Project in molecular Life science

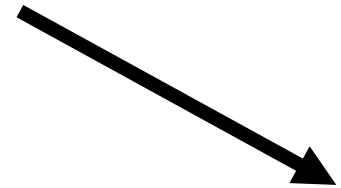
(KB8024/KB8025)

Steps of a Machine Learning Project

- Define a problem to solve (each of you will have one)
- Prepare the data (we prepared the datase for you)
- Evaluate different algorithms according to your problem (start with svm)
- Improve Results (try different parameter)
- Present Results (write the report)

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Projects (http://bioinfo.se/courses/project-in-molecular-life-science-kb7006/):



Updated later:
Datasets:
Dataset part 1
Dataset part 2
Grading
The grading will be based on the quality of your oral presentation, assignments, participation and the final report. To pass the course all assignments that are not marked in bold(see above) need to be performed. To obtain a grade C or higher you also need to perform the tasks in bold.
Tasks (see also schedule)
All days: Diary on github !!!

- Week 1 (Day 1-2)
  - Bash etc http://swcarpentry.github.io/shell-novice/
  - Git http://swcarpentry.github.io/git-novice
  - Python http://swcarpentry.github.io/python-novice-inflammation/
  - Linux Tips and Tricks
  - How to organize your project
  - Compulsory exercises
    - Write a bash script that when run creates a template project folder structure(see above on tips on how to organize your project).
    - Create a new repo on github (sign up if you do not already have an account) and push your file to this repo. Send a link to the repo to john.lamb@scilifelab.se.
  - Friday: Work on your own
  - Friday: Elofsson group meeting @scilifelab

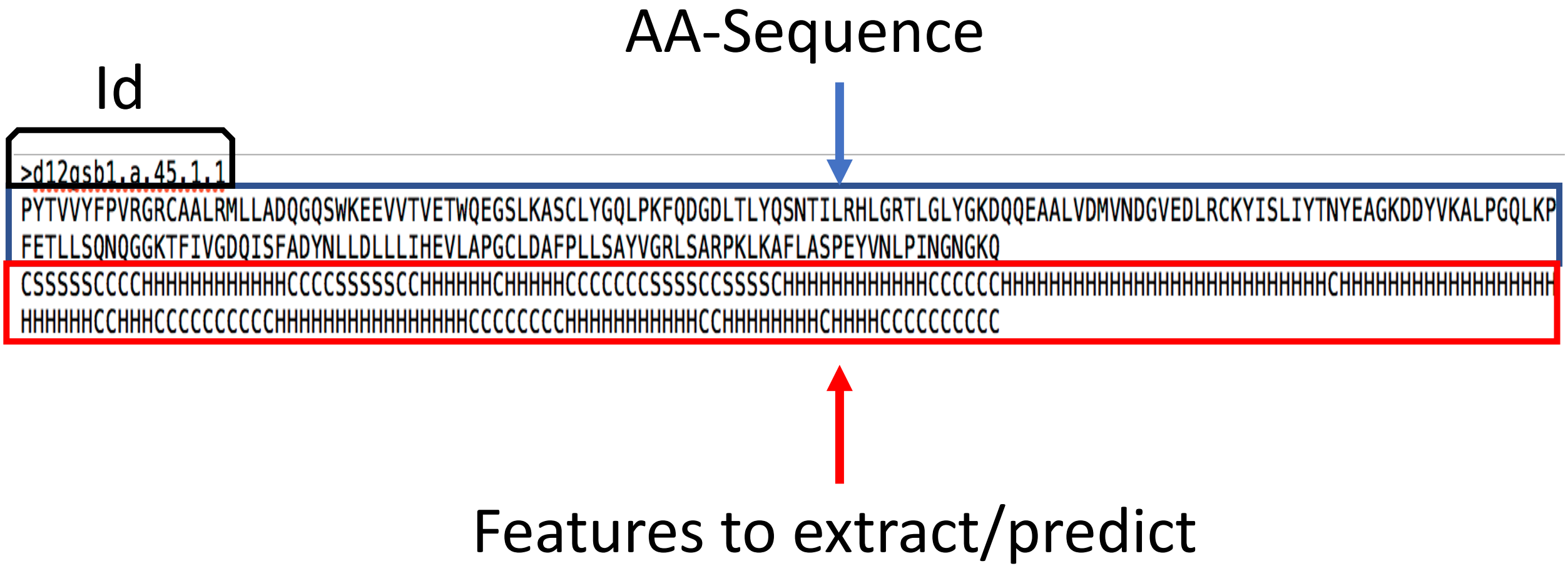
What you have to do **(Mandatory)**:

- Extract the feature from your dataset
- Create cross-validated sets
- Train a SVM using single sequence information, using sklearn
- Check different window sizes for the inputs
- Analyze the results and compare it to previous work
- Review the state of art for your predictor
- Write a report

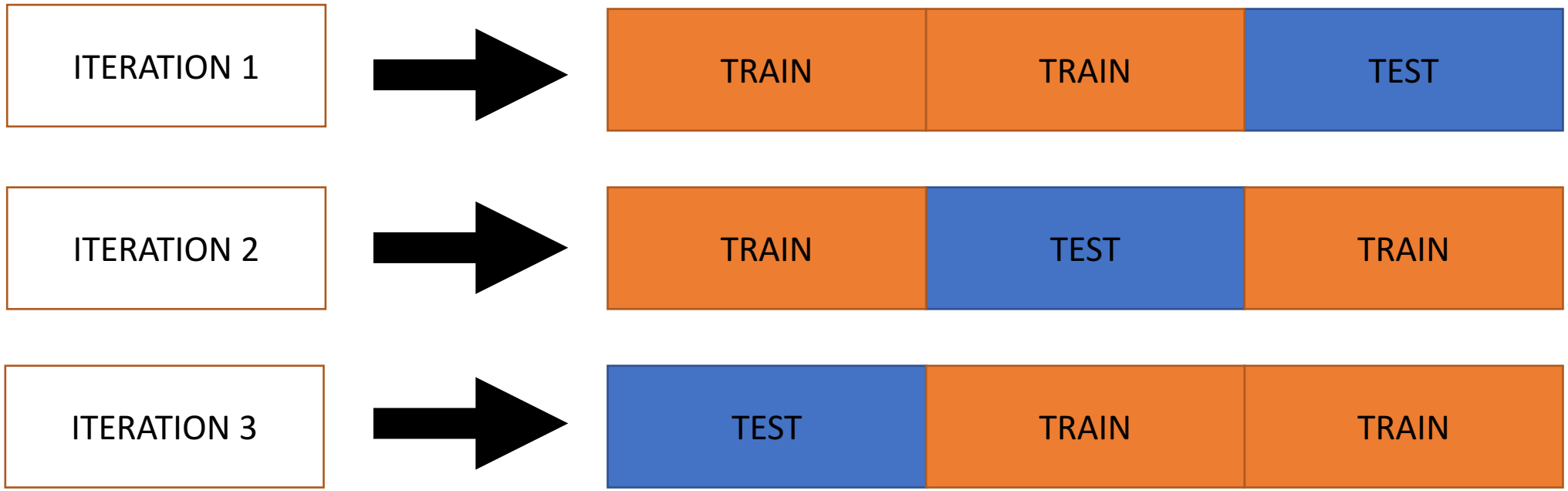
What you could do **(Not Mandatory)**:

- Add evolutionary information by running psi-blast. Use psi-blast information and train a SVM.
- Optimize the performance of the SVM. *(play with parameters)*.
- Train a random forests and a simple decision tree *(always using sklearn)*.
- Compare the performance with the SVM performance.
- Extract the data from 50 other proteins and test the performance.

Extract the feature from your dataset

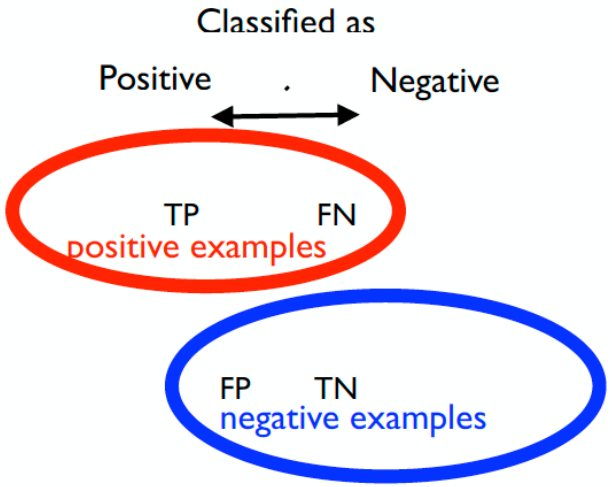


Create cross-validated sets



Analyze the results (Metrics) From Arne Lecture

	Classified as positive	Classified as negative
Positive example	TP	FN
Negative example	FP	TN



- TP = True positive = Correctly classified as positive example
- FP = False positive = Incorrectly classified as positive example
- FN = False negative = Incorrectly classified as negative example
- TN = True negative = Correctly classified as negative example

Analyze the results (Metrics) From Arne Lecture

precision=accuracy=TP/(TP+FP)

recall=sensitivity=TP/(TP+FN)

True Positive Rate=TPR=TP/(TP+FN)

False Positive Rate=FPR=FP/(FP+TN)

False Discovery Rate=FDR=FP/(FP+TP)

accuracy=1-FDR

Matthews correlation coefficient=
MCC=(TP*TN-FP*TN)/sqrt(TP+FP)(TP+FN)(TN+FP)(TN+FN)

Resources & Literature:

- https://machinelearningmastery.com/
- https://machinelearningmastery.com/machine-learning-in-python-step-by-step/
- Jones DT. (1999) Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292: 195-202.
- Noble WS (2009) A Quick Guide to Organizing Computational Biology Projects. PLoS Comput Biol 5(7): e1000424.
- Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf