Coursework

Your own computational biology project

T. Cragnolini - 29/01/2025

Coursework structure

- You are asked to write Python code that reads in some input, does some processing, then output some results.
- This should take the form of a complete project:
  - separate data files
  - An executable python script
  - Generation of an output
  - Commented and tested
  - With version control used

Coursework

- I will provide “real-life” examples, or as close to it as possible
- (Optional) Write me a paragraph about a topic you find interesting, related to bioinformatics / computing.
  - Or if you have a project in mind, write about that too!

Projects

Projects

- Classify cryo-EM particles (images)
- Identify a dog breed

Cryo-EM particle picker

- Context: you are building a pipeline for cryo-EM reconstruction
- Identify whether the picked(selected) particle are similar enough to manually selected ones.
- Input: images (adapted from EMPIAR)
- Output: a decision per image
- Stretch goal 1: image clusterings
- Stretch goal 2: clustering visualisation

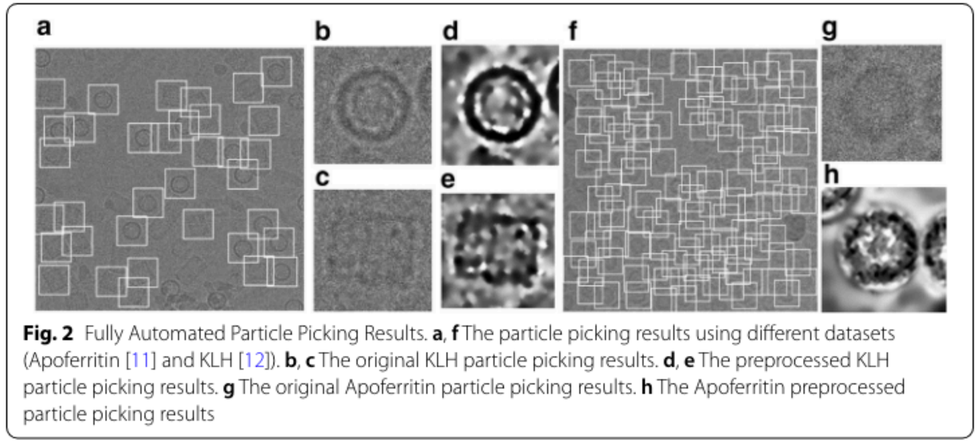


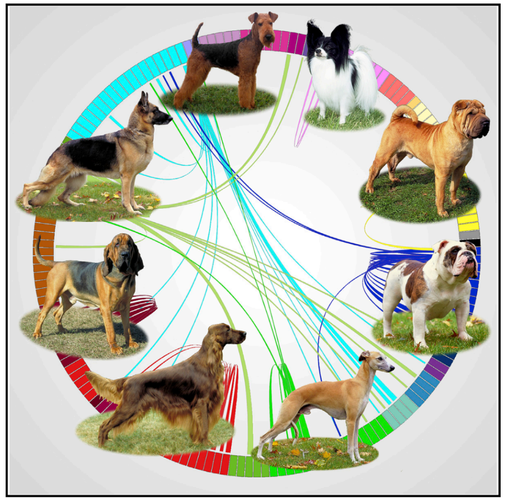
Fig. 2 Fully Automated Particle Picking Results. **a**, **f** The particle picking results using different datasets (Apoferritin [11] and KLH [12]). **b**, **c** The original KLH particle picking results. **d**, **e** The preprocessed KLH particle picking results. **g** The original Apoferritin particle picking results. **h** The Apoferritin preprocessed particle picking results

https://doi.org/10.1186/s12859-020-03885-9

- N.B.: I will provide the decision routine

Identify the most similar sequence

- Context: you want to develop a DNA identification service
- Identify the closest sequence in the database to the provided sequence
- Input: sequence database (adapted from GEO), test sequence
- Output: the closest sequence, and the difference
- Stretch goal 1: Probabilities across database, p-value
- Stretch goal 2: reconstructed phylogeny



- N.B.: I will provide the database and test sequence

http://dx.doi.org/10.1016/j.celrep.2017.03.079

Projects

- Both are real-life problems (although limited in scope here)
- Both make use of modern techniques (machine learning, NGS data)
- Both are about as hard as the other
- Choose one, stick with it!
- We will see how to structure projects
- It doesn’t have to be perfect, just good enough

Evaluation

Elements of evaluation

- Project organisation and quality:
  - Is the project organised in folders / subfolders
  - Is there documentation?
  - Are you using a VCS to manage changes?
- Correctness
  - Does the code run ?
  - Is the output correct?

Elements of evaluation

- Code quality
  - Is the code clear and consistent?
  - Are the variable/class/module names meaningful?
  - Is the code organised in function, class, modules?
    - N.B.: Do not use class if you can’t see the point of it!
  - Are there tests?
    - Unit, integration tests?

Example marks

- Pass: the code just about runs and give a correct/reasonable output (maybe with *very* minor changes)

- Merit: the above, and project is well documented, tested, and organised, with one stretch goal

- Distinction: all of the above, extensive docs and tests, both stretch goals