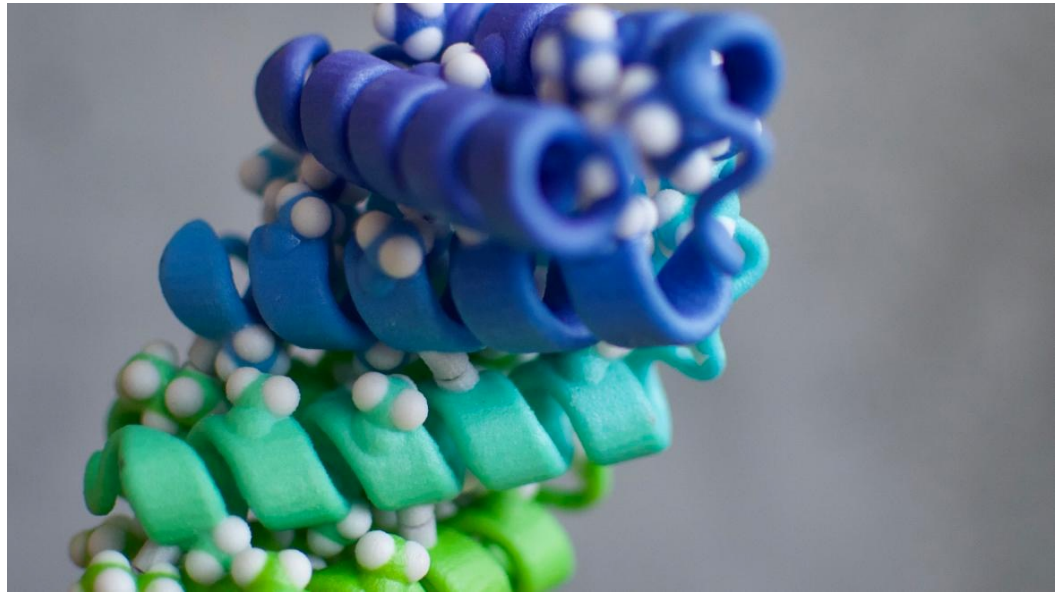
The Necessity for RDM in Computational Protein Design



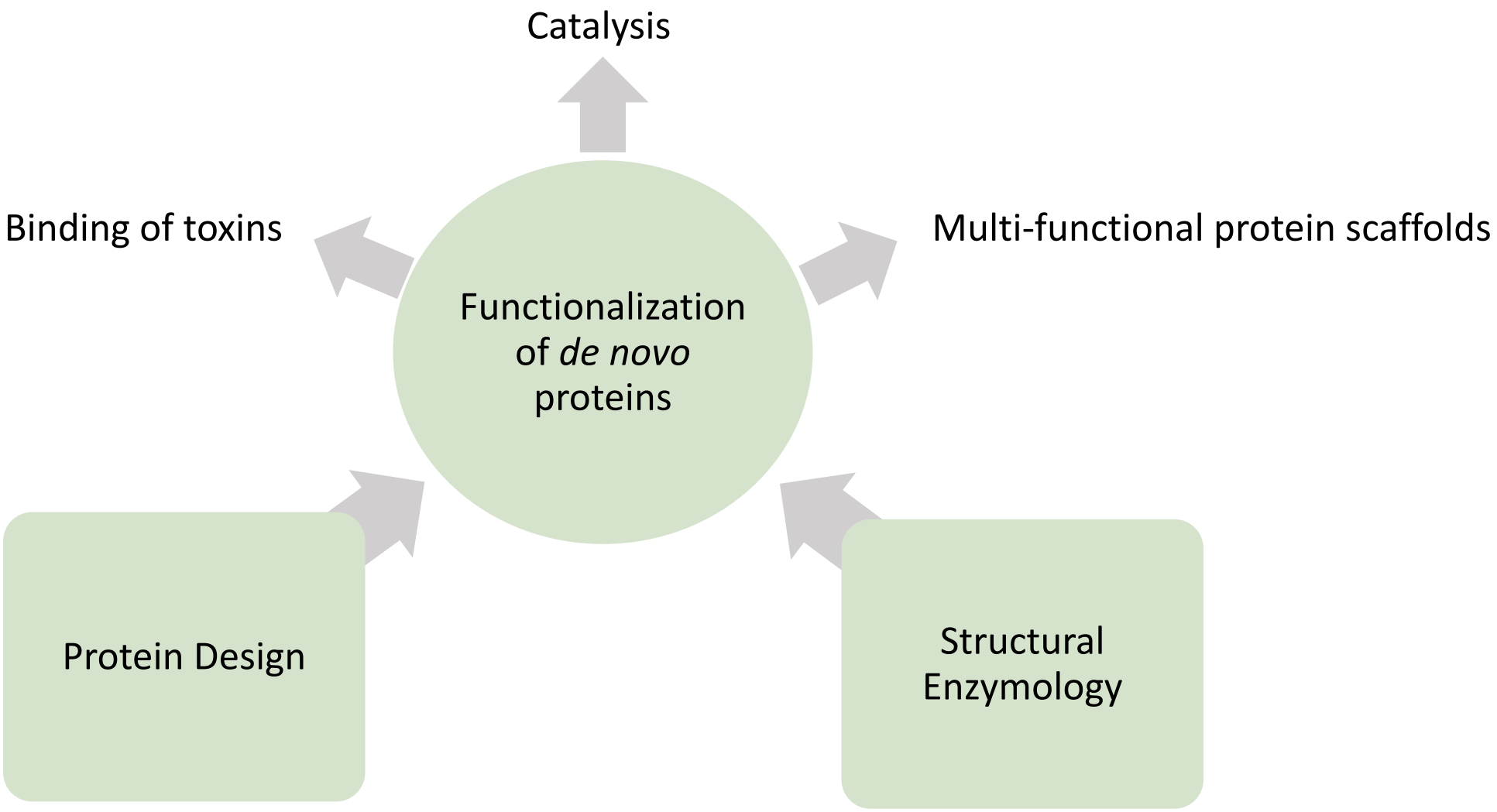
Gustav Oberdorfer

Graz University of Technology, Austria

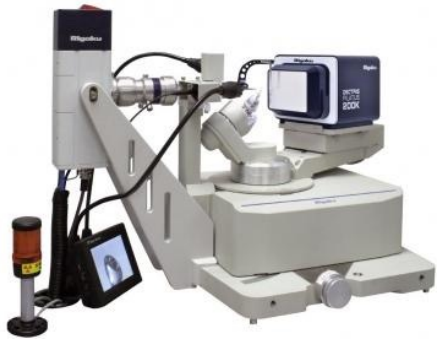
June 8th 2021

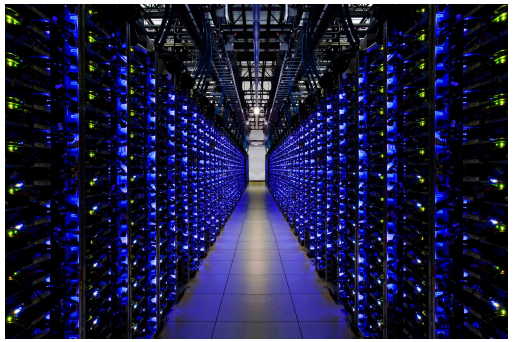
Graz University of Technology

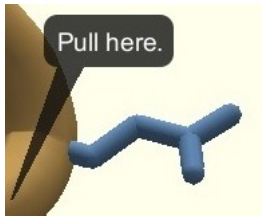
Routinely and robustly design catalytic or small molecule binding proteins



Methods and Techniques used



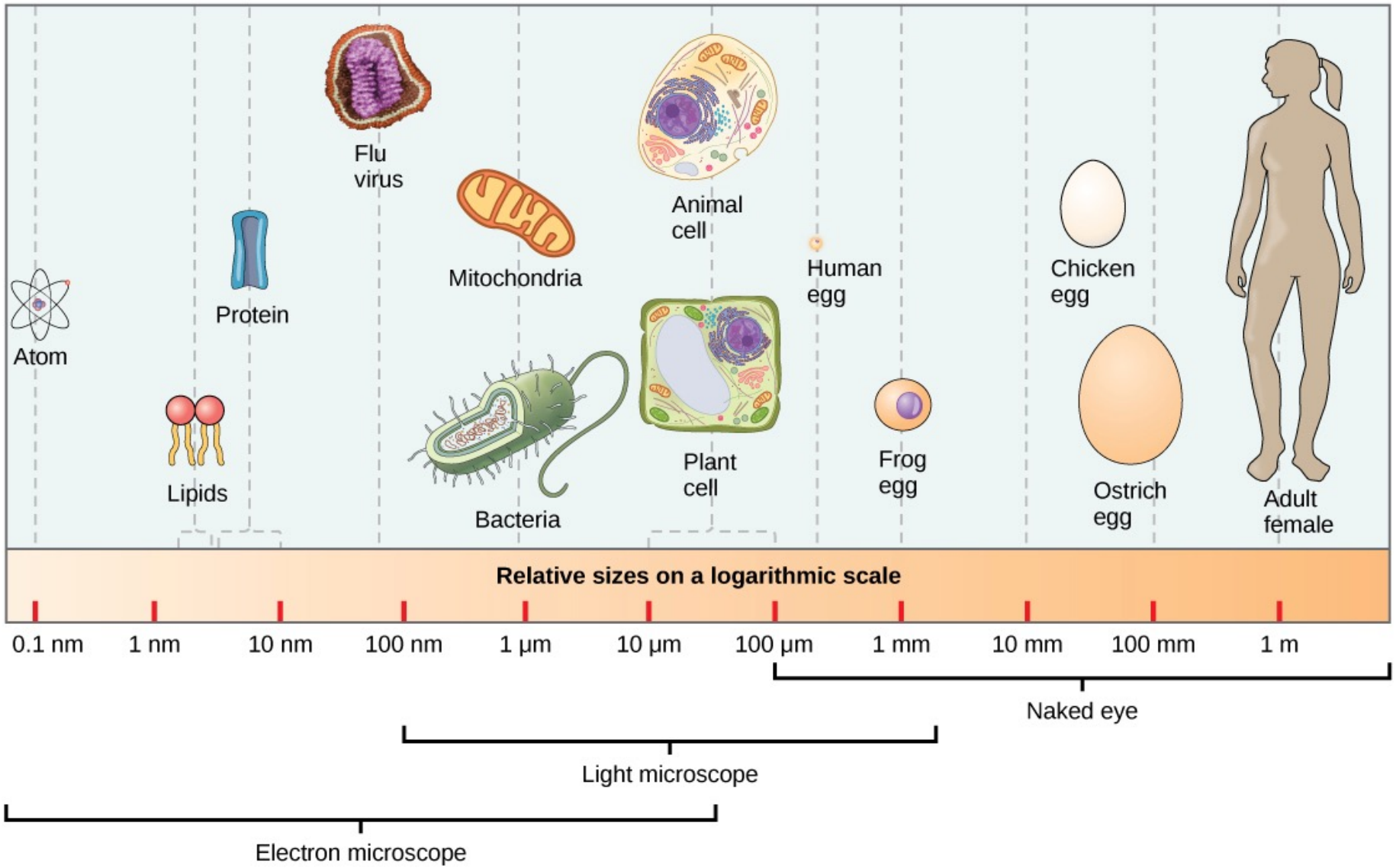




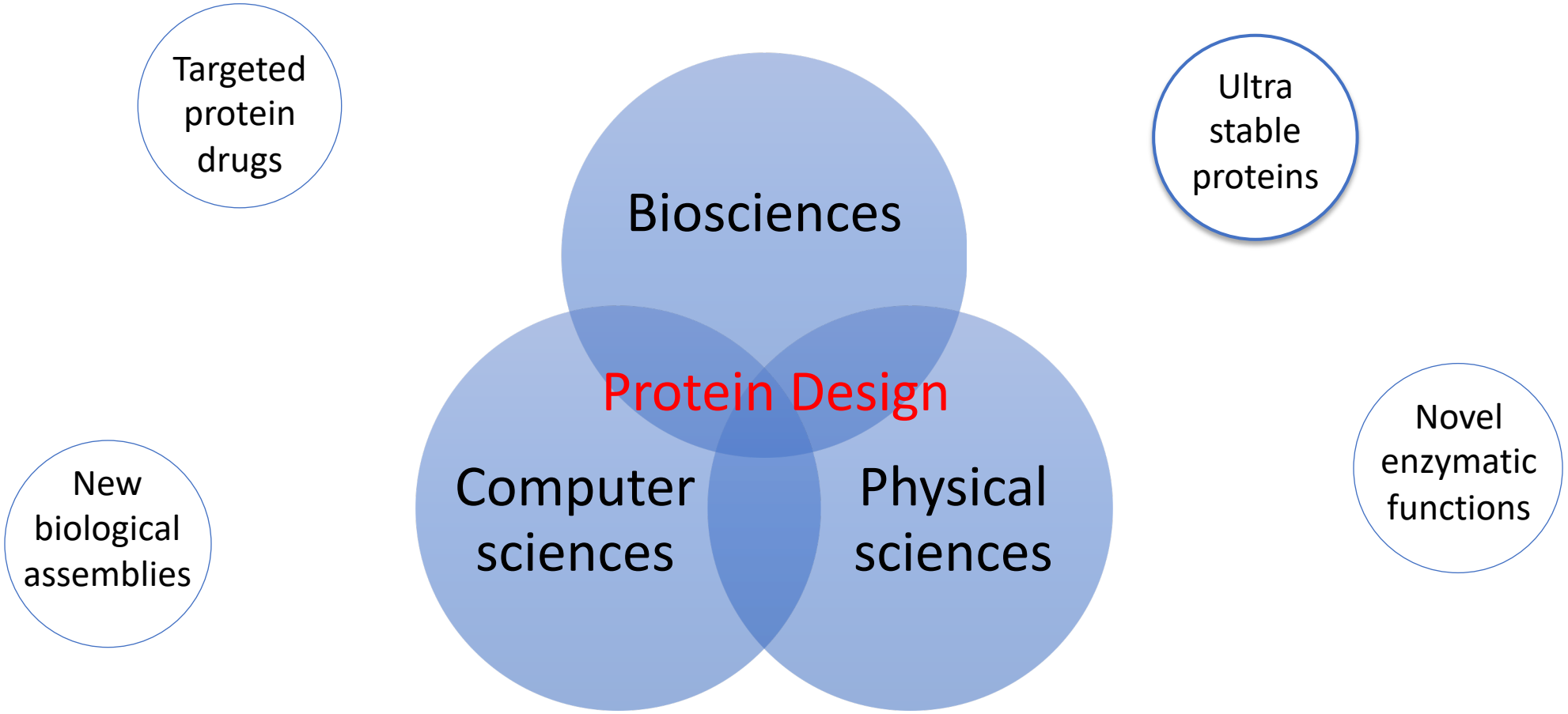
- Gene assembly
- Protein expression
- Biochemical, biophysical, functional and structural characterization (X-ray crystallography, Cryo-EM)

- Protein design calculations
- Molecular modeling
- Toolkit development

Protein design is a bottom-up approach that operates at the angstrom to nanometer level



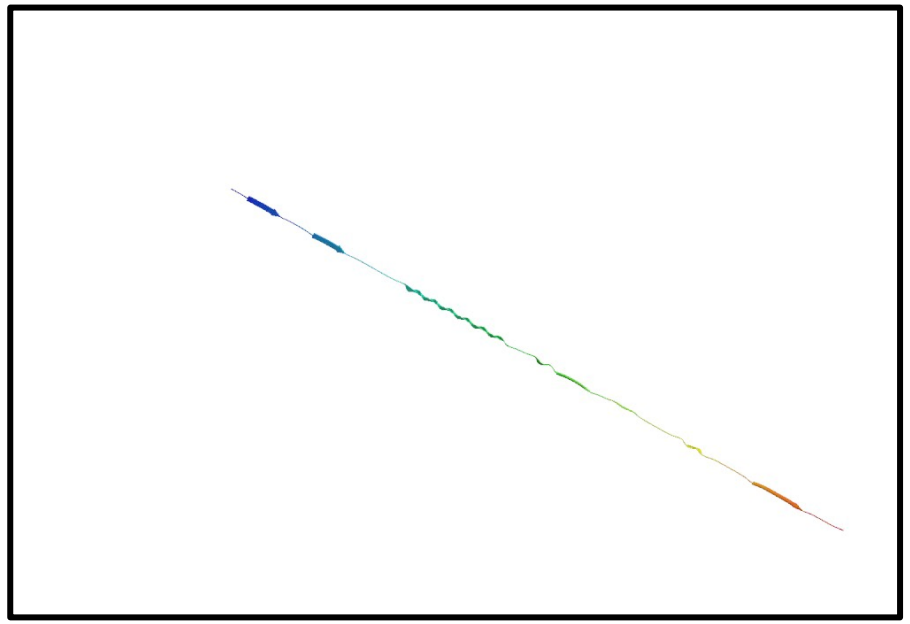
Computational protein design is a highly interdisciplinary effort



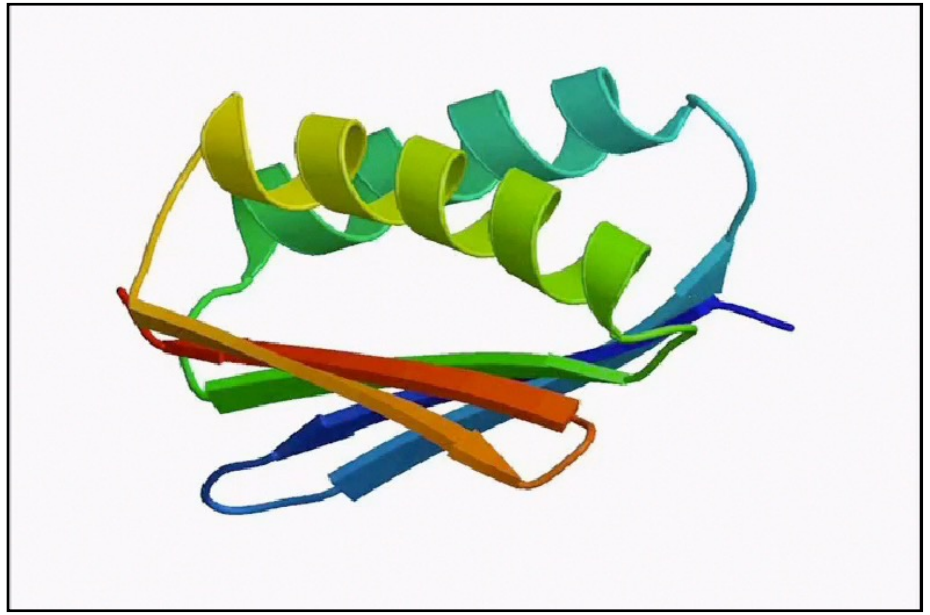
Searching for an energy **minimum**

Rosetta

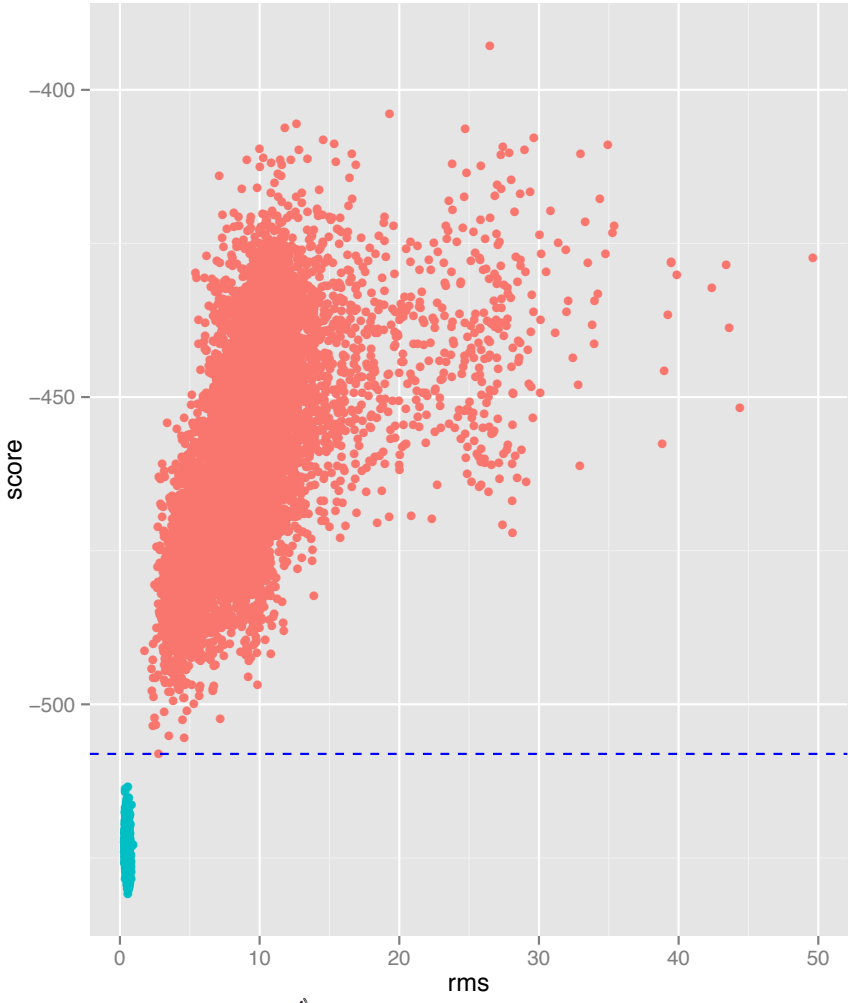
Protein structure prediction

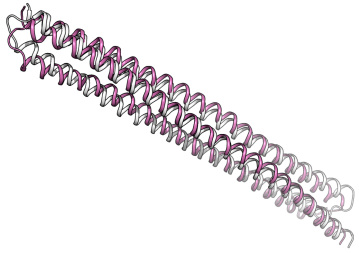


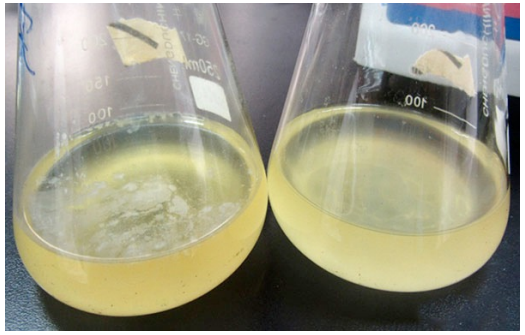
Protein structure design

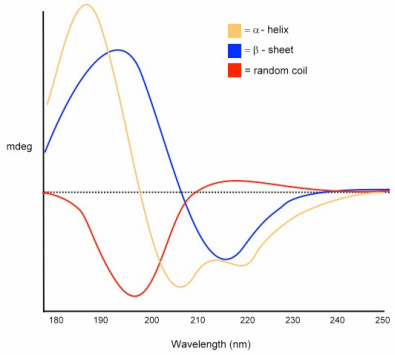


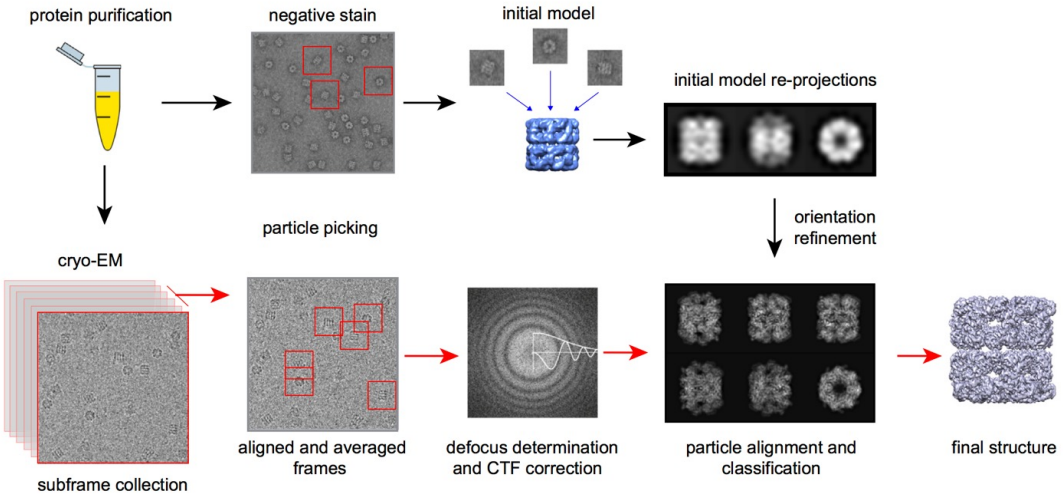
General protein design workflow

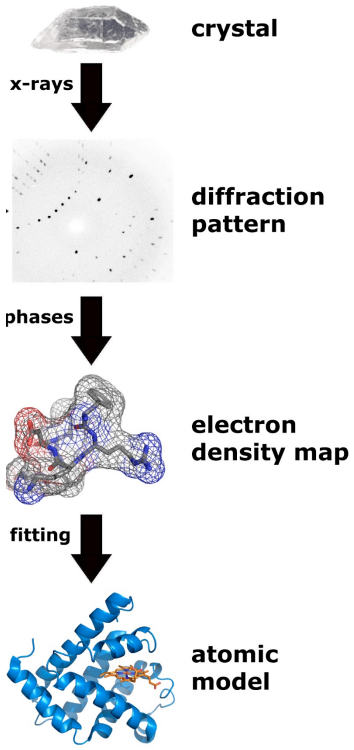












How we use managed data in the lab

WORLDWIDE PDB PROTEIN DATA BANK

SCOP

CATH

Big benefit: they all are indexed and use persistent identifiers!
Most of them (protein related) are cross-referenced

GitHub

PubMed

NCBI

How we manage our own data in the lab

Lab credo: **All data is universally available for everyone in the lab**

Lab-onboarding and a lab leaving document (e.g. where to store stuff)

Generate various kinds of data:

- Scripts (code in bash, python, xml, C++)
- Sequence Data (DNA and AA sequences of designed proteins)
- Protein expression (e.g. Images of SDS-Gels)
- Protein characterization (e.g. scattering and diffraction data)

Challenge 1: not all this data can be stored or treated the same way

Challenge 2: every project easily generates 100+ Gb of data

How we manage our own data in the lab

For computer related data:

- Use HPC resources to generate and analyze our data (TUG, CyVerse)
  - User group and individual users, standardized methods and workflows
- All data will be stored on a file-server with person-centric folder structure (currently stored on TUG cloud systems)
- All computationally derived data must include a log-file (includes identifiers for software versions used e.g.)
- In the process of building up in-house protein design database
- All program code is versioned and shared (group internally) via GitLab
- Raw data from experiments (X-ray diffraction, X-ray scattering etc) is stored on a file server with naming conventions
- Each group member is supposed to back-up their data to the central file server

How we manage our own data in the lab

For wet-lab related data:

- Coworkers are free to choose either lab-notebooks or a LIMS for documentation (LIMS: Benchling, open source)
- Lab internal protocols stored in a shared folder (will be transferred to a wiki soon)
- SOPs for machines and protein design approaches in place
- Database for strains, plasmids (includes QC-data in case of synthetic genes)

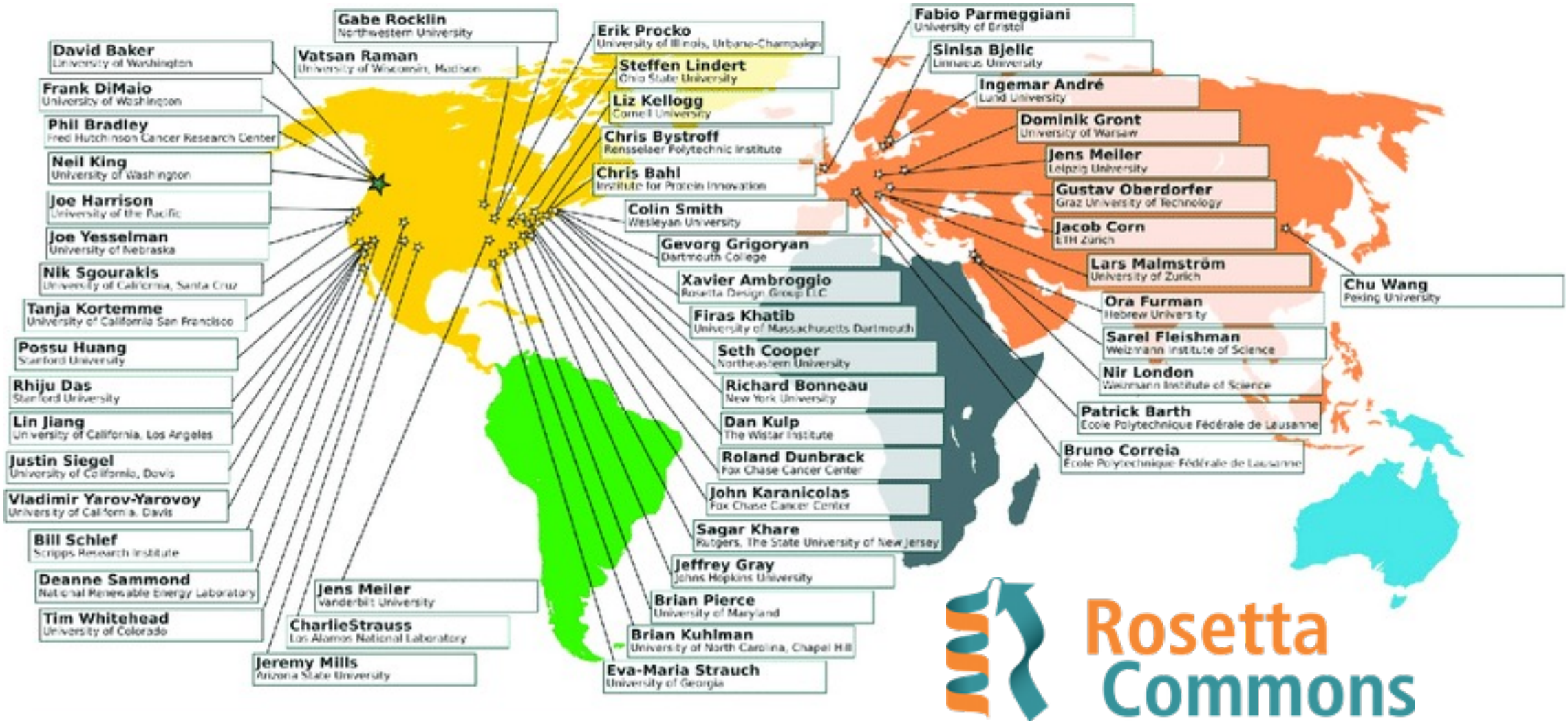
How we manage our own data in the lab

- Open-source formats wherever we can (e.g. fasta, pdb, mmcif, png, etc)
- All published data is made publicly available
  - We try to ensure the data is FAIR (Findable, Accessible, Interoperable, Reusable)

GitHub GitLab ProThermDB

WORLDWIDE PDB PROTEIN DATA BANK

How we manage data within the community



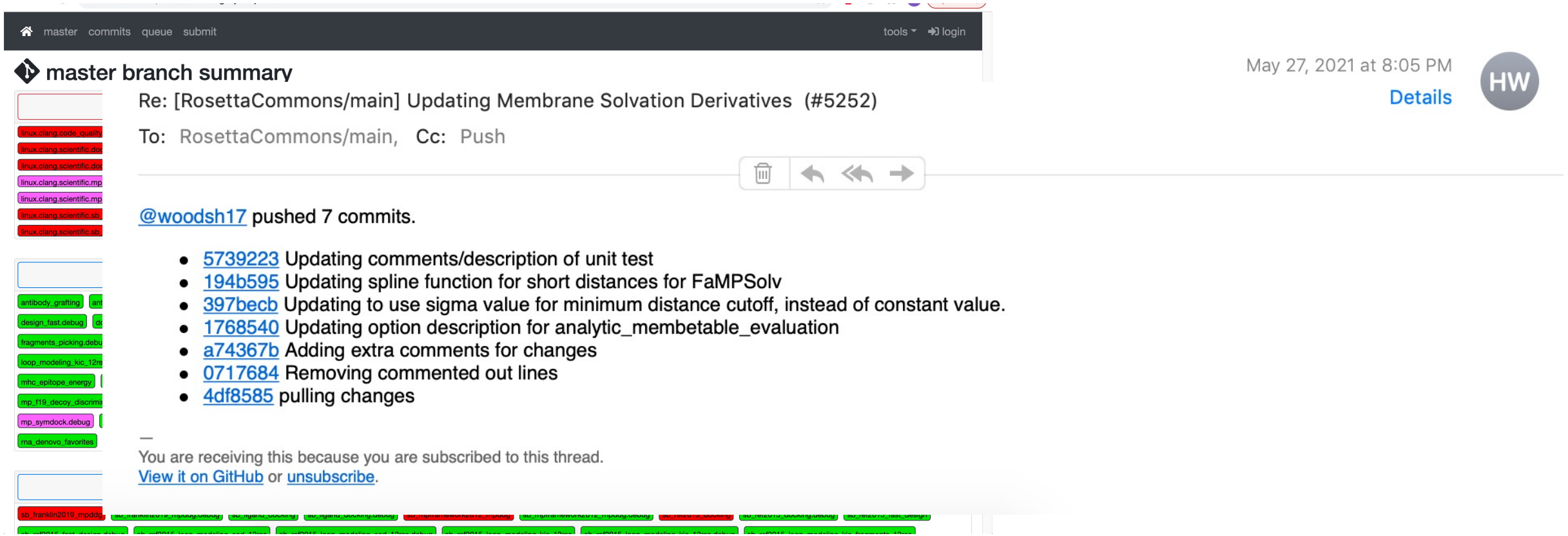
How we manage data within the community

- Coding conventions (e.g. descriptions, namespaces, pointers, indexing,…)

```
1  // -*- mode:c++;tab-width:2;indent-tabs-mode:t;show-trailing-whitespace:t;rm-trailing-spaces:t -*-
2  // vi: set ts=2 noet:
3  //
4  // (c) Copyright Rosetta Commons Member Institutions.
5  // (c) This file is part of the Rosetta software suite and is made available under license.
6  // (c) The Rosetta software is developed by the contributing members of the Rosetta Commons.
7  // (c) For more information, see http://www.rosettacommons.org. Questions about this can be
8  // (c) addressed to University of Washington CoMotion, email: license@uw.edu.
9
10 /// @file   protocols/forge/remodel/RemodelData.cc
11 /// @brief
12 /// @author Possu Huang (possu@u.washington.edu)
13 /// @author Yih-En Andrew Ban (yab@u.washington.edu)
14
```

How we manage data within the community

- Test servers (e.g. Integration test, scientific tests)
- Reviewing, pull-request conventions



How we manage data within the community

- Own metadata file format (silent files)

new.rosettacommons.org/docs/latest/rosetta_basics/file_types/silent-file

Silent File

Search… Home Feedback

Silent file is a compact format file which stores information from multiple structures. Silent files can be generated by many Rosetta simulations. Such as Abintio and Ligand Docking. You also can use silent file tools to combine regular pdb files to be binary format silent file to save computer storage spaces. We will briefly introduce silent file format here.

Protein Silent Struct is usually seen in the Abinitio outputs.

- Header

```
SEQUENCE                    Structure sequences presented by one letter
SCORE                       Rosetta score terms
```

- Body

```
Colunms 1-4                 Residue sequence number
Colunms 5-7                 Seconary structure one letter code
Colunms 8-17                Phi angle
Colunms 18-26               Psi angle
Colunms 27-35               Omega angle
Colunms 36-44               CA atom coordinates x
Colunms 45-53               CA atom coordinates y
Colunms 55-62               CA atom coordinates z
Colunms 64-98               Chi angle real data if possible
```

Binary Silent Struct File is very useful to compress multiple pdbs and save computer spaces.

TUGraz

Veronica Delsoglio
Wael Elaily
Stefanie Ferstl
Alexandra Grebe
Birgit Grill
Nina Grujicic
Horst Lechner
Alma Makic
Julia Messenlehner
Lena Parigger
Laura Rammer
Massimo G. Totaro
Adrian Tripp
Florian Wieser

Thanks for your attention

TU Graz
Graz University of Technology

erc
European Research Council
Established by the European Commission

802217 HelixMold

FWF

P 30826-B21

European Commission
HORIZON 2020

GA 863170 ArtiBLED