Software-Project Scientific Computing

Frank Noé
frank.noe@fu-berlin.de

Learning Aims

- Learned to conduct a scientific software development project
- Be able to acquired information of the numerical methods to be implemented
- Understand a few important algorithms for analyzing high-dimensional scientific data
- Collaborate in a small group (<10)
- Be familiar with:
  source code repository management,
  public development platform,
  source code testing,
  continuous integration
- Have a rough idea of software architecture design
- Improve programming skills, in particular python, C(++), CUDA.

Rules

- If you are in, stay in.
- We are one team, everyone helps the others
- The project is finished when the features are functional, or at the end of the semester. We all finish together.
- Nobody does something and then leans back for the second half of the semester.
- If you are done with something, regroup
- Everyone has to present and contribute
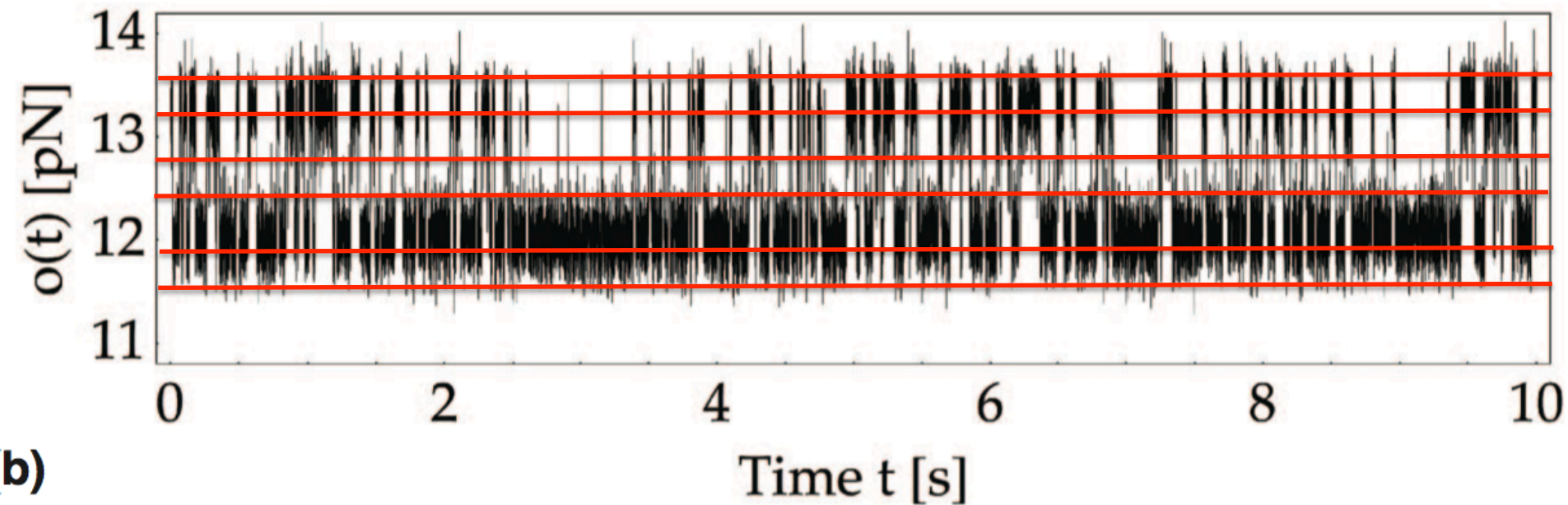
Data processing algorithms
3. Hidden Markov model
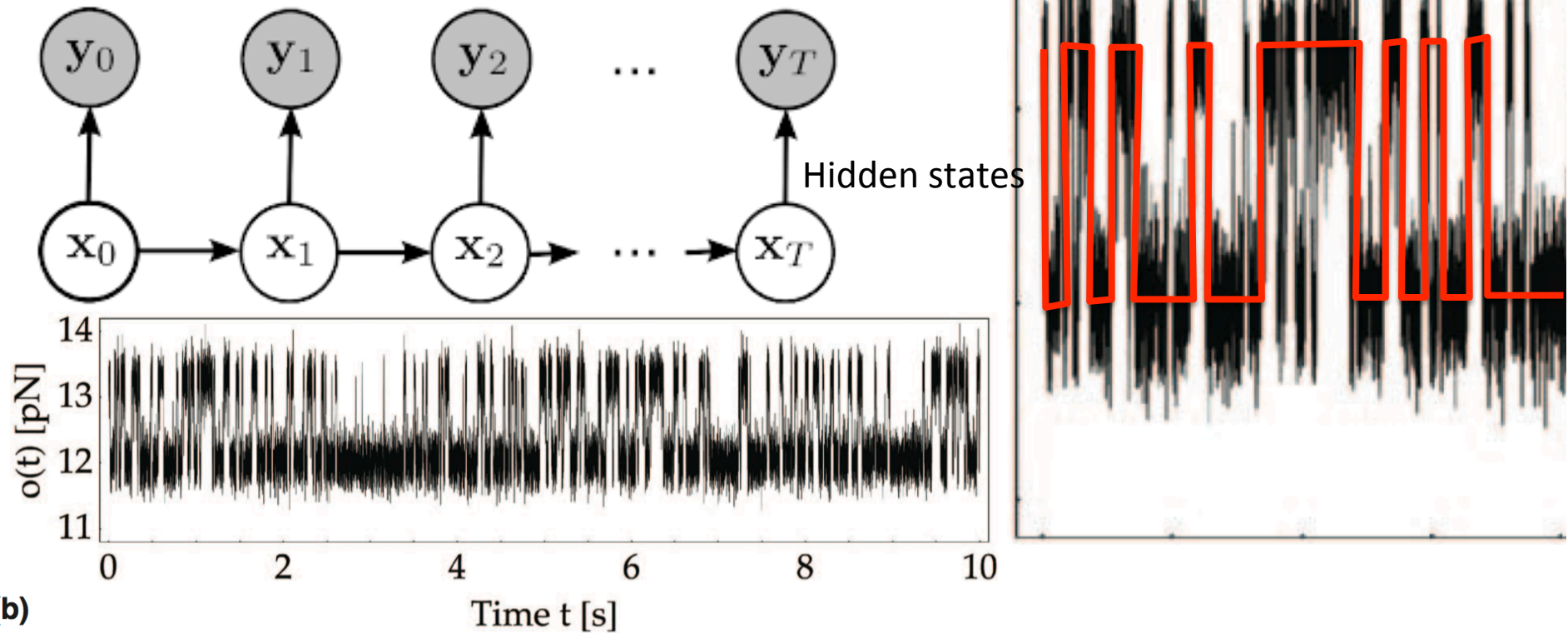
Observed states

Hidden states

- discretize data without suffering from curse of dimensions
- works well in a few dimensions (typically <10)

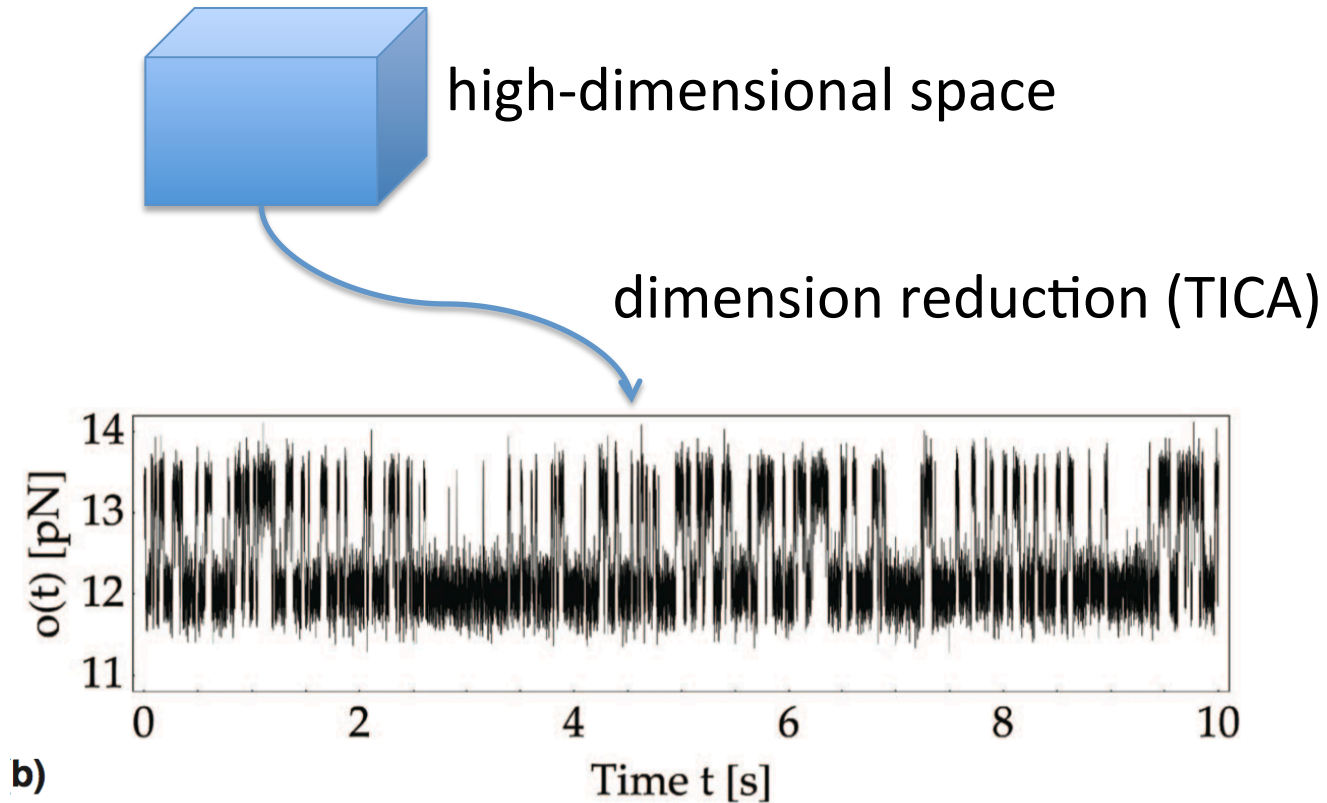# Data processing algorithms
2. k-means clustering

- discretize data without suffering from curse of dimensions
- works well in a few dimensions (typically <10)

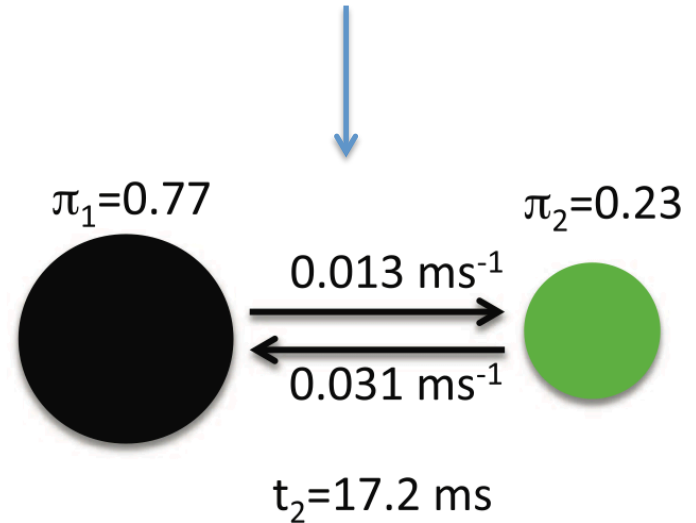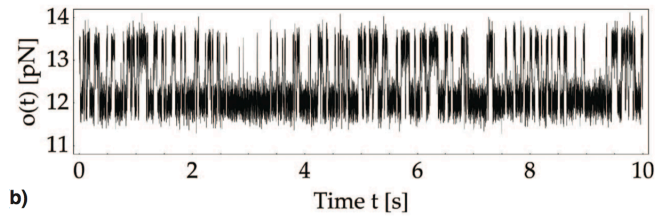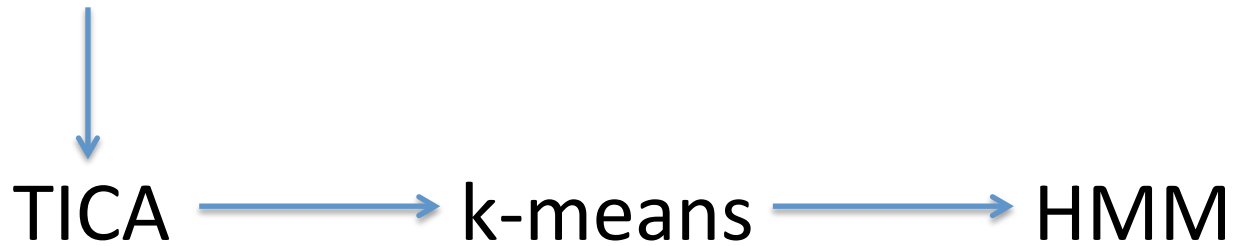# Data processing algorithms
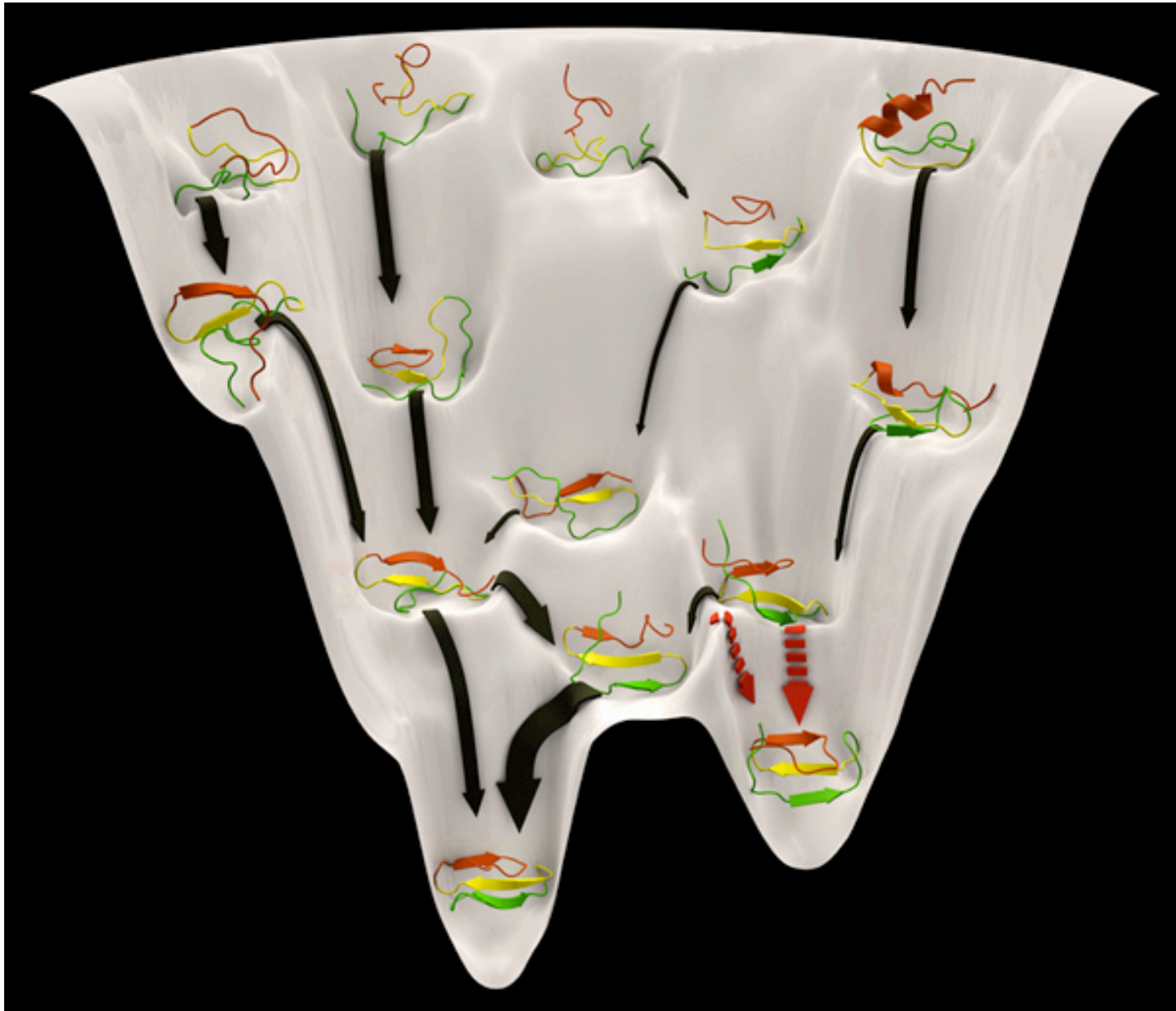# 1. time-lagged independent component analysis

high-dimensional space

dimension reduction (TICA)

b)

Pipeline:

high-dimensional data stream

TICA → k-means → HMM

b)
$o(t)$ [pN] vs Time t [s]

$\pi_1=0.77$ 0.013 ms^{-1} $\pi_2=0.23$

0.031 ms^{-1}

$t_2=17.2$ ms

Application: molecular dynamics

Our software library: EMMA

Overview

HTTPS ▾   https://fnoe@bitbucket.org/cmb_fu.   Share

Last updated	2014-10-14	**4** Branches	**1** Tag
Language	Python		
Access level	Admin (revoke)	**0** Forks	**7** Watchers

Invite users to this repo

Send invitation

Recent activity

return_conv argument in mle_trev (insi...
Issue #89 commented on in cmb_fu/EMMA
Guillermo Pérez-Hernández · 22 hours ago

return_conv argument in mle_trev (insi...
Issue #89 commented on in cmb_fu/EMMA
Guillermo Pérez-Hernández · 23 hours ago

return_conv argument in mle_trev (insi...
Issue #89 created in cmb_fu/EMMA
Guillermo Pérez-Hernández · 23 hours ago

TICA: support xtc trajectories
Issue #28 commented on in cmb_fu/EMMA
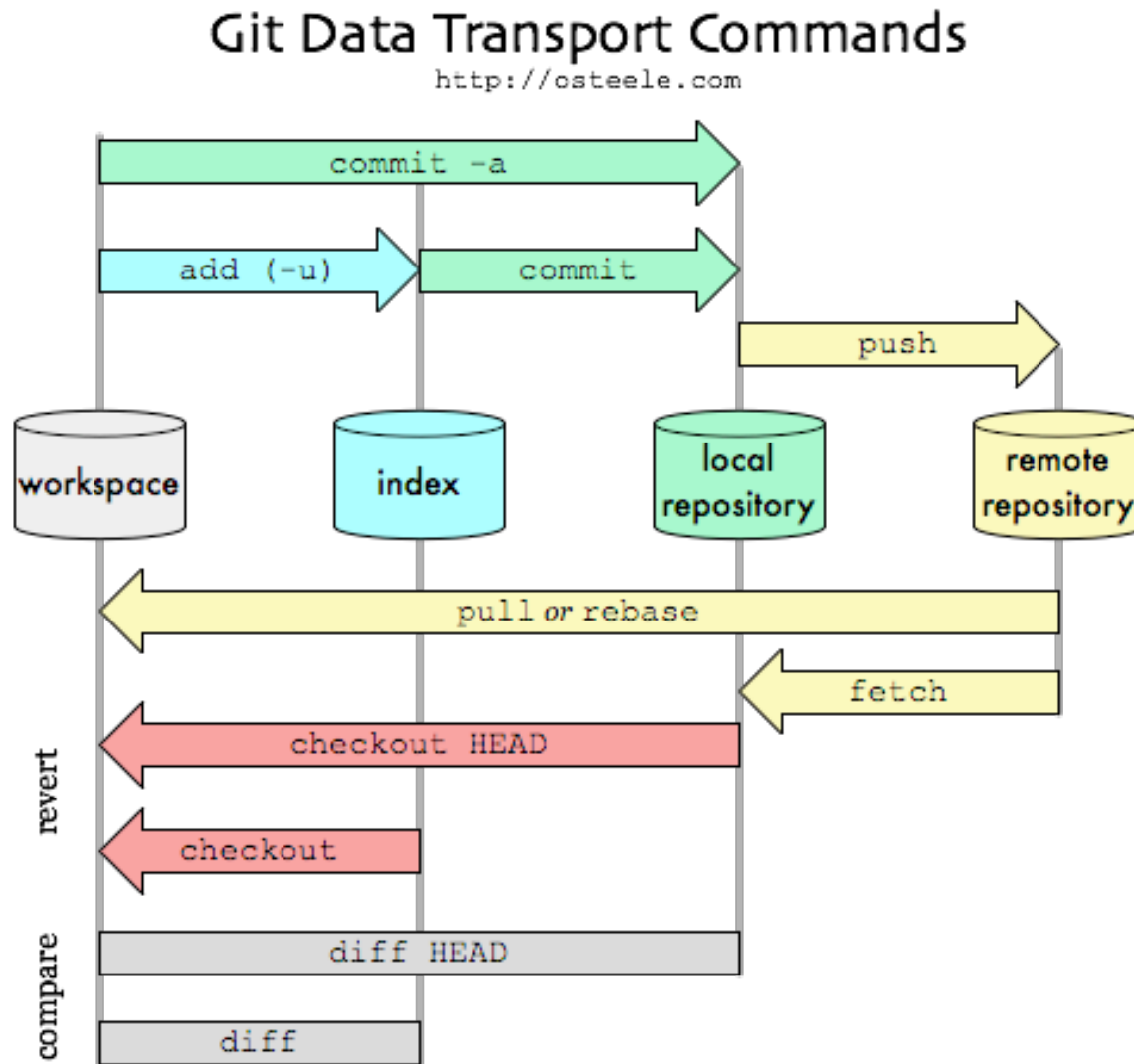Frank Noe · yesterday

2 commits
Pushed to cmb_fu/EMMA
5cfd7d7   Merge branch 'devel' of bitbucket...
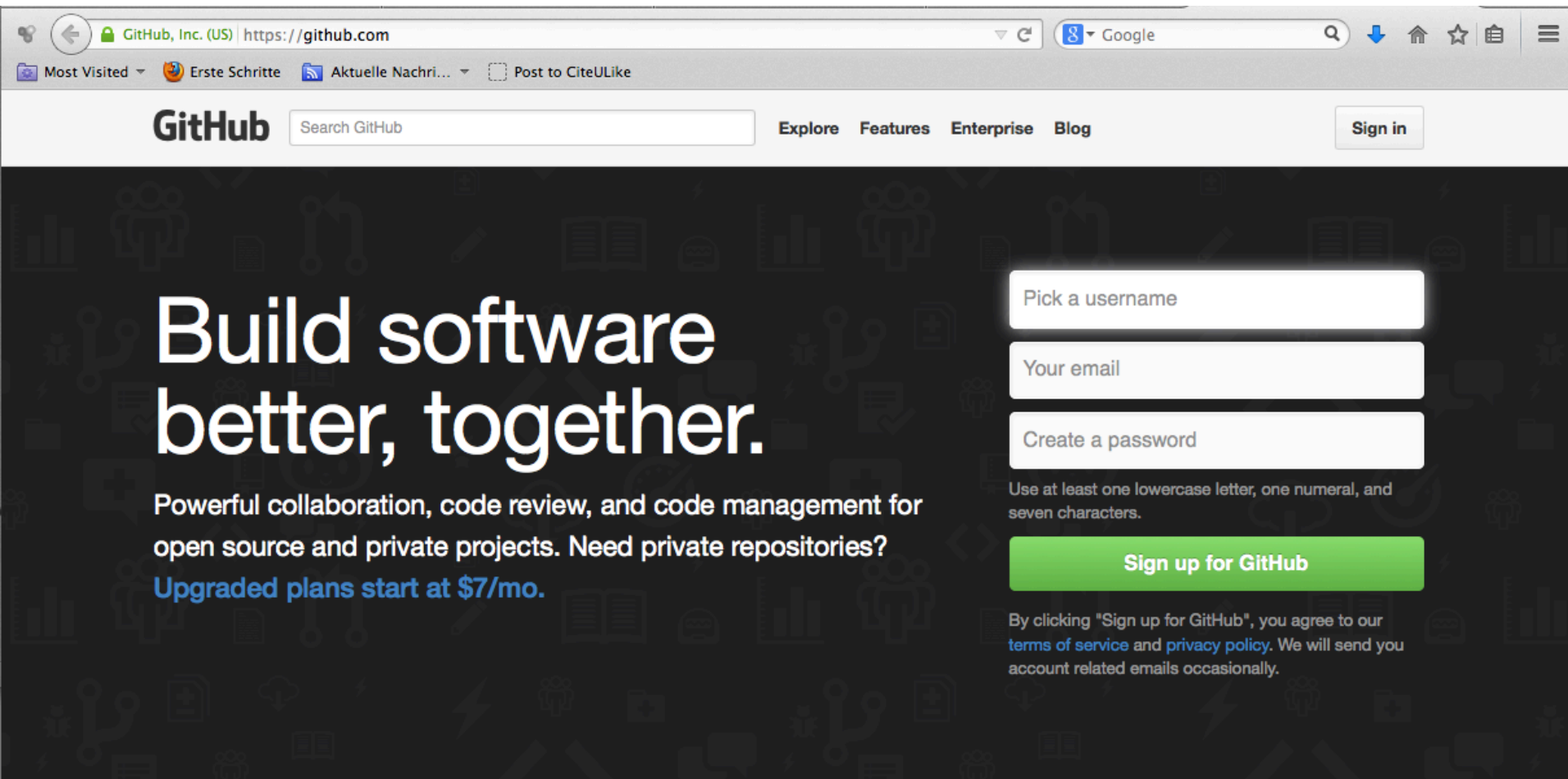5ea06b4   [setup] require new version sche...
Martin Scherer · 2 days ago

✎ Edit README

EMMA (Emma's Markov Model Algorithms)

What is it?

This software is a collection of algorithms to analyze trajectories generated from any kind of simulation (e.g. molecular trajectories) via Markov state models (MSM).

It provides an API for estimation and analyzing MSM and various utilities to process input data (clustering, coordinate transformations etc). For documentation of the API, please have a look at the sphinx docs in doc directory.

For some examples on how to apply the software, please have a look in the ipython directory, which shows the most common use cases as documented IPython notebooks.

Installation

To install it refer to the installation guide in doc/source/INSTALL.rst

Support

For support/bug reports/sugguestions/complains please file an issue on GitHub. http://github.com/cmb-fu/emma

source code repository: **git**

Git Data Transport Commands

http://osteele.com

commit -a

add (-u)   commit

push

workspace   index   local repository   remote repository

pull *or* rebase

fetch

checkout HEAD

checkout

diff HEAD

diff

revert

compare

You'll need a repository whenever you are not working alone.
But even when you work alone it has many advantages.

public development site: **www.github.com**

- Github offers you free git repositories, combined with issue management and other features in a nice website.
- „Github is like facebook for coding"

software design / continuous integration

| run_tica | run_kmeans | run_hmm | shell scripts |

| tica | kmeans | hmm | Python packages |
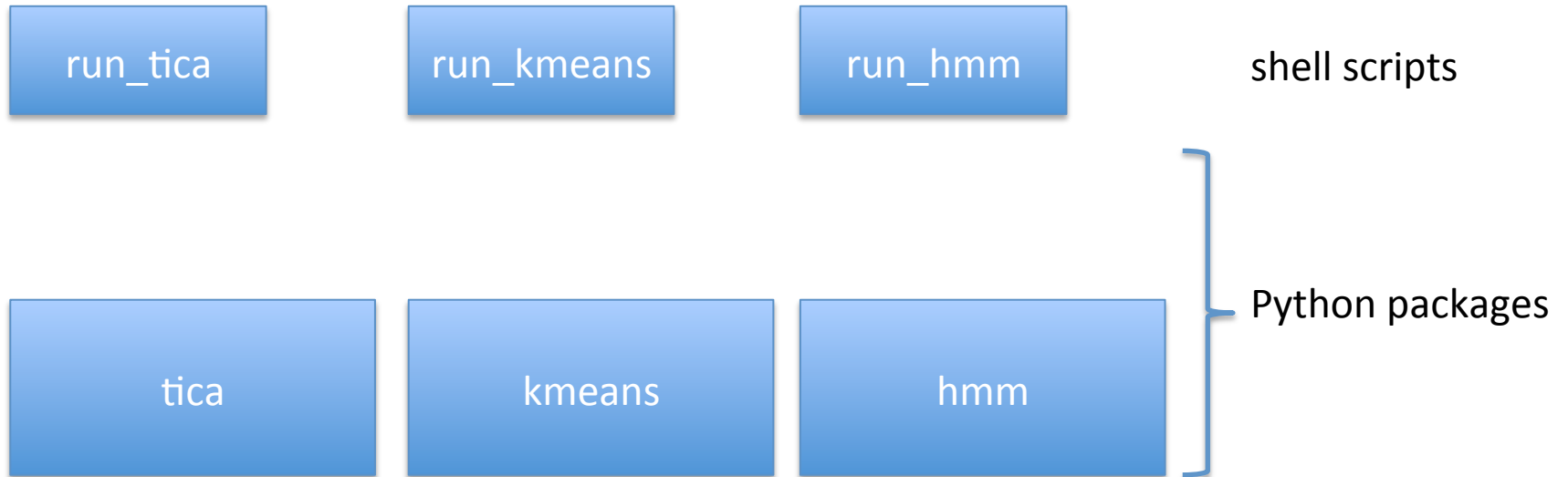
software design / continuous integration
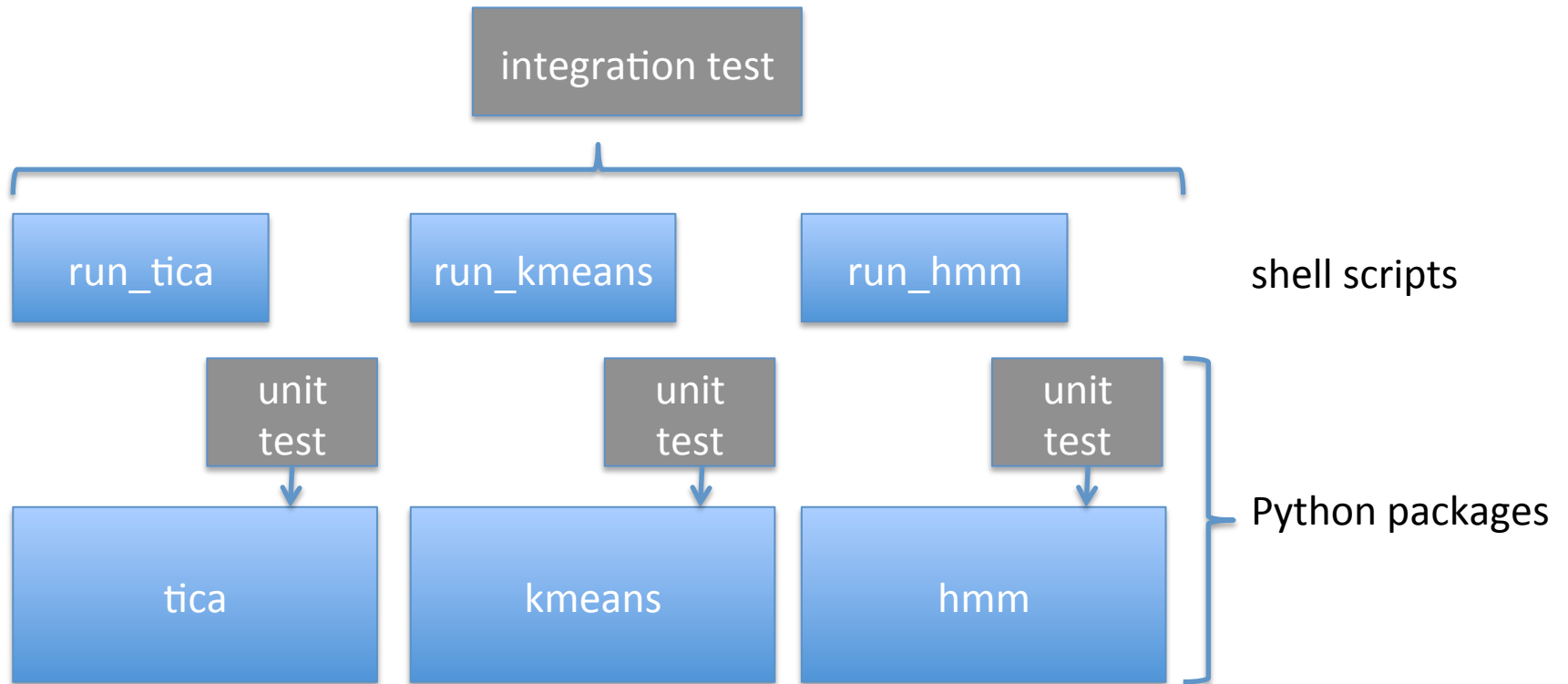
software design / continuous integration
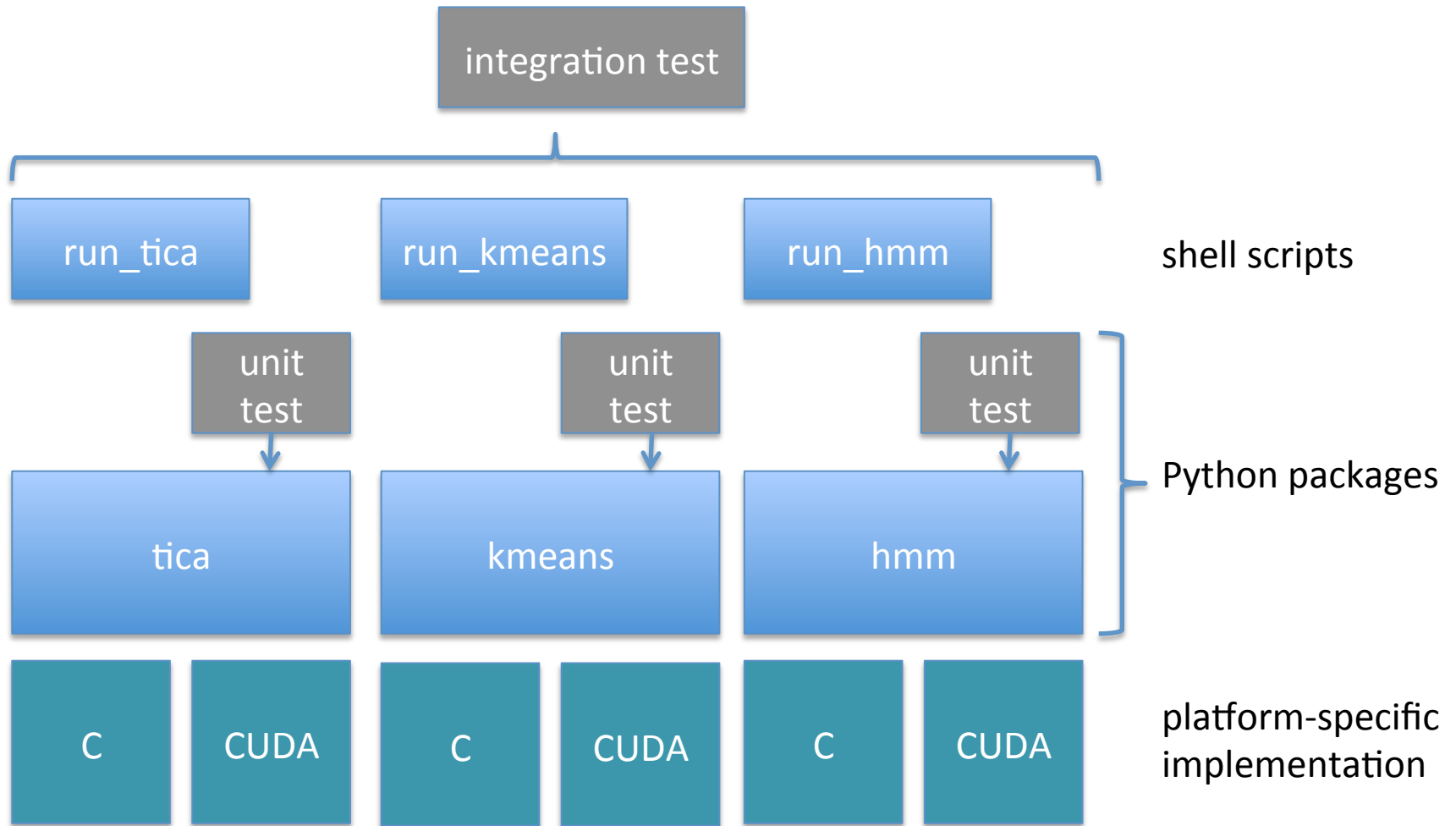
Meetings

- Suggestion: Every two weeks, roughly for 2 hours

Assignments for next meeting (Oct 31st)

- All: Decide if you are in!
- Form teams: TICA (2+), k-means (2+), HMM (3+)
- Every team should have at least one person designing and implementing tests
- Every team should have one representative
- Elect one or two project leaders.
- Project leader(s) are github administrators and create a repository
- I upload three papers to the repository (theory)
- Make sure all project members can communicate and get technical help from others (mailing list)
- All: Get github accounts, make sure you have push+pull rights in the project
- Read and understand theory
- Whenever possible, start with python prototypes (you can use any availble code or package for the prototype!)