

The Necessity for RDM in Computational Protein Design

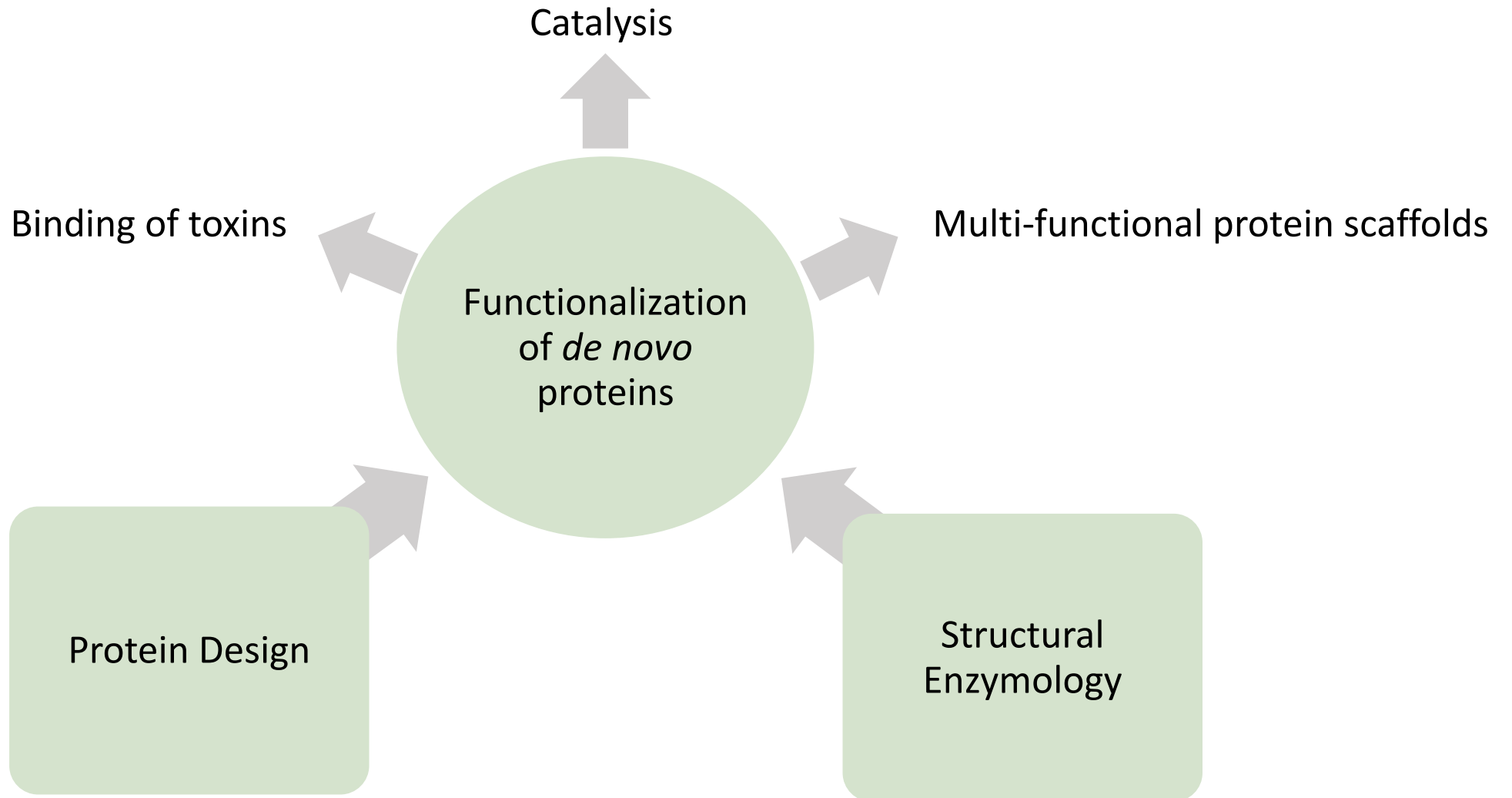


Gustav Oberdorfer

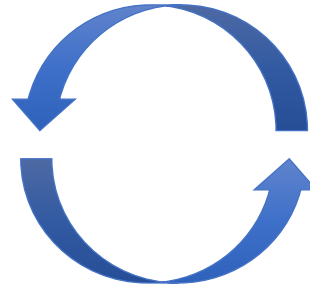
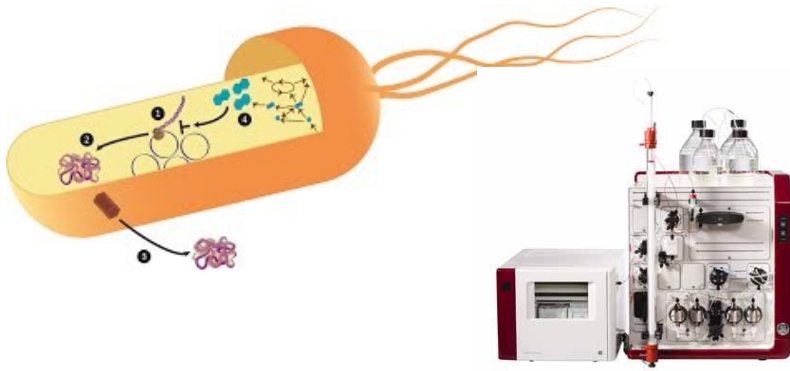
Graz University of Technology, Austria

June 8th 2021

Routinely and robustly design catalytic or small molecule binding proteins



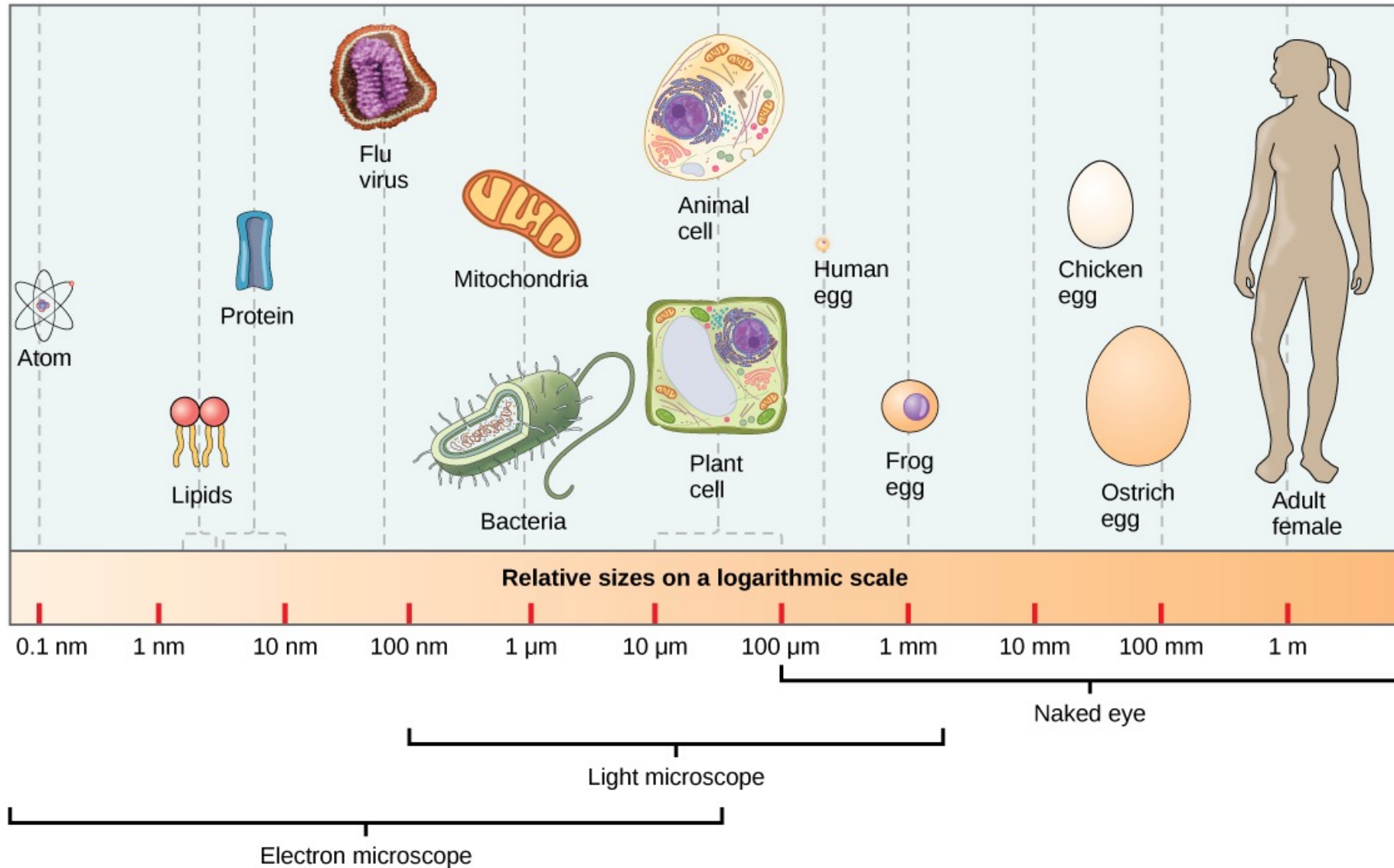
Methods and Techniques used



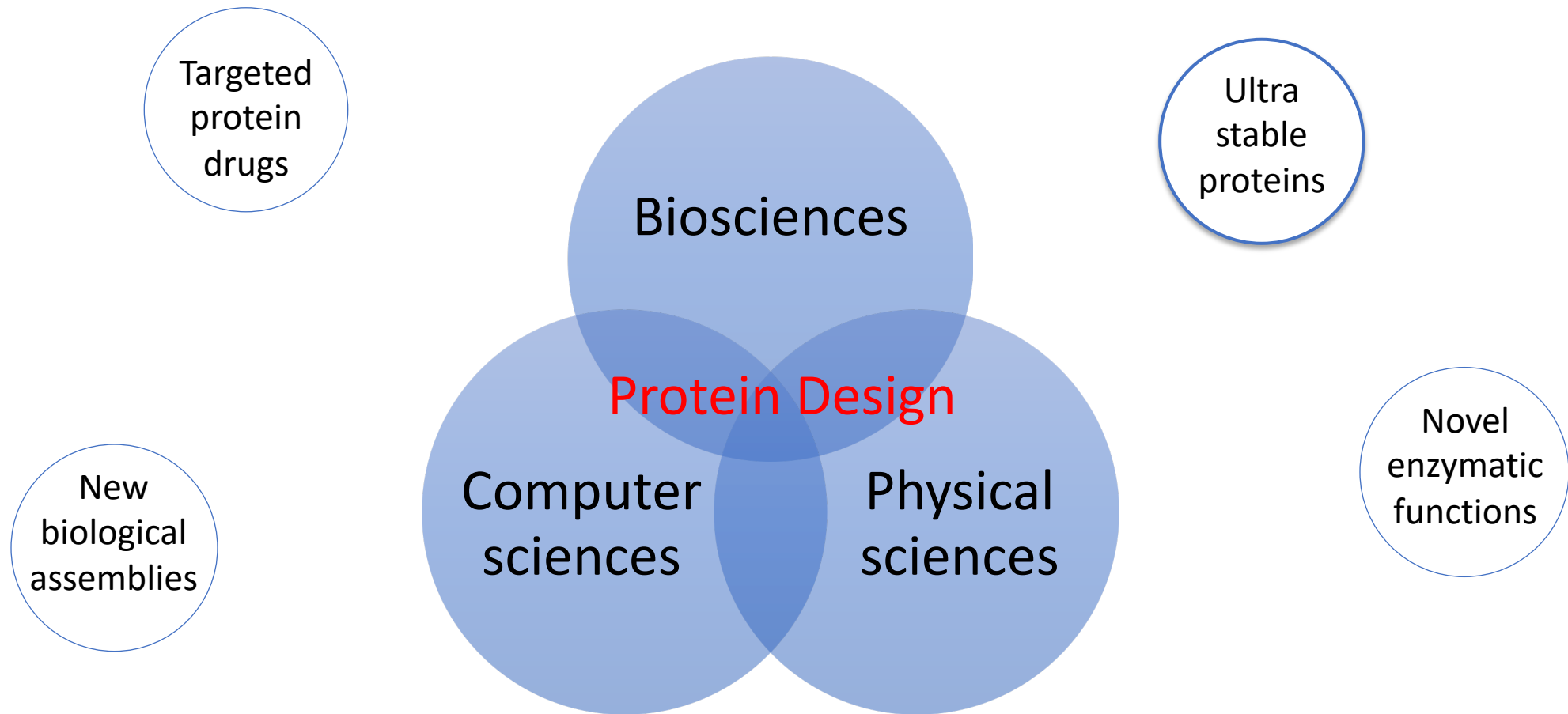
- Gene assembly
- Protein expression
- Biochemical, biophysical, functional and structural characterization (X-ray crystallography, Cryo-EM)

- 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

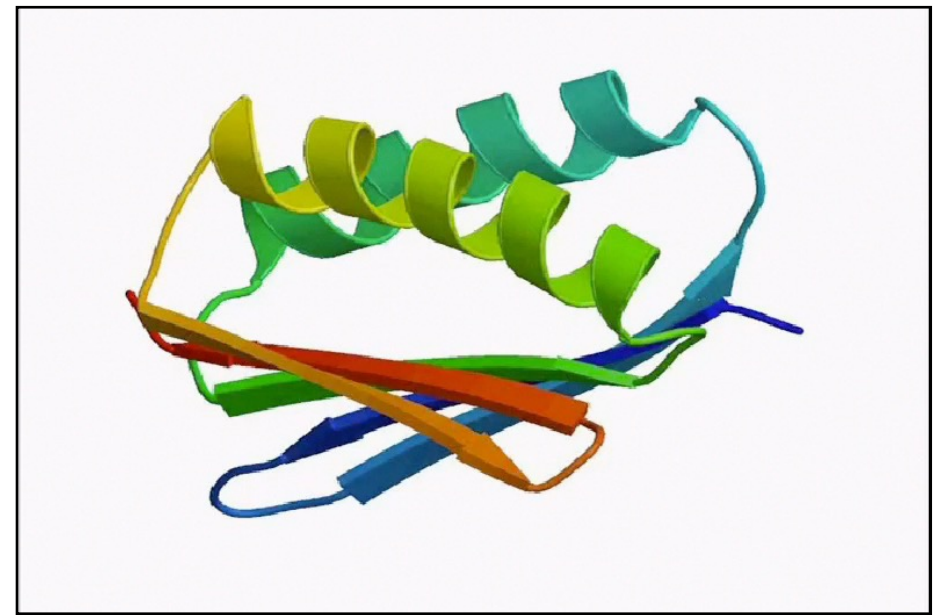
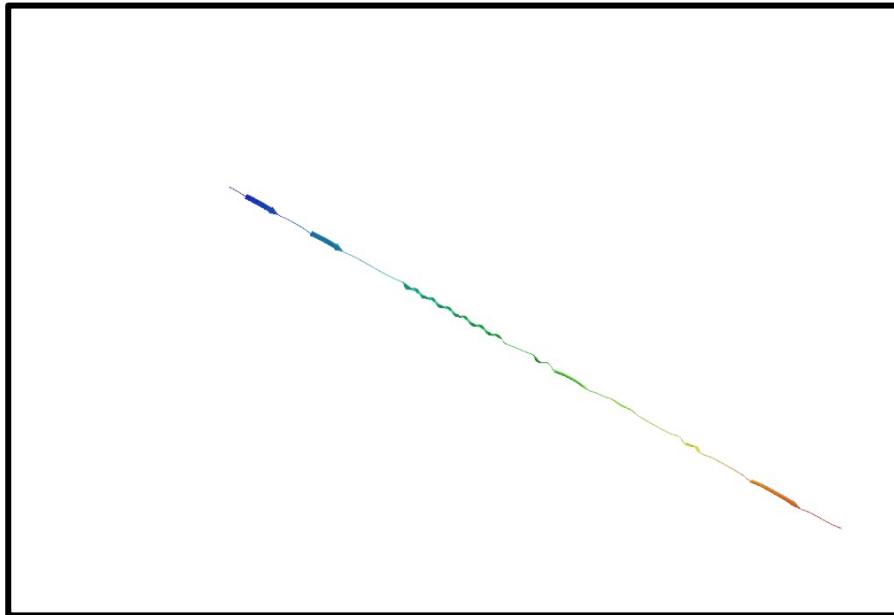


Searching for an energy minimum

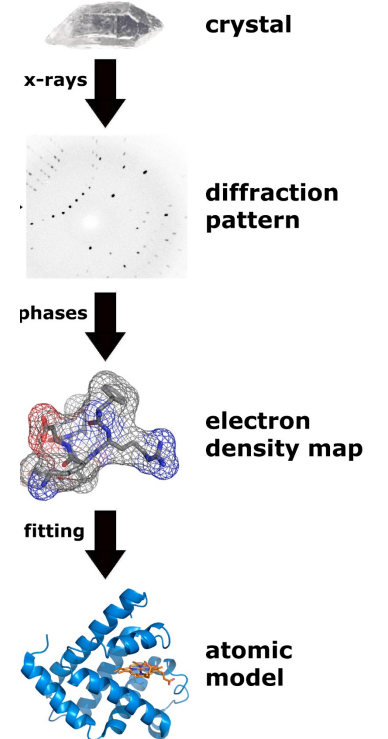
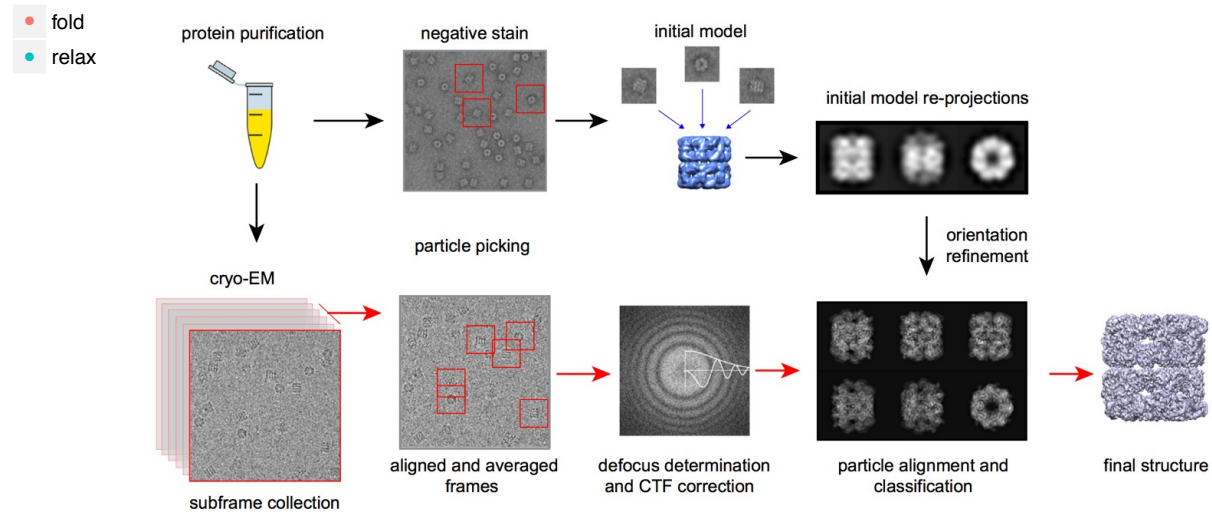
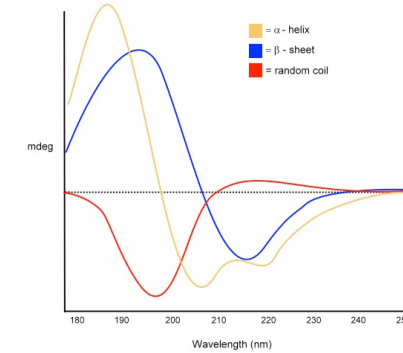
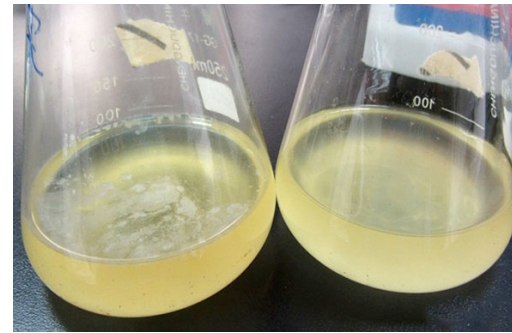
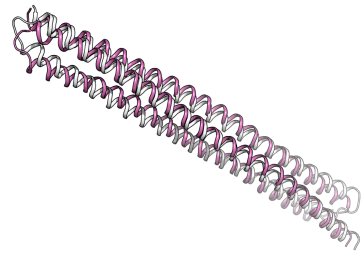
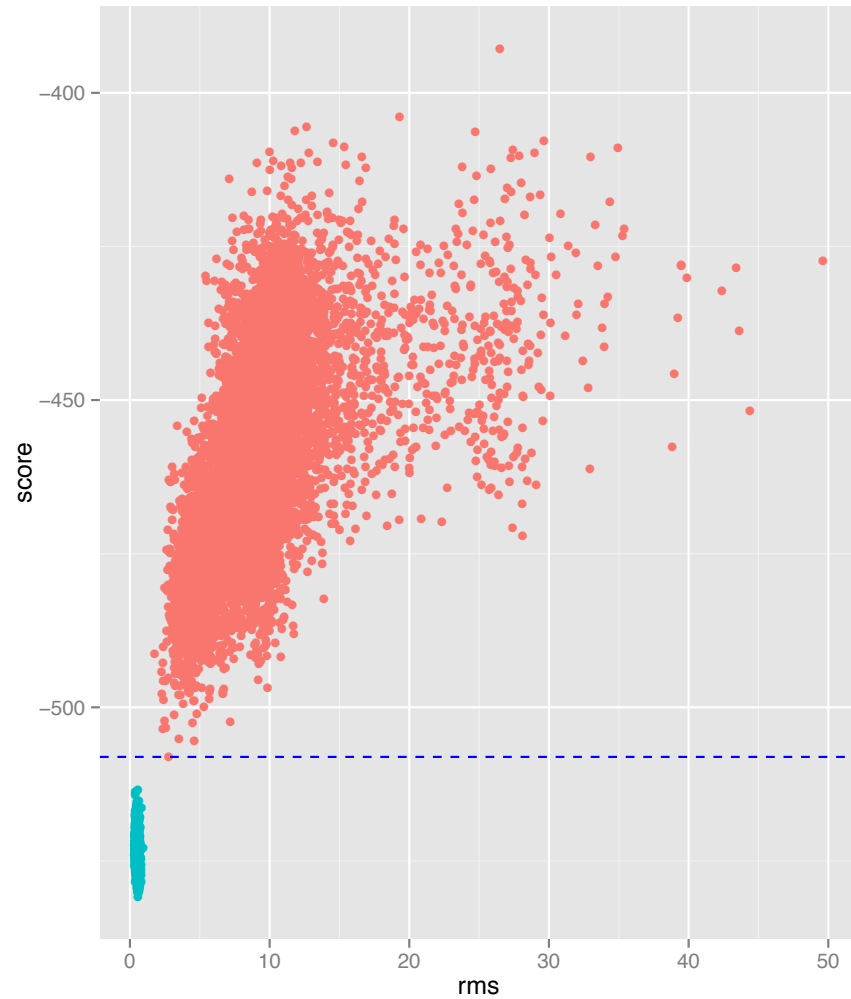
Rosetta

Protein structure prediction

Protein structure design



General protein design workflow



How we use managed data in the lab



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



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

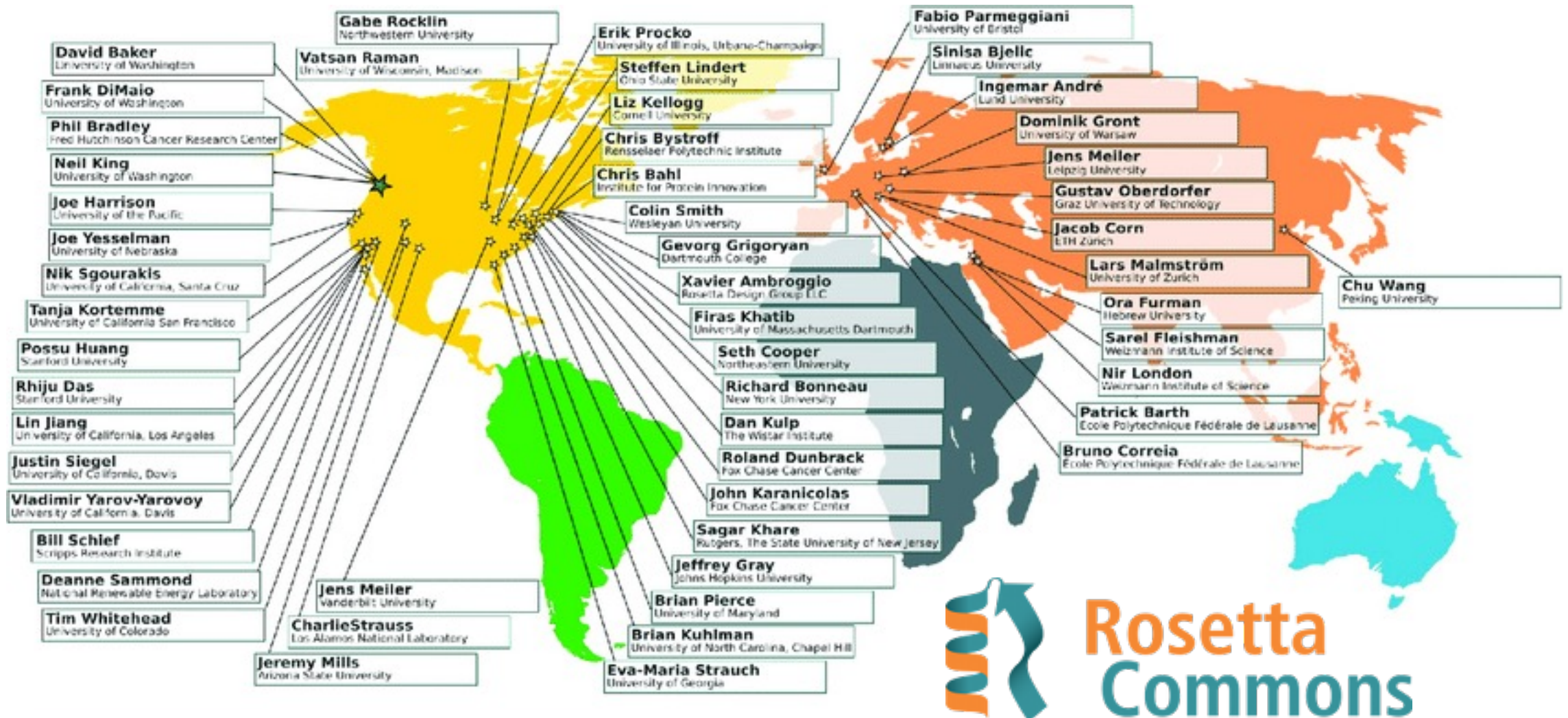


GitLab

ProThermDB



How we manage data within the community



How we manage data within the community


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

```
1 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

[master](#) [commits](#) [queue](#) [submit](#) tools ↗ login

 **master branch summary**

linux clang code quality

linux clang scientific doc

linux clang scientific doc

linux clang scientific.mp

linux clang scientific.mp

linux clang scientific.sb

linux clang scientific.sb

antibody_grafting ant

design_last debug ok

fragments_picking debug

loop_modeling_kic_12re

mhc_epitope_energy

mp_f19_decoy_discrim





mp_symdock.debug


rna_denovo_favorites

sb_franklin2019_mpdg

Re: [RosettaCommons/main] Updating Membrane Solvation Derivatives (#5252)

To: RosettaCommons/main, Cc: Push

May 27, 2021 at 8:05 PM [Details](#) 

[@woodsh17](#) pushed 7 commits.

- [5739223](#) Updating comments/description of unit test
- [194b595](#) Updating spline function for short distances for FaMPSolv
- [397becb](#) Updating to use sigma value for minimum distance cutoff, instead of constant value.
- [1768540](#) Updating option description for analytic_membetable_evaluation
- [a74367b](#) Adding extra comments for changes
- [0717684](#) Removing commented out lines
- [4df8585](#) pulling changes

You are receiving this because you are subscribed to this thread.

[View it on GitHub](#) or [unsubscribe](#).

How we manage data within the community

- Own metadata file format (silent files)

The screenshot shows a web browser window with the URL `new.rosettacommons.org/docs/latest/rosetta_basics/file_types/silent-file`. The page title is "Silent File". On the left is a sidebar with navigation links: "Getting Started", "Build Documentation", "Rosetta Tutorials", "Rosetta Basics" (with sub-links: "Running Rosetta", "Units in Rosetta", "How Rosetta works", "File Formats in Rosetta", "Non-protein Residues", "Preparing Structures", "Command line options"), "Rosetta Applications", "Rosetta Scripting Interfaces", and "Developer". The main content area has a search bar and "Home" and "Feedback" buttons. The text explains that a silent file is a compact format for multiple structures, generated by Rosetta simulations like Abinitio and Ligand Docking, and used to save storage space. It mentions that "Protein Silent Struct" is seen in Abinitio outputs. Under the "Header" section, a table shows the first two columns of a silent file: "SEQUENCE" (Structure sequences presented by one letter) and "SCORE" (Rosetta score terms). Under the "Body" section, a table lists the columns of a silent file: "Columns 1-4" (Residue sequence number), "Columns 5-7" (Secondary structure one letter code), "Columns 8-17" (Phi angle), "Columns 18-26" (Psi angle), "Columns 27-35" (Omega angle), "Columns 36-44" (CA atom coordinates x), "Columns 45-53" (CA atom coordinates y), "Columns 55-62" (CA atom coordinates z), and "Columns 64-98" (Chi angle real data if possible). The page concludes by stating that the Binary Silent Struct File is useful for compressing multiple PDBs to save computer space.

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 Abinitio 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

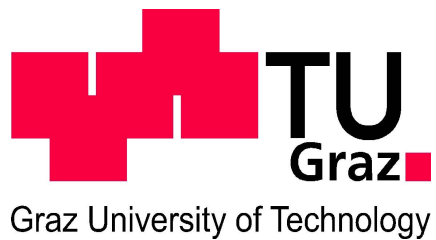
Columns 1-4	Residue sequence number
Columns 5-7	Secondary structure one letter code
Columns 8-17	Phi angle
Columns 18-26	Psi angle
Columns 27-35	Omega angle
Columns 36-44	CA atom coordinates x
Columns 45-53	CA atom coordinates y
Columns 55-62	CA atom coordinates z
Columns 64-98	Chi angle real data if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

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

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



European Research Council
Established by the European Commission

802217 HelixMold



P 30826-B21



GA 863170 ArtiBLED