper()
    test_res = func_res == 'C'
    > assert test_res
    E    assert False

tests/test_main.py:19: AssertionError

def test_c():
    func_res = str(main.complementary('C')).upper()
    test_res = func_res == 'G'
    > assert test_res
    E    assert False

tests/test_main.py:25: AssertionError

def test_complete():
    func_res = str(main.complementary('ATGC')).upper()
    test_res = func_res == 'TACG'
    > assert test_res
    E    assert False

tests/test_main.py:31: AssertionError

================================= 5 failed in 0.05 seconds
Check for syntax error if you receive feedback like this:

Feedback
Error in method tests.test_main:

collection failure
Thank you!

QUESTIONS?