Gromacs - where are we now, what's new, what's coming in the future

Christian Blau

blau@kth.se

GROMACS origins

- Originally a hardware project at the University of Groningen in the early 1990s – custom-designed 32-processor ring using PVM
- Design and simulation methodology largely adopted from existing GROMOS molecular simulation package
- Key decision: develop in ANSI C, rather than FORTRAN
- Early adopters and core developers:
  - David van der Spoel, Berk Hess, Anton Feenstra, Rudi van Drunen

GROMACS origins

- First public release – 2.0 some time in the 1990s
- First paper – “GROMACS: a message-passing parallel molecular dynamics implementation” Comp. Phys. Comm. 1995
- first CVS commit in 1997, already 750 files and 145k LOC
- first FOSS release – 3.0 in August 2001, GPL 2.0
- provide preparation and analysis tools along with the core simulation engine
- aim for performance portability x86 assembly for key kernels

Who develops GROMACS

- Developers in Stockholm / Uppsala
  - Mark Abraham (project manager)
  - Paul Bauer (project/release manager)
  - Szilard Pall (GPU acceleration)
  - Artem Zhmurov (GPU acceleration)
  - Berk Hess (enhanced sampling, domain decomposition, algorithm development)
  - Erik Lindahl (SIMD, PME, large scale project coordination)
  - David van der Spoel (Energy analysis, analysis tools, force field development)
  - Joe Jordan (NB-LIB)
  - Christian Blau (external forces)

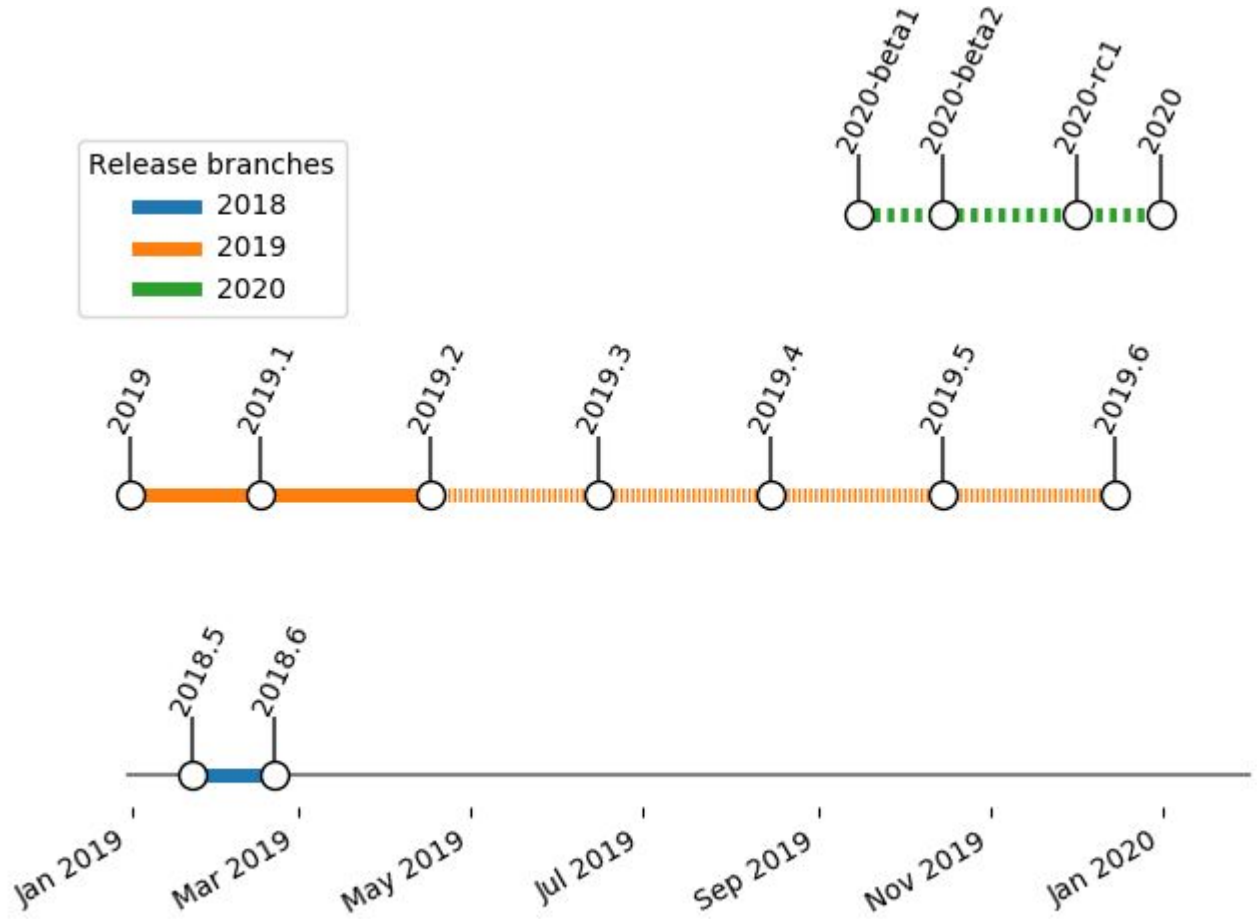
Who develops GROMACS

- Developers in Finland
  - JYU Jyväskylä (Gerrit Groenhof, Dmitry Morozov, Noora Aho, Emmi Pohjolainen)
    - constant PH, QM/MM
  - Teemu Mortula (options, selections, analysis tools, C++ transition)
- Developers in Germany
  - Carsten Kutzner (MPI-BPC Göttingen, ion swapping, enforced rotation)
  - Thomas Ullman (MPI-BPC Göttingen, lambda dynamics)
  - Vytautas Gapsys (MPI-BPC Göttingen, pmx, free energy)
  - Markus Rampp group (MPI-Computing and Data Facility)

Who develops GROMACS

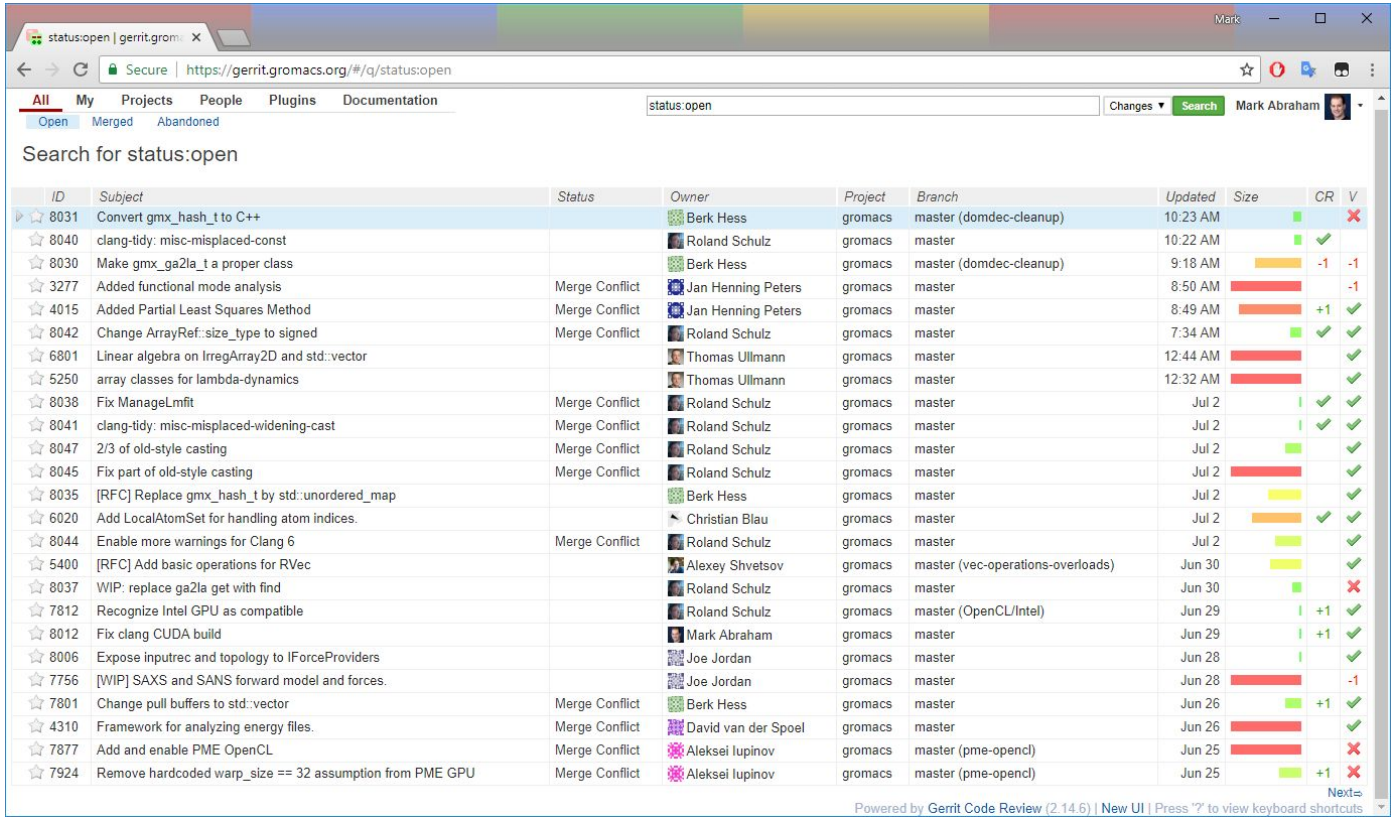
- US and others
  - Roland Schulz (@intel, parallelisation and acceleration, C++ transition)
  - Alan Gray, Jonathan Vincent (@NVidia, GPU acceleration)
  - Eric Irrgang, Peter Kasson (python api, ensemble simulations)
  - Pascal Merz, Michael Shirts (general integrator framework)
  - Kevin Boyd (code refactoring)

GROMACS development cycles



How to contribute

- Know what you want
- Ask developers
  - personally
  - mailing list
  - redmine
- Describe and document your feature
- Have proof of concept
- Upload to gerrit
- Make small commits



Documentation is a must

- All code has to be documented using Doxygen
    - Allows automatic generation of webpage content
    - Can be cross referenced in other parts
    - Will be used to determine public and internal API
- General documentation using ReStructured Text (rst)
    - Generates all remaining webpages and PDF manual
    - Makes it possible to have cross referencing between the sections
- If you use or develop for GROMACS, the manual is the best way to get started

GROMACS user and developer mailing lists

- General questions should be asked on user mailing list
    - gmx-users@gromacs.org
- Check first the archive if the questions have been asked
- Developer questions better suited at
    - gmx-developers@gromacs.org
- Plan to move both to a BioExcel organized forum
    - ask.bioexcel.eu

New features in GROMACS 2020

- Density-guided simulations
- Python api to gromacs: gmxapi 0.1
- Modular simulator
- Parrinello-Rahman pressure for the md-vv integrator.
- Performance enhancements
  - Speed-up for free energy simulations
  - More SIMD acceleration
  - own-FFTW configured better
  - Faster bonded interactions on GPUs

Development challenges (why don't you just...)

New code needs to be

- Maintainable
  - Understanding and working on a piece of code may not be tied to a single person
  - everything will come to bite us eventually
- Portable
- Scientifically correct
- Performant

How to add good new code

- Have a case
  - Start with the minimal unit that is useful
  - Well cited papers help
- Make a case
  - Write a careful complete documentation of what you want
  - discuss with other developers
- Have tests
  - Write small tested functional units
  - Write a regression test, where all of your functionality plays together
- Avoid complex interactions
  - Touch as little code as possible
- Provide generally useful infrastructure
  - Start with for necessary code that will benefit the rest of GROMACS

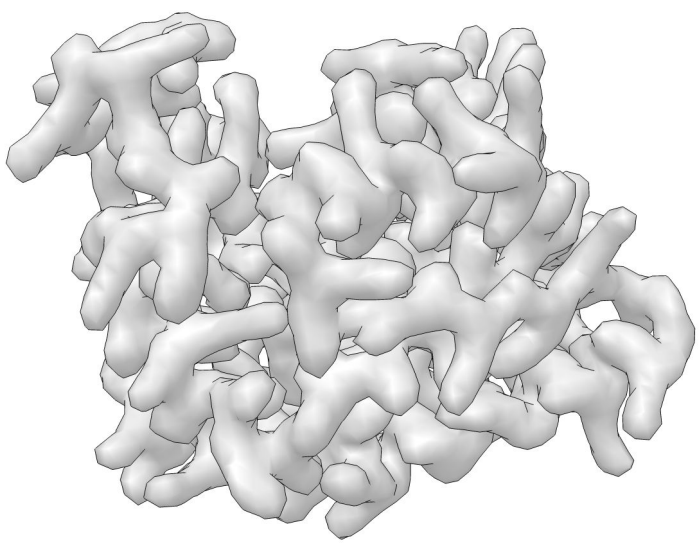
New in GROMACS 2020 - under the hood

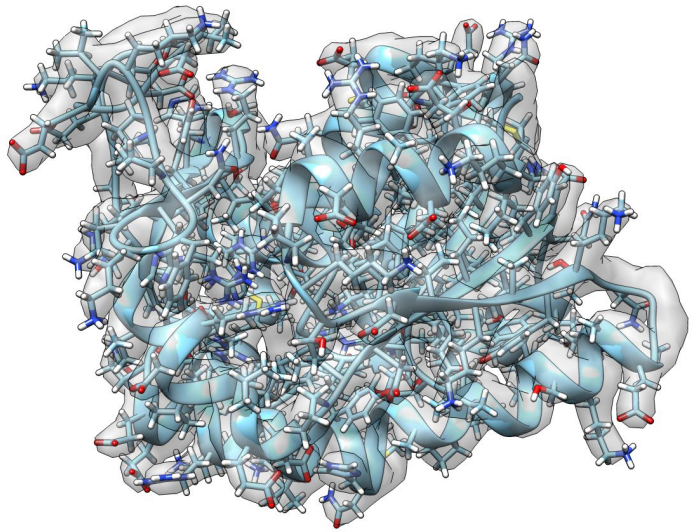
Extensive code refactoring

- More tests
- Removal of the “group” scheme - loads of lines of code removed :)
- Domain-decomposition refactoring
- GPU code and communication refactoring
- extending the MDModule functionality
- modern Multidimensional arrays (mdspan)

GROMACS 2020 - density-guided simulations

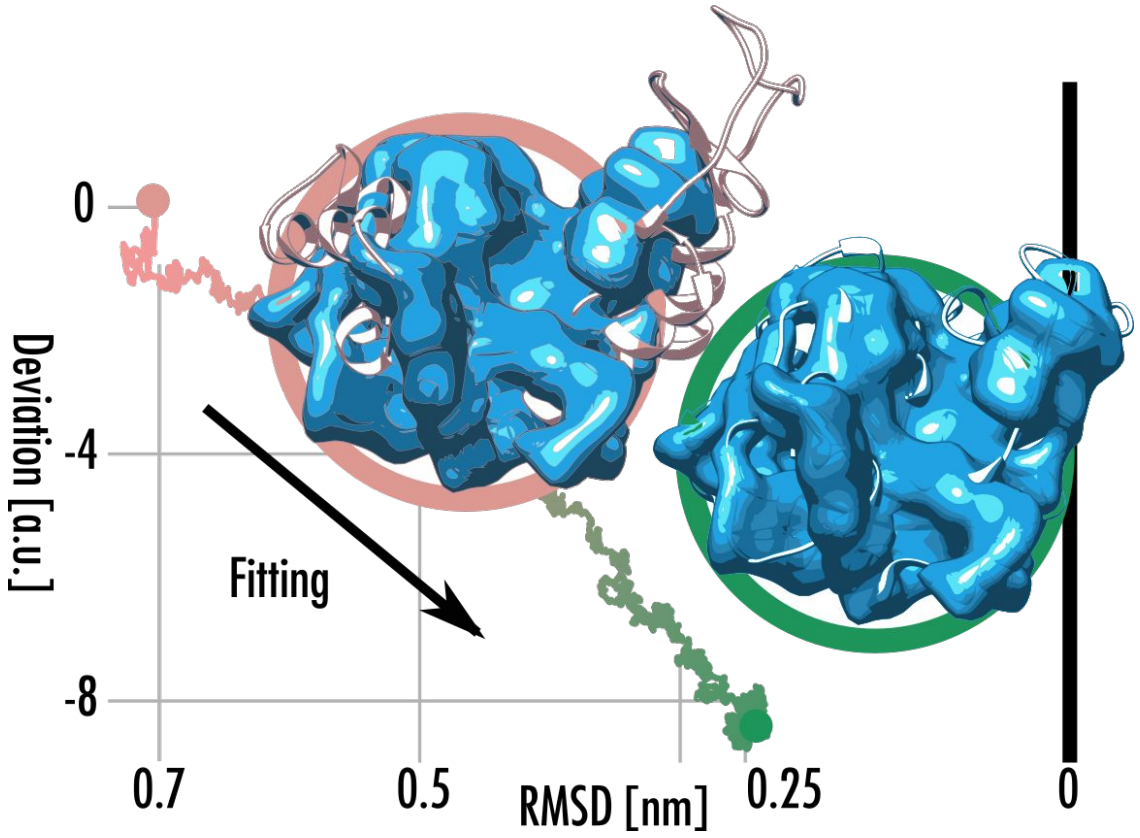
- Making the most use of cryo-EM densities
  - Automated density fitting to speed up time-to-result in the modeling process
  - Explore the all the structures that fit a given cryo-EM density
  - Allowing to balance between bio-chemistry and cryo-EM data influence
  - See structural transitions between states





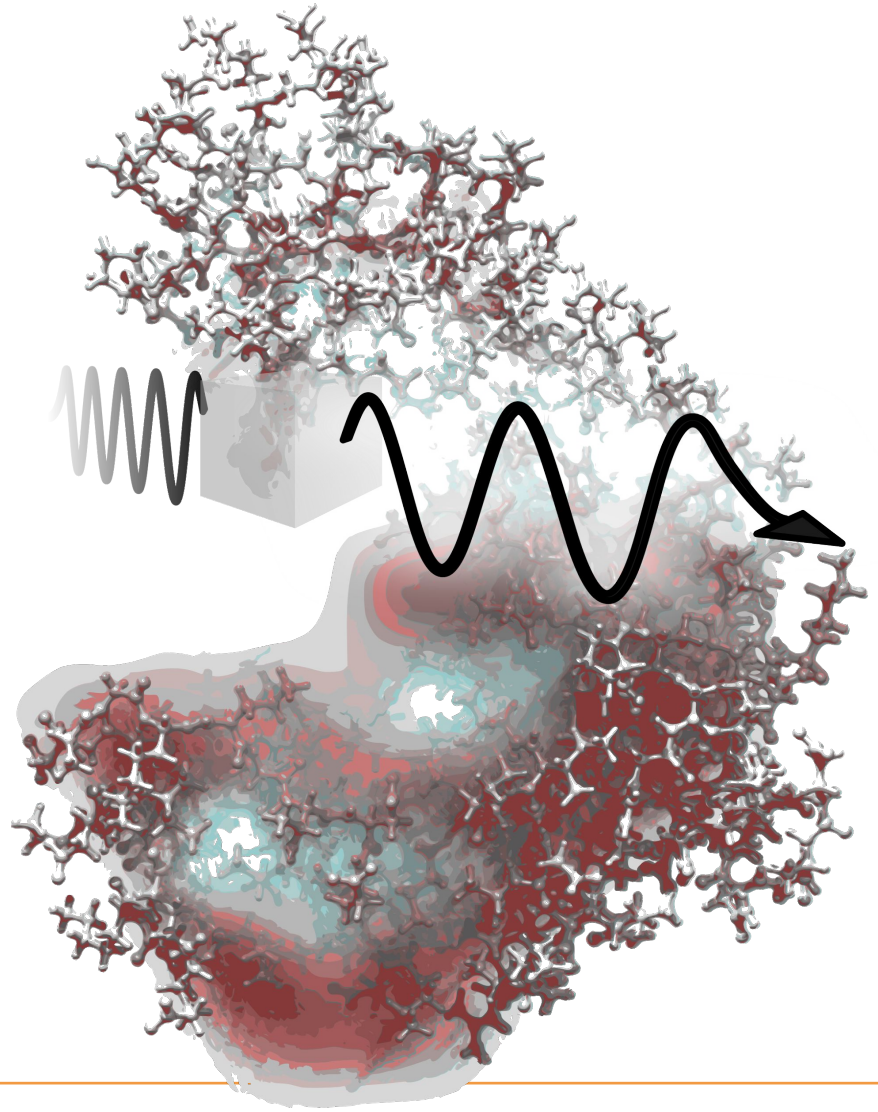
GROMACS 2020 - density-guided simulations

- The pipeline
  - set up a vanilla molecular dynamics simulation
  - add a reference density (.mrc or .ccp4 file)
  - choose balance between force field and density information
  - convergence usually within a day of walltime running



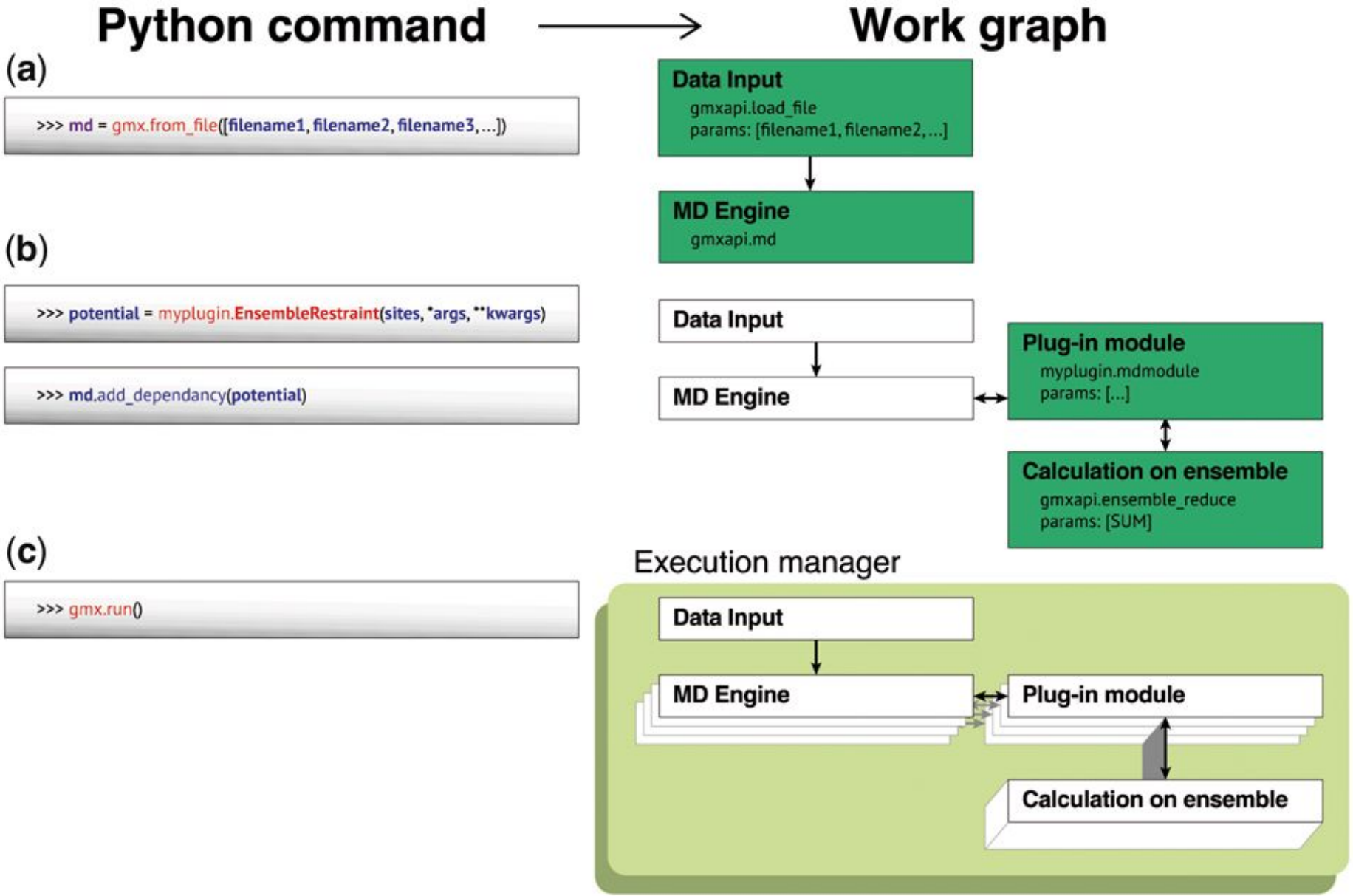
GROMACS 2020 - density-guided simulations

- Further features
  - force calculation derived from modeling underlying physics
  - different models of comparing densities (relative entropy, cross-correlation,...) allows fine-tuning depending on the research question
  - not only limited to cryo-EM, but applicable to any density / shape information



GROMACS API

- Running simulations from a python interface
- “Simple things should be simple, hard things not impossible”
- Vision: More features with less effort, shared in a community at maximum performance
- Requirement: Extensive code refactoring


Python command
Work graph
(a)
>>> md = gmx.from_file([filename1, filename2, filename3, ...])
Data Input
gmxapi.load_file
params: [filename1, filename2, ...]
MD Engine
gmxapi.md
(b)
>>> potential = myplugin.EnsembleRestraint(sites, *args, **kwargs)
>>> md.add_dependancy(potential)
Data Input
MD Engine
Plug-in module
myplugin.mdmodule
params: [...]
Calculation on ensemble
gmxapi.ensemble_reduce
params: [SUM]
(c)
>>> gmx.run()
Execution manager
Data Input
MD Engine
Plug-in module
Calculation on ensemble

GROMACS interfaces workflows

- Wrapping gromacs is (comparably) easy, maintaining performance and sanity is hard
- Reproducible science workflows require reproducible workflows
- Challenges
  - Define interfaces
  - Be general, but use specific hardware features
  - Manage data

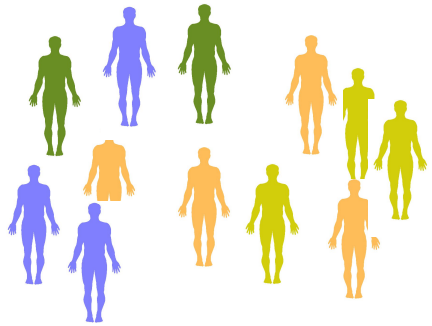
Who enables GROMACS

- Funding agencies - BioExcel, PRACE, …
- GROMACS community - research groups, “Freelancers”
- Compute centers
- …

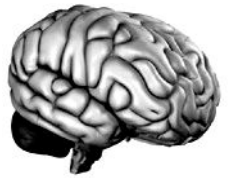
Future of GROMACS hinges on

- Code and Research
- Training and Dissemination
- Qualified personnel and funding

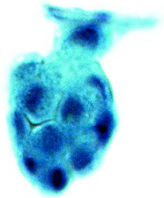
Life Science and HPC



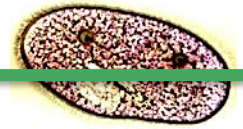
Population



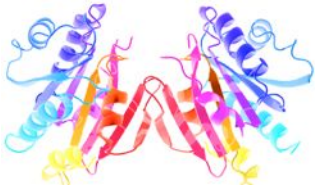
Organ



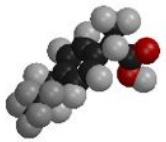
Tissue



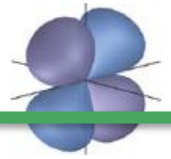
Cell



Macromolecules



Small Molecule



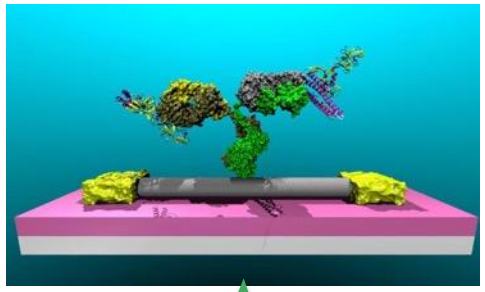
Atom

BioExcel
Area of Expertise

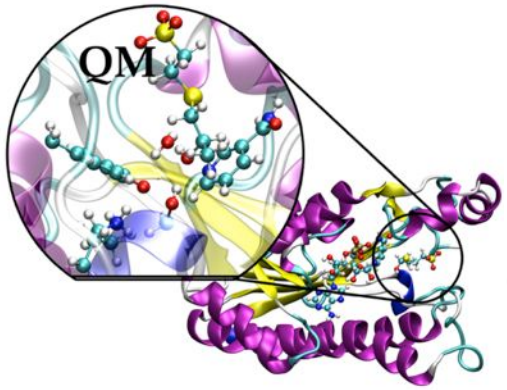
bioexcel

Biomolecular Modeling and Simulations

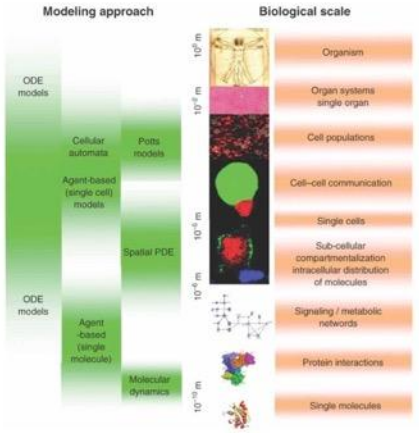
Biomarkers design

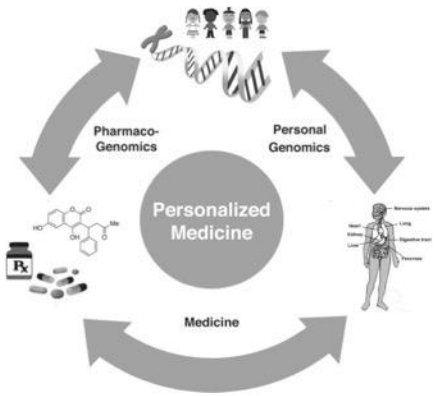


Electronic structure

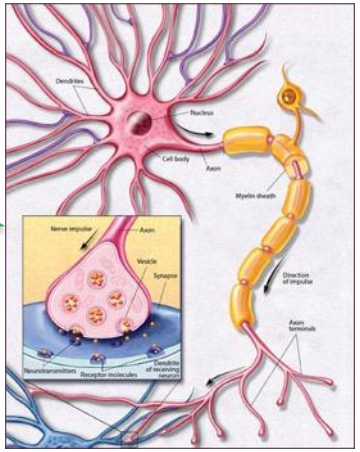


Physiology

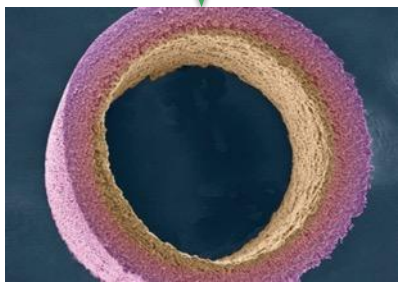




Drug Design
Personalized Medicine



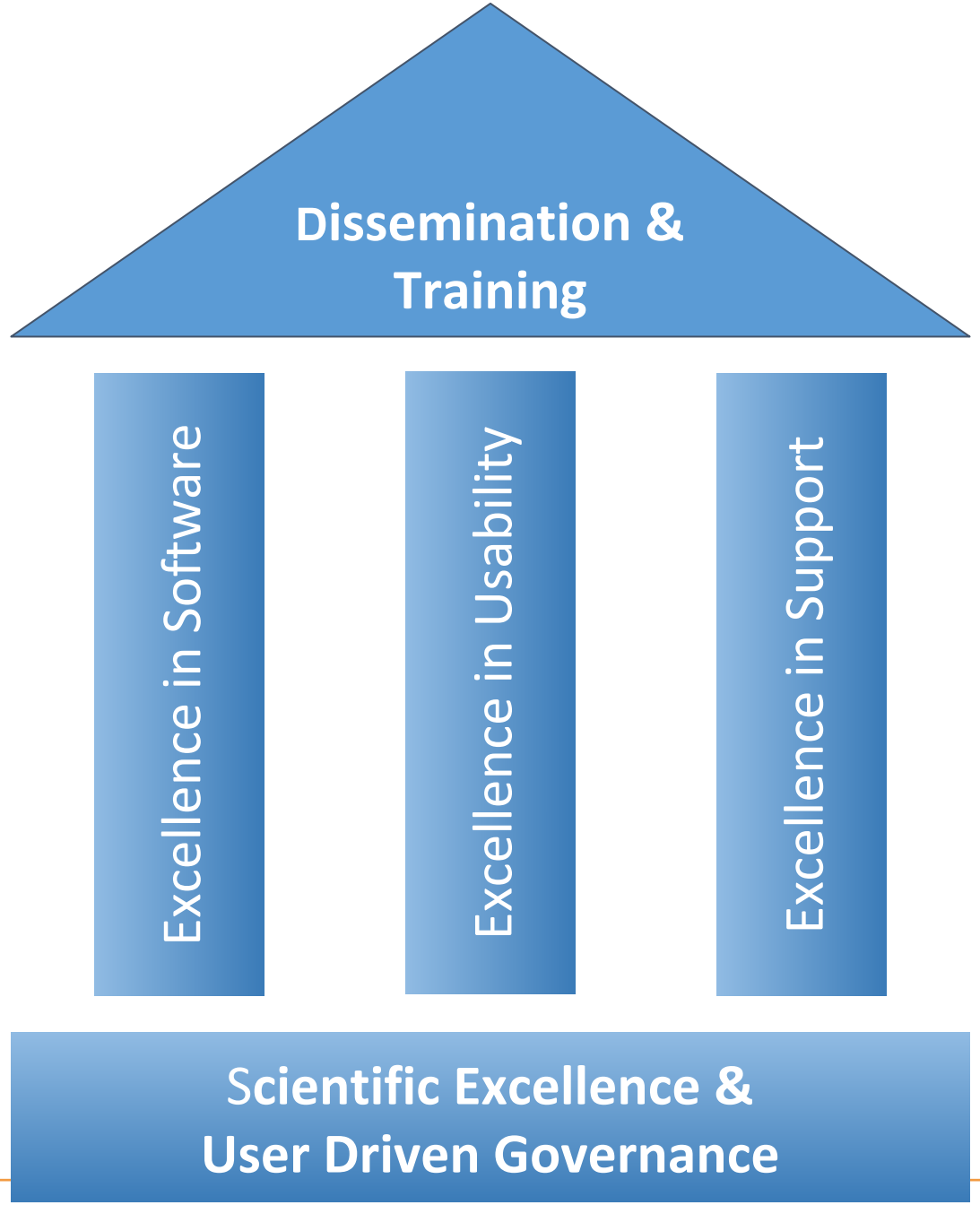
Neuroinformatics



Biomaterials science and

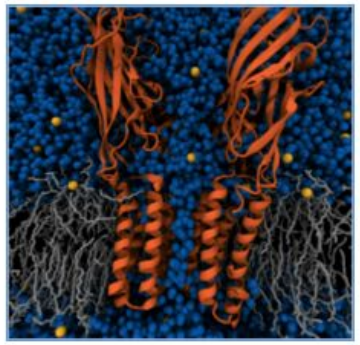
A central hub for biomolecular modeling and simulations

Enabling better science by **improving** the most popular biomolecular **software** and spreading best practices and expertise among the communities through **consultancy** and **training**

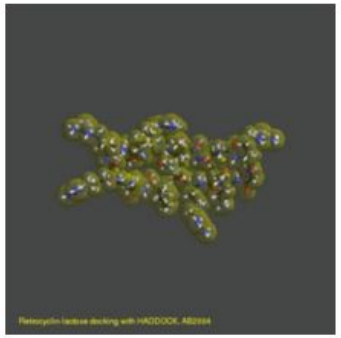

Dissemination &
Training
Excellence in Software
Excellence in Usability
Excellence in Support
Scientific Excellence &
User Driven Governance

Improve the performance, efficiency and scalability of key codes

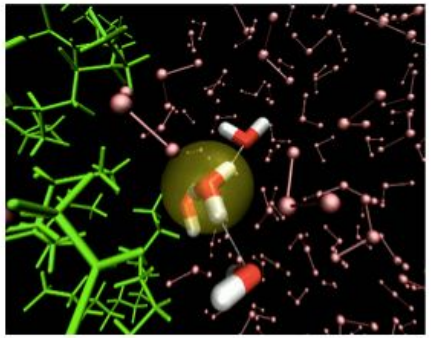
- GROMACS (Molecular Dynamics Simulations)
- HADDOCK (Integrative modeling of macro-assemblies)
- CP2K (hybrid QM/MM code for enzymatic reactions, photochemistry and electron transfer processes)



MD simulations
/GROMACS/



Docking
/HADDOCK/



QM/MM
/CP2K/

Training and workshops

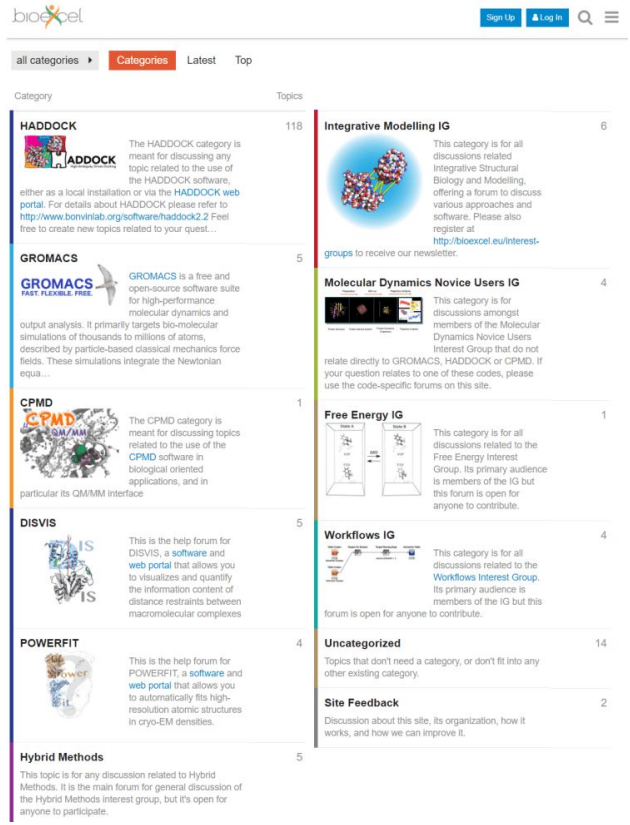
- Closing the “knowledge gap”
  - show users what is available
  - transferring new software developments to users
    - enable users to apply the newest and most advanced features
  - ensure software is used in the best possible way
    - scientific correctness
    - high performance
    - ease-of-use
  - follow up on user questions
  - feed-back into the development process

Training and workshops

- BioExcel workshops
  - focused events for 15-50 participants from beginners to advanced levels
  - industry participation is encouraged
  - combination of BioExcel tools
  - selection of events organized or supported by BioExcel
    - Minisymposium: Biomolecular simulation in drug discovery (with Scilifelab), Stockholm
    - Alchemical Free Energy Workshop 2019, Göttingen, Germany
    - BioExcel/PRACE Seasonal School, Stockholm
    - BioExcel Summer School on Biomolecular Simulations, Pula, Italy
    - Advanced Gromacs, HADDOCK + PMX workshop, Espoo, Finland

Training and workshops

- Webinars
  - dozens of sessions on all topics related to BioExcel from free energy calculations to software development at https://bioexcel.eu/category/webinar/
  - Session every other week
    - enables discovery new aspects of BioExcel tools
    - deepen understanding of use
- BioExcel forum
  - ask.bioexcel.eu
- Knowledge resource center with tutorials
  - krc.bioexcel.eu



bioexcel

Centre of Excellence for Computational Biomolecular Research

BioExcel Partners 2019

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

NORMAN CONSULTING

EMBL-EBI

KTH VETENSKAP OCH KONST

THE UNIVERSITY OF EDINBURGH

JÜLICH FORSCHUNGSZENTRUM

IRB BARCELONA
INSTITUTO DE INVESTIGACIÓN BIOMÉDICA

Utrecht University

ih Ian Harrow Consulting

acrosslimits

MAX-PLANCK-GESELLSCHAFT

MANCHESTER 1824

UNIVERSITY OF JYVÄSKYLA

NBD NOSTRUM BIODISCOVERY

Horizon 2020
European Union Funding
for Research & Innovation

BioExcel is funded by the European Union Horizon 2020 program under grant agreements 675728 and 823830.