Stony Brook University

Oganov's Lab

COMPUTATIONAL MATERIALS DISCOVERY LABORATORY

Crystal and protein structure modeling, software development and applications

Maksim Rakitin

Stony Brook University

BNL, 07/30/2015

Outline

- Crystal and protein structure modeling from first principles:
  - BeF_2
  - alanine polypeptide
- Software development and applications

Education and professional background

2002-2006 – BS (SUSU)

2006-2008 – MS (SUSU)

2008-2012 – PhD (SUSU)

2013-Present - Postdoc

Stony Brook University

Oganov's Lab
COMPUTATIONAL MATERIALS DISCOVERY LABORATORY

Applied math and physics: LMTO calculations (**LMTO-ASA**), Fe-H systems

Applied math and physics: DFT calculations (**WIEN2k**), Fe-H, Fe-Me-H

Condensed matter physics: DFT calculations (**WIEN2k**), Fe-Me-H

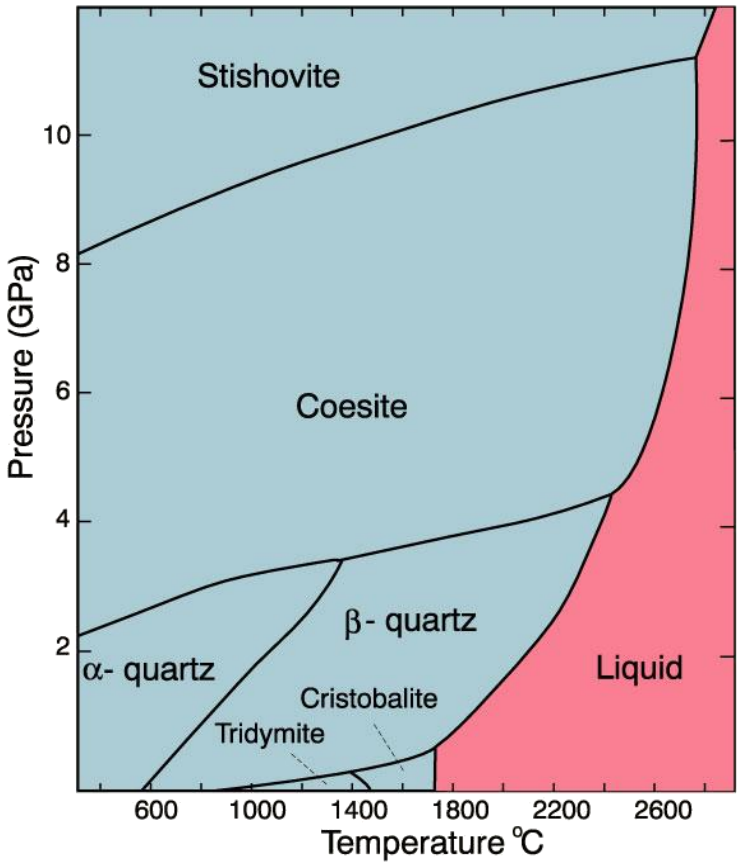
Crystal structure prediction (**USPEX**), DFT calculations of BeF_2 and SiO_2 phases under pressure (**VASP**, **QE**, **Phonopy**), proteins secondary structure prediction

Novel phase of BeF_2

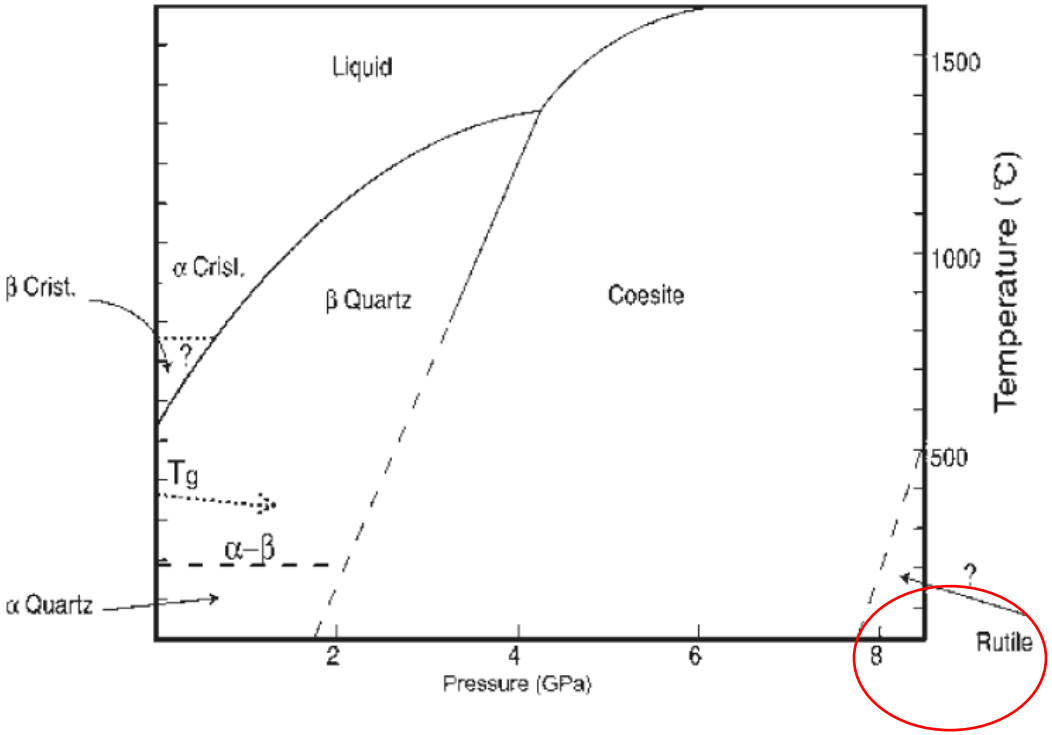
Novel phase of BeF_2

p-T phase diagrams for SiO_2 and BeF_2:

SiO_2:



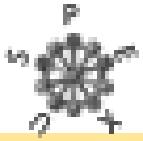
BeF_2:



Used in:

- biochemistry (protein crystallography)
- liquid-fluoride nuclear reactors

Calculation method



Prediction of a novel phase: **USPEX** and **VASP** DFT package

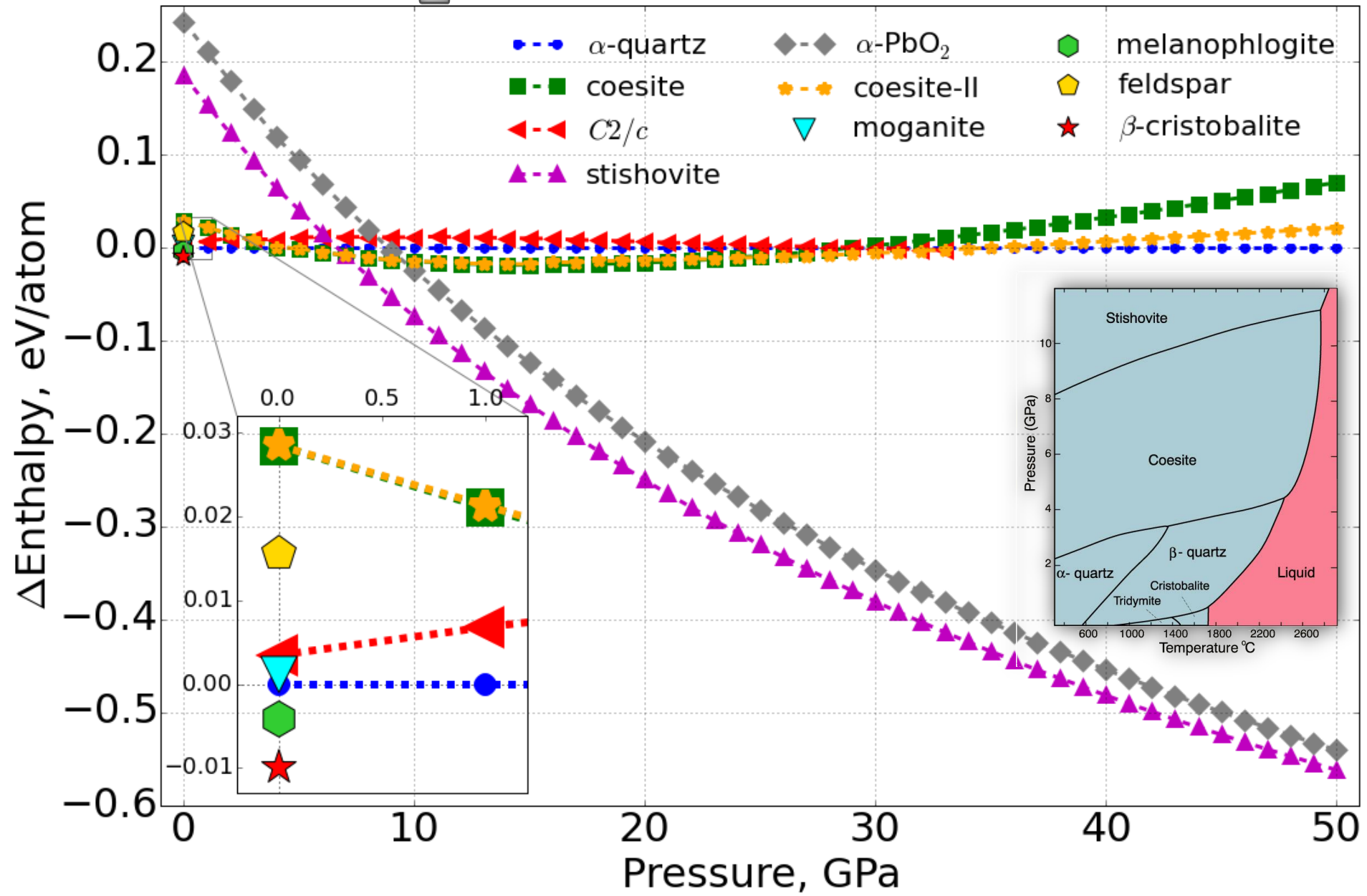
Simulation of all phases from 0 to 50 GPa: **VASP**

Calculation of phonons and other properties: **VASP**, **Quantum Espresso**, **Phonopy**

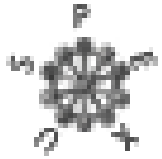
Automatic batch submission of calculations on a supercomputer

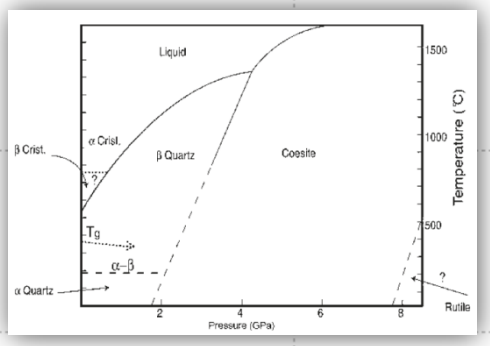
Bulk data analysis and plotting in **Python**

SiO_2 phase diagram

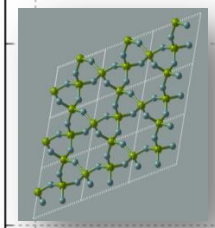
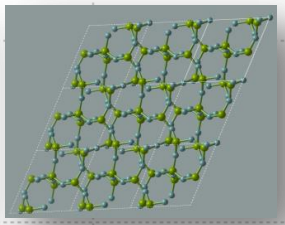
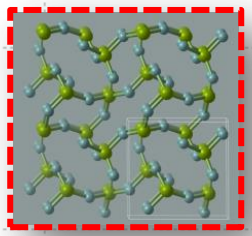
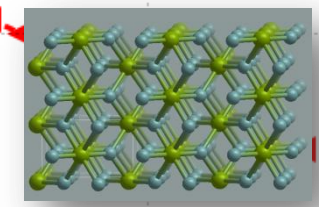


BeF_2 phase diagram





α-quartz
$C2/c$
α-PbO_2
coesite
stishovite
coesite-II

ΔEnthalpy, eV/atom

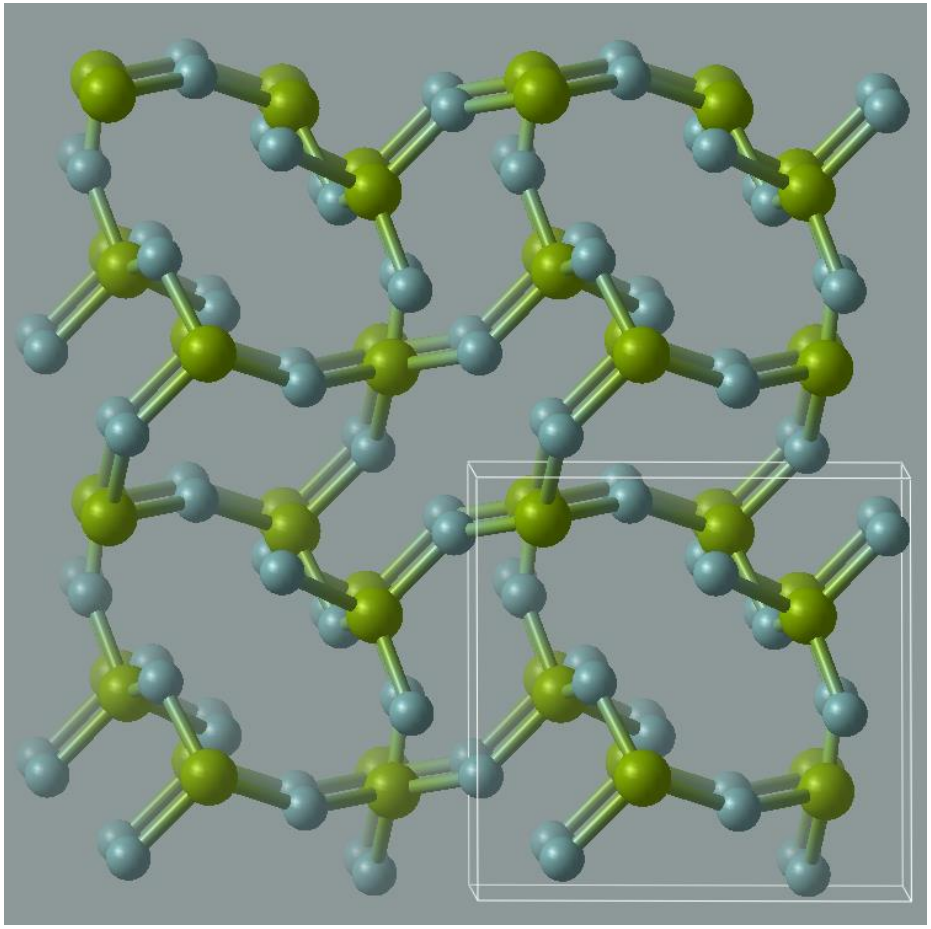
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06

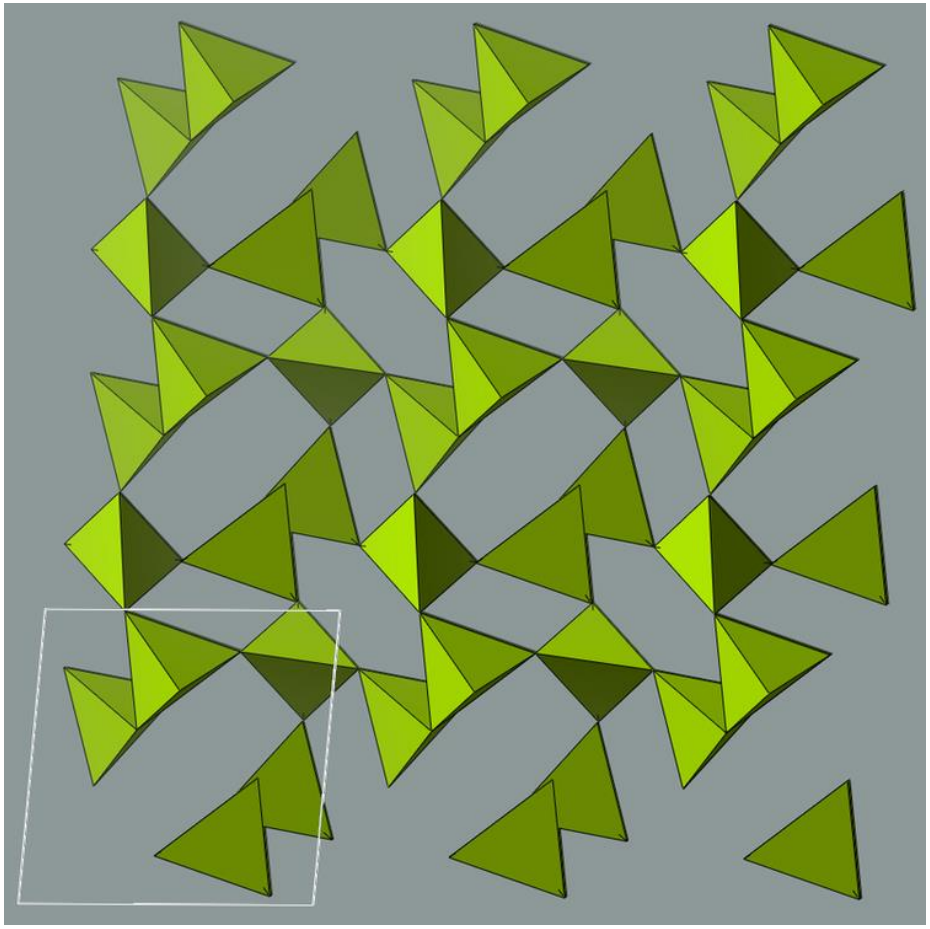
0
10
20
30
40
50

Pressure, GPa

New structure of BeF_2

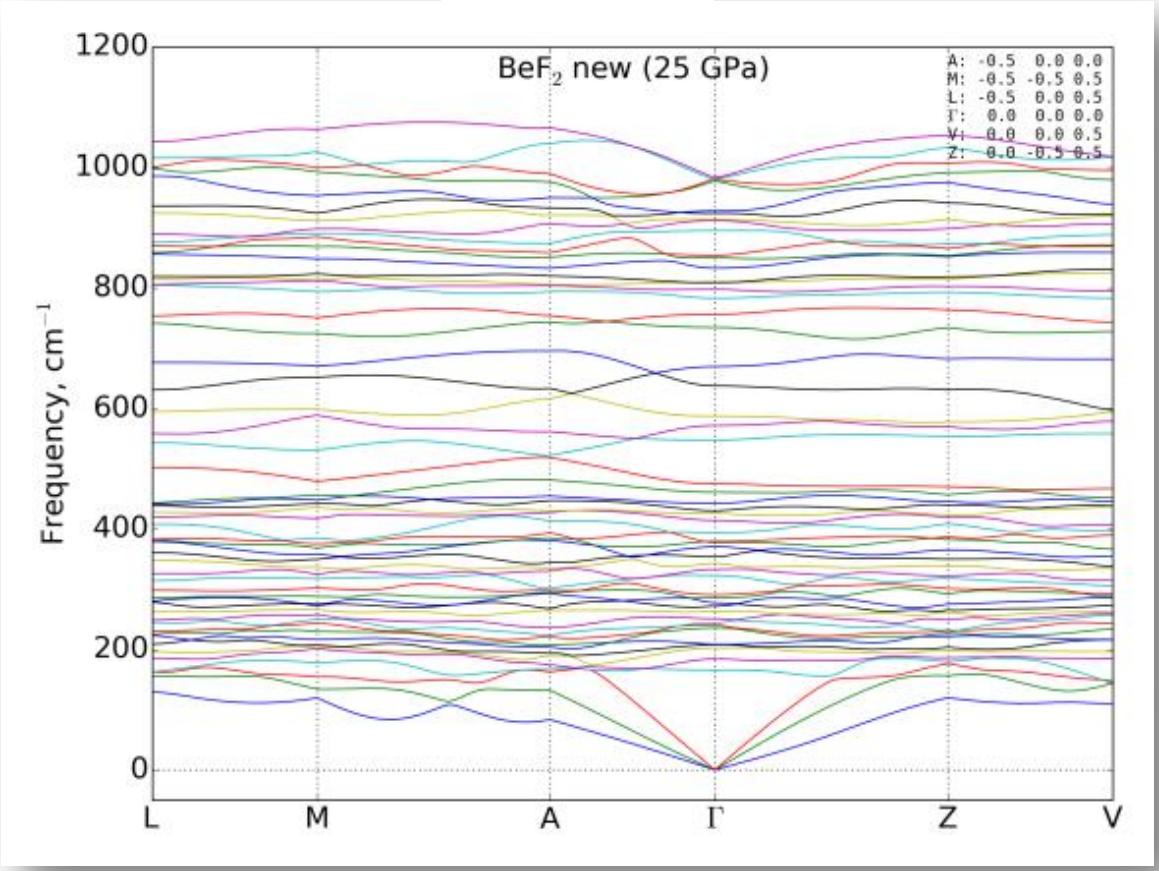
Using USPEX a new structure has been found at **18-27 GPa**:



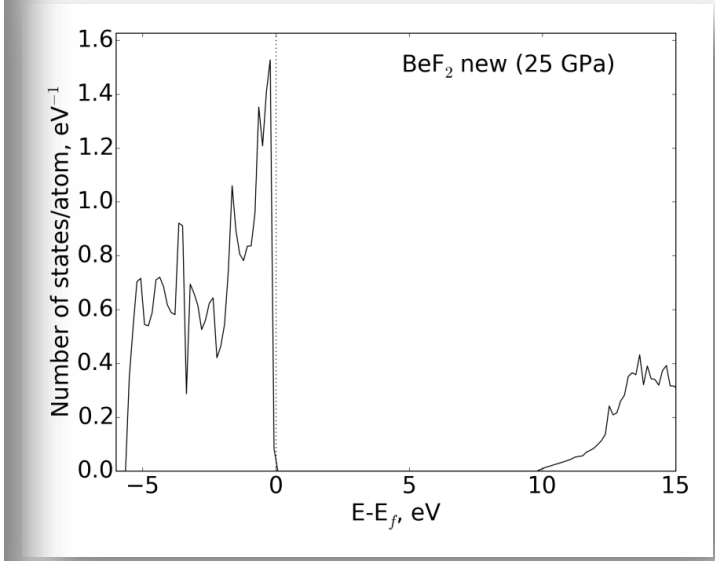


New C2/c structure properties

Phonons



DOS



M.S. Rakitin, A.R. Oganov, *et al. Novel phase of beryllium fluoride at high pressure* (2015), Submitted to PCCP, http://arxiv.org/abs/1506.03861

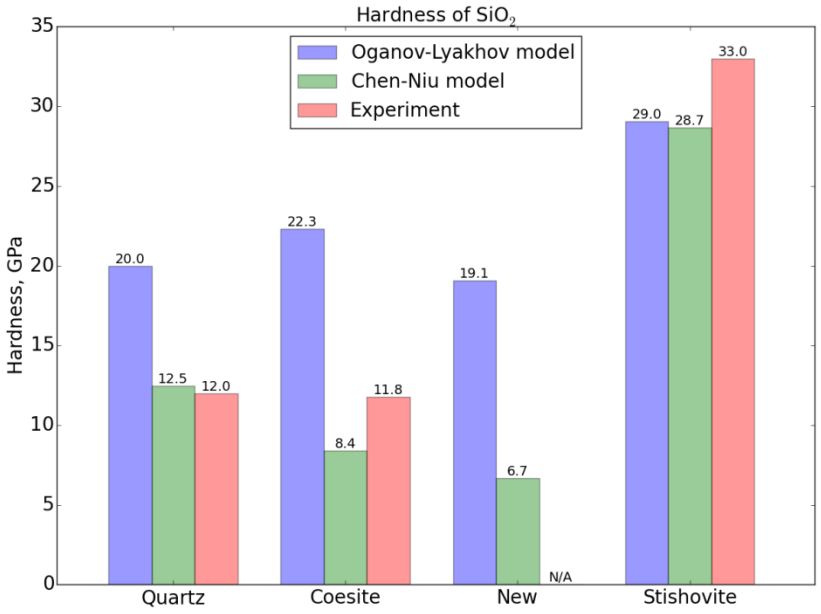
New $C2/c$ structure properties

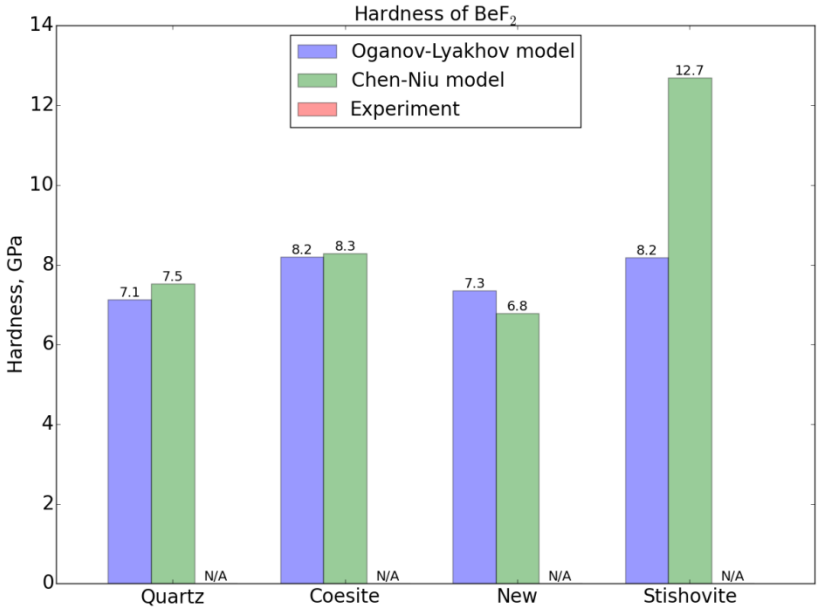
Density

System	Number of atoms	Volume, Å/cell	Density, g/cm^3
BeF_2 at 0 GPa:			
α-quartz	9	105.167	2.244
coesite	24	254.636	2.472
coesite-II	96	1021.960	2.464
$C2/c$	18	213.696	2.209
stishovite	6	47.771	3.294
BeF_2 at 20 GPa:			
α-quartz	9	73.078	3.230
coesite	24	202.001	3.116
$C2/c$	18	145.159	3.252
stishovite	6	41.492	3.793
SiO_2 at 0 GPa:			
α-quartz	9	116.934	2.580
coesite	24	283.341	2.839
coesite-II	96	1137.296	2.830
$C2/c$	18	243.569	2.477
stishovite	6	48.185	4.174
α-PbO_2-type	12	94.623	4.251

Hardness, GPa

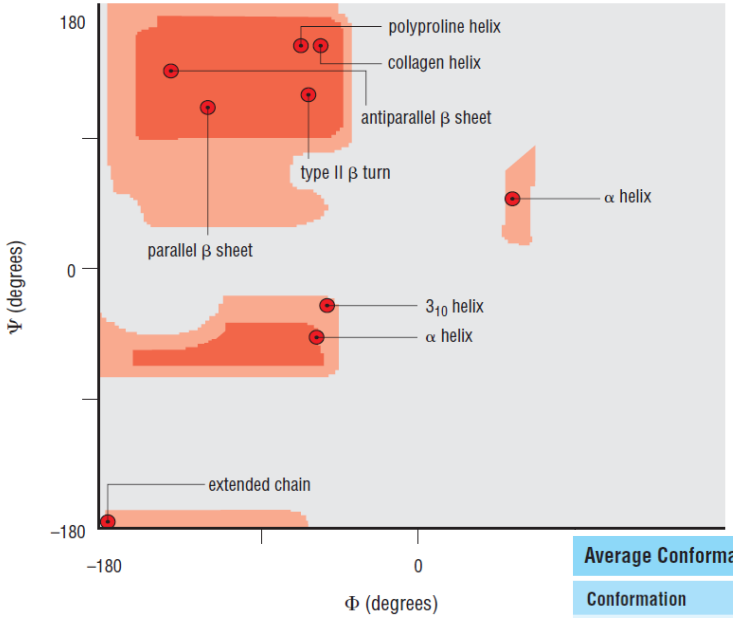
	Lyakhov-Oganov	BeF_2 Chen-Niu	Mukhanov *et al.*
Quartz	7.1	7.5	11.0
Coesite	8.2	8.3	11.7
New structure	7.3	6.8	13.5
Stishovite	8.2	12.7	15.1

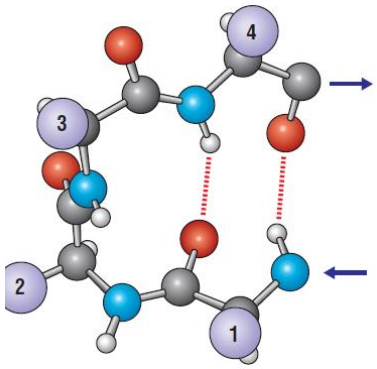


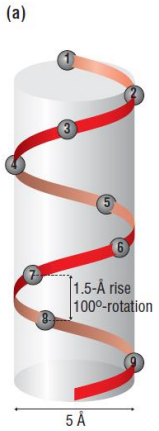


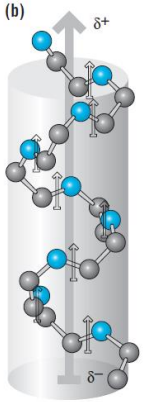
Proteins secondary structure predictions

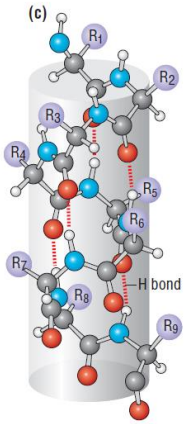
Proteins secondary structures





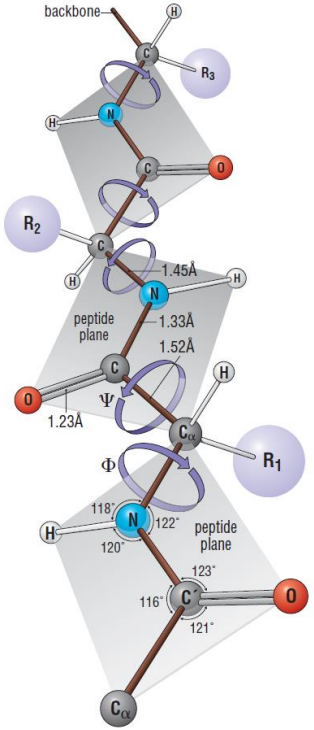


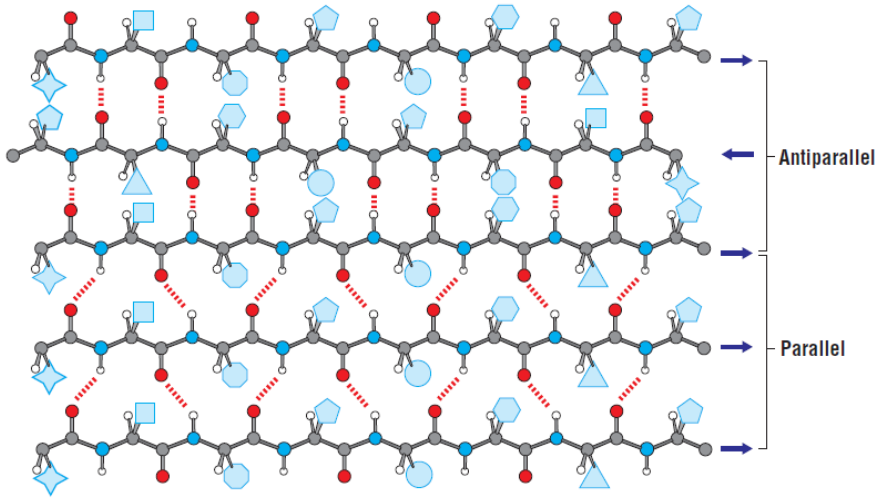


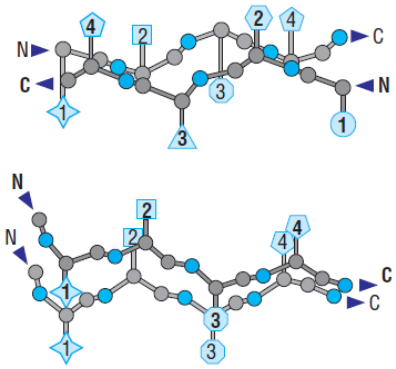


Average Conformational Parameters of Helical Elements

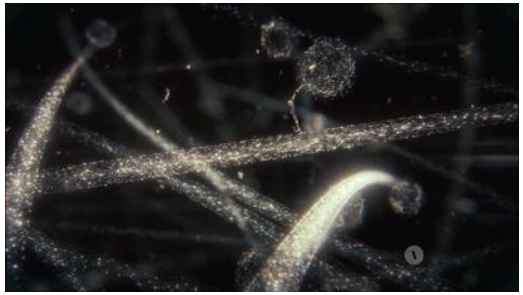
Conformation	Phi	Psi	Omega	Residues per turn	Translation per residue
Alpha helix	–57	–47	180	3.6	1.5
3-10 helix	–49	–26	180	3.0	2.0
Pi-helix	57	–70	180	4.4	1.15
Polyproline I	–83	+158	0	3.33	1.9
Polyproline II	–78	+149	180	3.0	3.12
Polyproline III	–80	+150	180	3.0	3.1

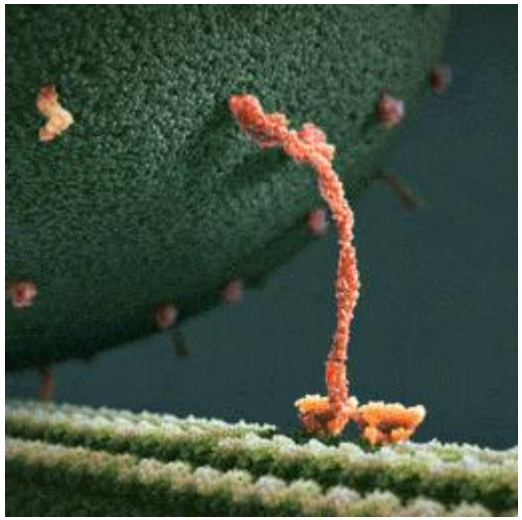


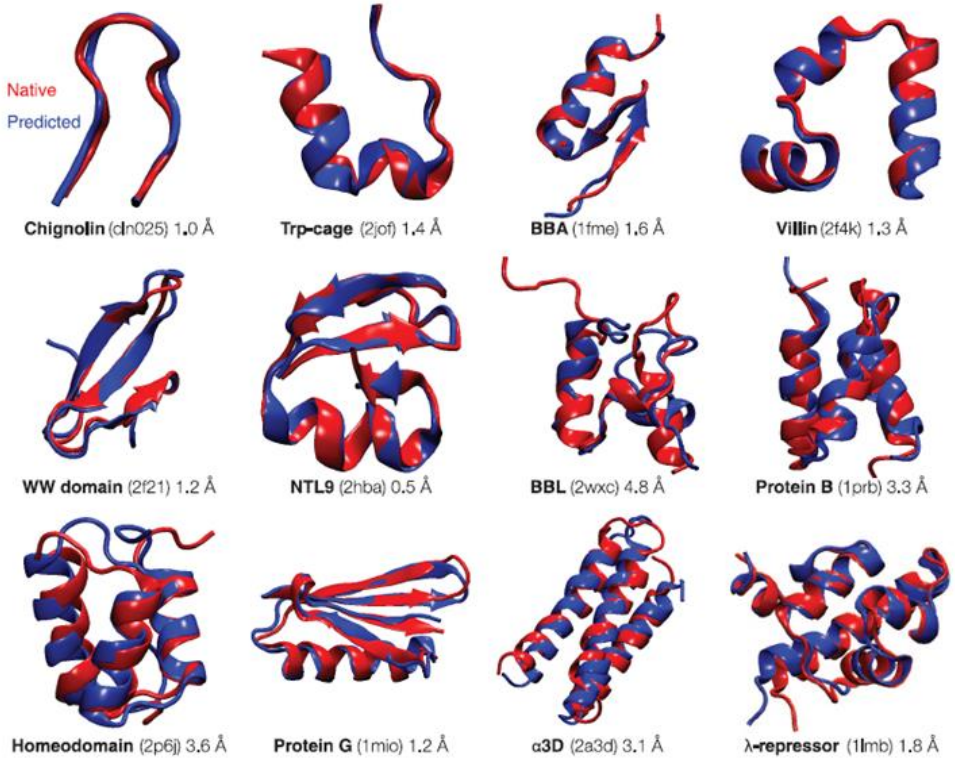




Proteins secondary structure predictions



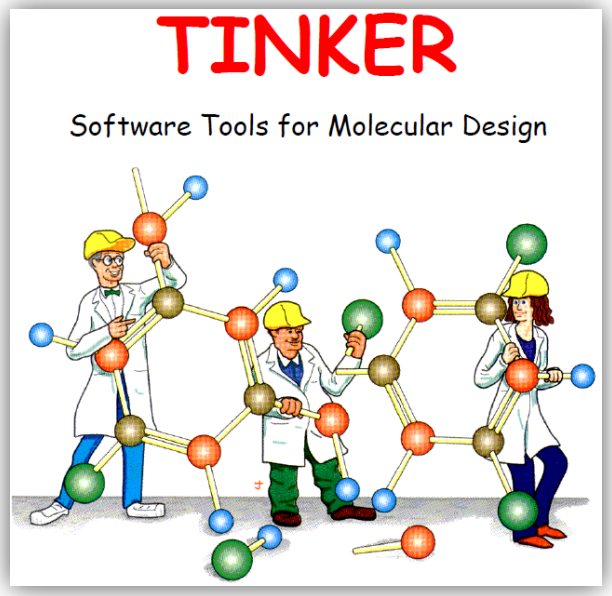




The Protein-Folding Problem, 50 Years On Ken A. Dill and Justin L. MacCallum
Science 23 November 2012: 338 (6110), 1042-1046.

- USPEX interface for Tinker molecular modelling program for prediction of secondary structures of proteins
- Critical Assessment of protein Structure Prediction (CASP) competitions

USPEX-Tinker interface



INPUT.txt:

```
USPEX : calculationMethod
-400  : calculationType
1     : optType

0 : doSpaceGroup

50    : populationSize
50    : initialPopSize
60    : numGenerations
60    : stopCrit

1     : AutoFrac

abinitioCode
12
ENDabinit
```

Specific/ folder:

input.key_1:

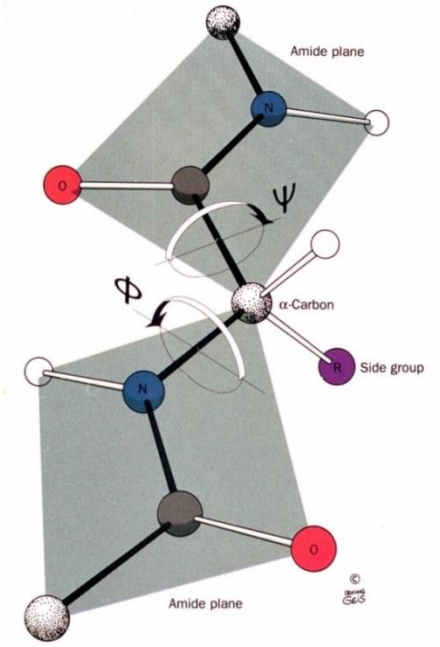
```
Parameters charmm22.prm
enforce-chirality
SOLVATE HCT
```

input.make0_1:

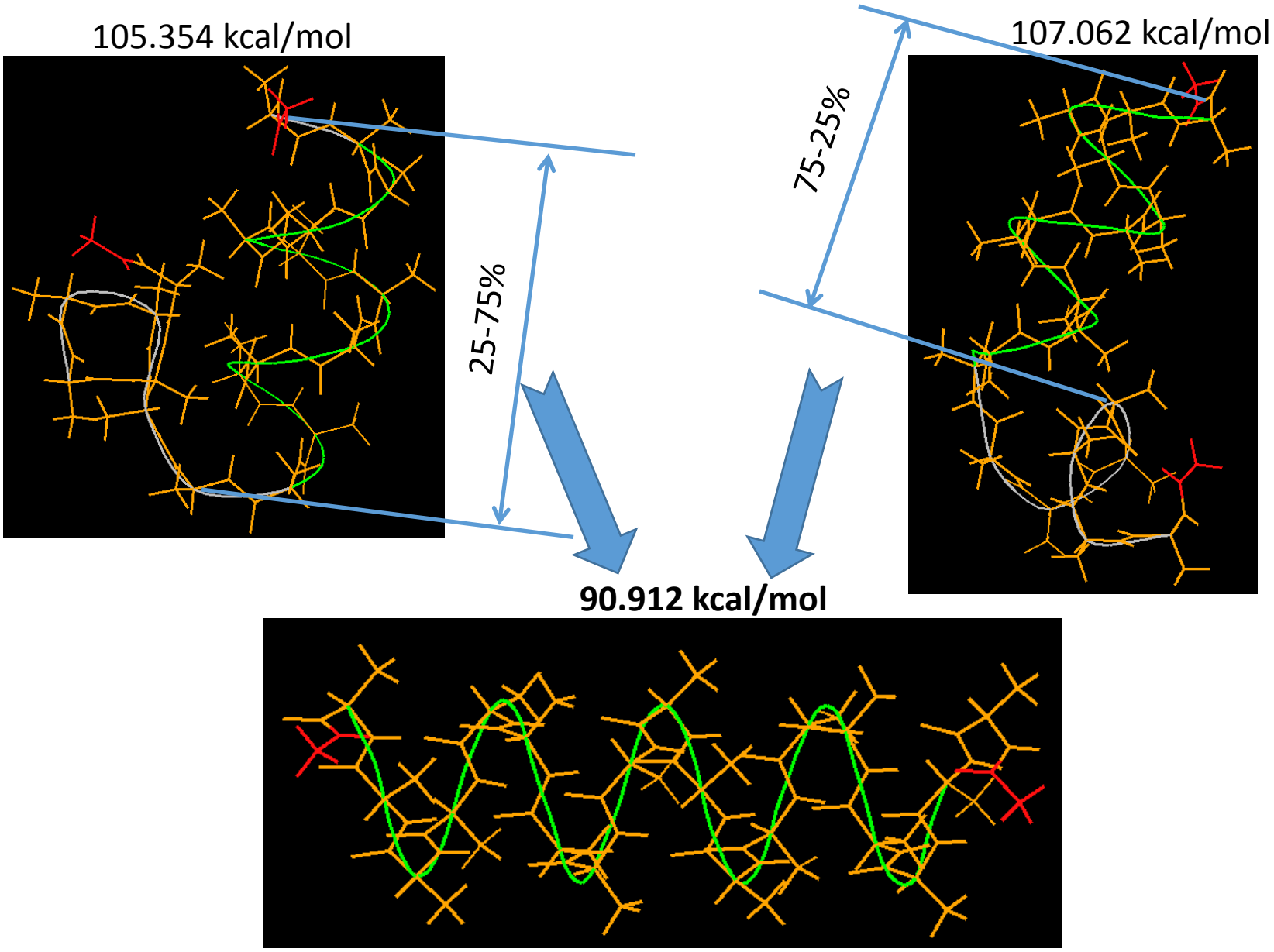
```
input
Alanine 20-peptide
ace
ala  0.001  -0.001  180
ala  0.002  -0.002  180
ala  0.003  -0.003  180
ala  0.004  -0.004  180
ala  0.005  -0.005  180
ala  0.006  -0.006  180
ala  0.007  -0.007  180
ala  0.008  -0.008  180
ala  0.009  -0.009  180
ala  0.010  -0.010  180
ala  0.011  -0.011  180
ala  0.012  -0.012  180
ala  0.013  -0.013  180
ala  0.014  -0.014  180
ala  0.015  -0.015  180
ala  0.016  -0.016  180
ala  0.017  -0.017  180
ala  0.018  -0.018  180
ala  0.019  -0.019  180
nme

N
```

Variation operators:

- Random
- Heredity
- Swap
- Secondary Switch
- Shift Borders



Example of Heredity operator



Results

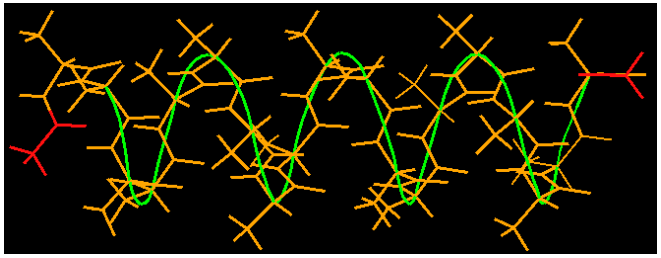
Note: other methods[1,2] operate with **alanine dipeptide** for performance tests!

ALA20:

Found structures: 19–20 Total calculations: 20

Success rate: 95–100.00%

Average structures: 1167–1391

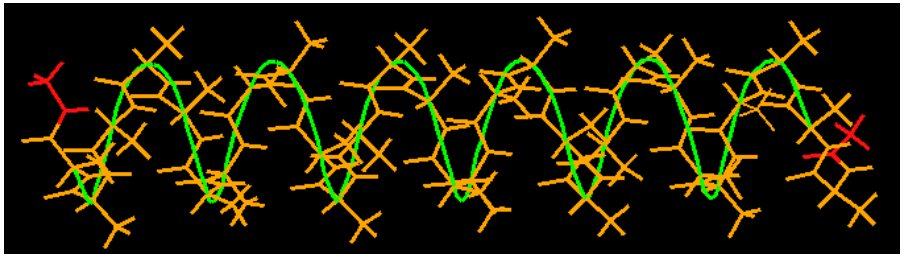


ALA30:

Found structures: 17 Total calculations: 20

Success rate: 85.00%

Average structures: 1783



[1] Well-Tempered Metadynamics (DOI: 10.1103/PhysRevLett.100.020603)

[2] Robotics-inspired methods (DOI: 10.1002/jcc.21931)

Conformational Preferences of the Amino Acids

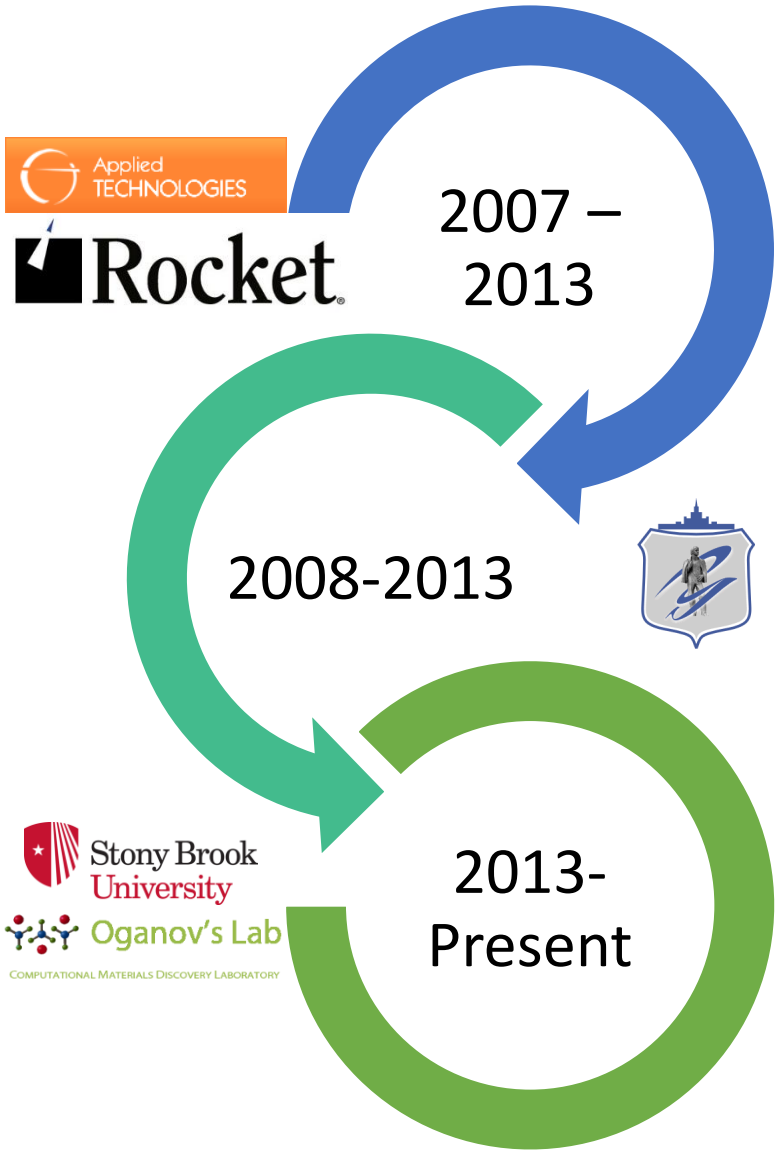
Amino acid	Preference α-helix	β-strand	Reverse turn
Glu	**1.59**	0.52	1.01
Ala	**1.41**	0.72	0.82
Leu	**1.34**	1.22	0.57
Met	**1.30**	1.14	0.52
Gln	**1.27**	0.98	0.84
Lys	**1.23**	0.69	1.07
Arg	**1.21**	0.84	0.90
His	**1.05**	0.80	0.81
Val	0.90	**1.87**	0.41
Ile	1.09	**1.67**	0.47
Tyr	0.74	**1.45**	0.76
Cys	0.66	**1.40**	0.54
Trp	1.02	**1.35**	0.65
Phe	1.16	**1.33**	0.59
Thr	0.76	**1.17**	0.90
Gly	0.43	0.58	**1.77**
Asn	0.76	0.48	**1.34**
Pro	0.34	0.31	**1.32**
Ser	0.57	0.96	**1.22**
Asp	0.99	0.39	**1.24**

Corresponds | Doesn't correspond

Is being calculated | No secondary structure

Software development

Software development

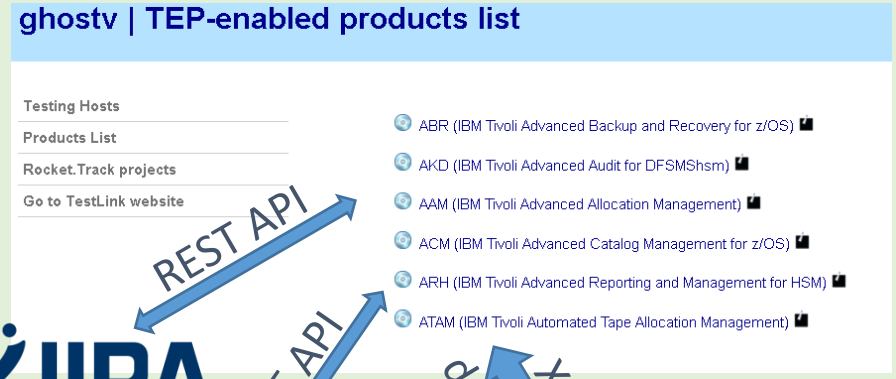


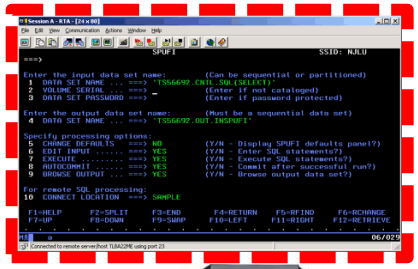
- QA team lead in an IBM-related software project, testing automation (**TEP Automated Testing System/TATS**):
  - Python, SOAP, RITA for the backend;
  - PHP, HTML, JavaScript, jQuery, XML, JSON, REST API for the frontend.
- JIRA, CVS, SVN, Bazaar, Linux/Unix/Windows/z/OS;
- Other projects: PostgreSQL, MySQL, SQLite, Perl, AutoIt3, REXX, bash, …

- Python/shell scripts development for monitoring of WIEN2k jobs with email notification in the PBS and SLURM queue systems;
- C++, Python, shell scripts for data analysis;
- LaTeX + BibTeX for PhD thesis.

- **USPEX** code development (Python, Matlab, Octave);
- Online utilities (PHP, HTML, JavaScript, jQuery, WebGL, Python);
- USPEX continuous integration scheme: nightly builds + automated testing system (Python);
- LaTeX + BibTeX + plasTeX for USPEX manual (PDF, HTML, some parts are generated by Python);
- Trac bug tracking system, SVN.

TATS overview





REST API

REST API

XML data files

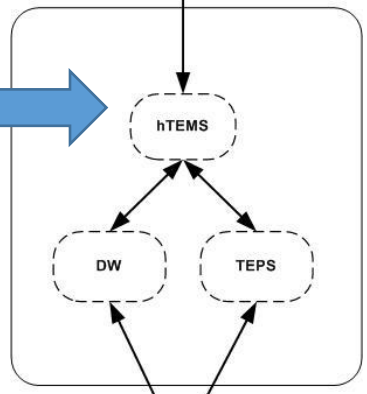
Special FTP interface, RITA

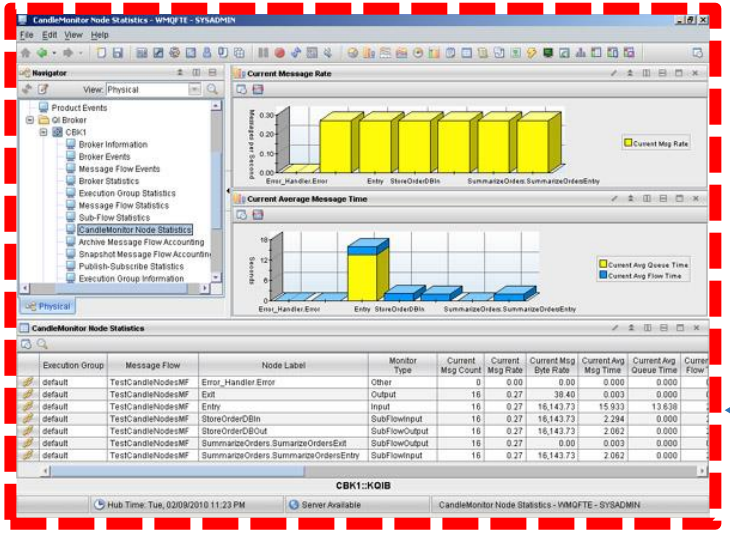
SOAP

JIRA

Test Link

python





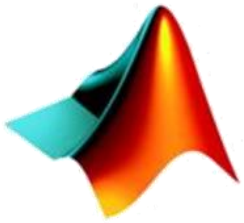
IBM/Rocket Software product

My development

Development in Oganov's Lab

- **USPEX evolutionary algorithm:** http://uspex.stonybrook.edu/uspex.html:
  - Matlab code support: releases, new features, bug fixes (team work)
  - Porting Matlab code (60,000+ rows) to Python (sole developer)
- **USPEX continuous integration scheme** (Python, nightly builds + auto-testing)
- **Online utilities** http://han.ess.sunysb.edu/ (currently about 20 utilities):
  - http://han.ess.sunysb.edu/crystal_form/ (Python, WebGL)
  - http://han.ess.sunysb.edu/stokes/ (simplified REST API, web interface)
  - http://han.ess.sunysb.edu/volume_estimation/ (Python, API)
  - . . .
- **USPEX manual:** http://han.ess.sunysb.edu/uspex_manual/

Matlab to Python porting



MATLAB

python™ powered

- Good in math, but…
- Current USPEX implementation:
  - Poor modularity
  - Strongly correlated code (global vars,…)
  - Hard to scale
  - Hard to add new functionality
  - Works only under Linux/Unix (since unix-related commands are used internally)
- General Matlab disadvantages:
  - Single function files
  - Poor in work with strings
  - Doesn't support command line options
  - Hard and expensive to compile
  - Expensive and not available to all users
- Python implementation of USPEX:
  - Better modularity and organization of the code, in-module testing
  - Independent modules can be imported as well as used as scripts
  - General classes for inputs, investigated system properties, etc.
  - API support
  - Will work under all platforms (Windows/Linux/Unix/Mac)
- General Python advantages:
  - Lots of available libraries
  - GUI with native TkInter or PyQT
  - Rich support of command line options
  - Possible to compile (.pyc, .pyd)
  - Free

Simplified REST API

Representational State Transfer Application Programming Interface

Problem:

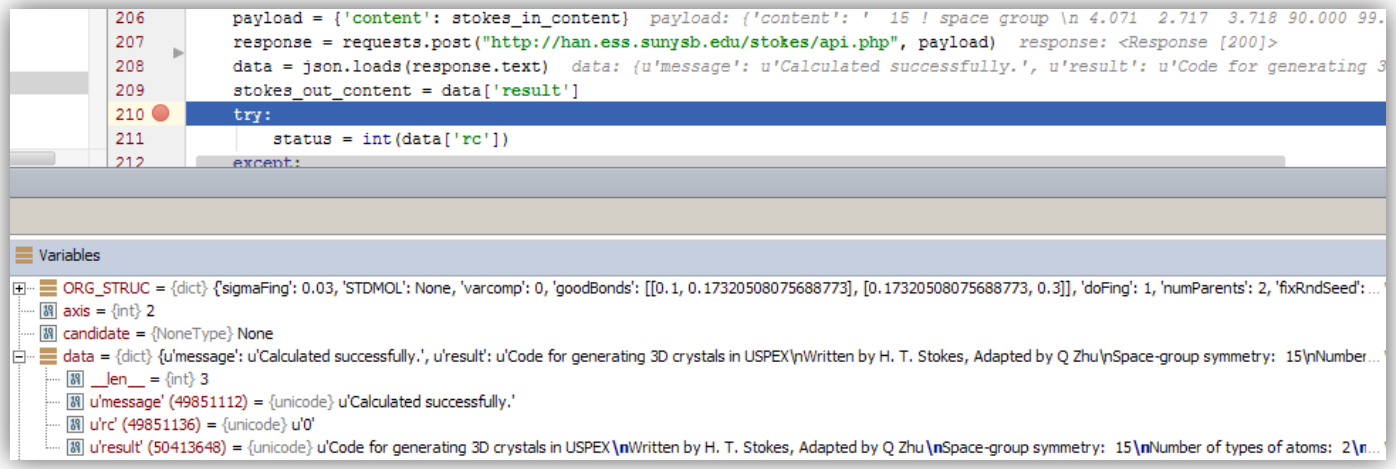
Fortran program by Harold T. Stokes to randomly generate structures for USPEX with the specified space group cannot be compiled on a Windows machine with MinGW/gfortran.

Possible solutions:

- Get **ifort** compiler for Windows and compile the source code – expensive.
- Rewrite the code in Python – it took H.T. Stokes 20+ years to write the program – challenging and time-consuming task and high risk to introduce bugs.
- **Create web-based API to the compiled program on Linux server – fast and free!**

Implementation:

- http://han.ess.sunysb.edu/stokes/api.php – PHP code which processes HTTP POST requests and returns response in JSON format;
- http://han.ess.sunysb.edu/stokes/ – web-page with JavaScript which uses the API;
- Python module uses the API to get data in the code using requests library:



Notes: The API is not RESTful, i.e. no GET/PUT/DELETE requests are processed, no authentication, no https, but it's enough for this problem.

Interactive visualization

Problem:

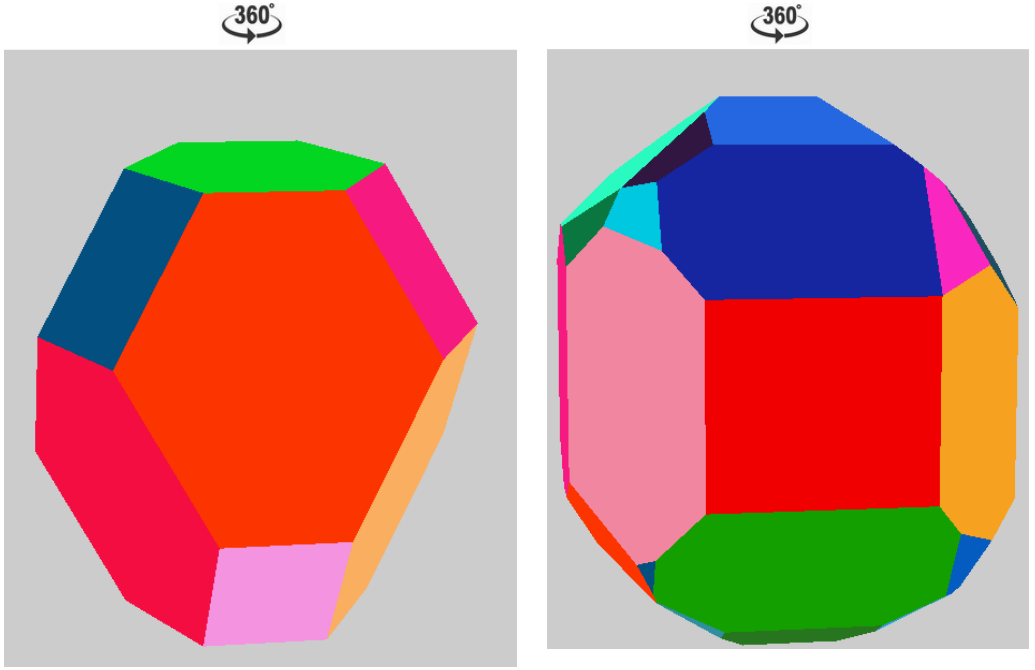
Visualize 3D matplotlib plot of the Python program for prediction of crystal forms on a web-page, so that the resulted image could be rotated, zoomed, etc.

Possible solutions:

- D3 viewer for matplotlib – the idea is good, but only 2D interactive plots are supported;
- https://plot.ly/ – hard to plot surfaces;
- http://www.pythonocc.org/ – complicated input;
- Polyhedron 1, Polyhedron 2 – pure **WebGL**, large input;
- More options are listed here;
- **Use three.js library to plot the polygon using a minimal set of input data!**

Implementation:

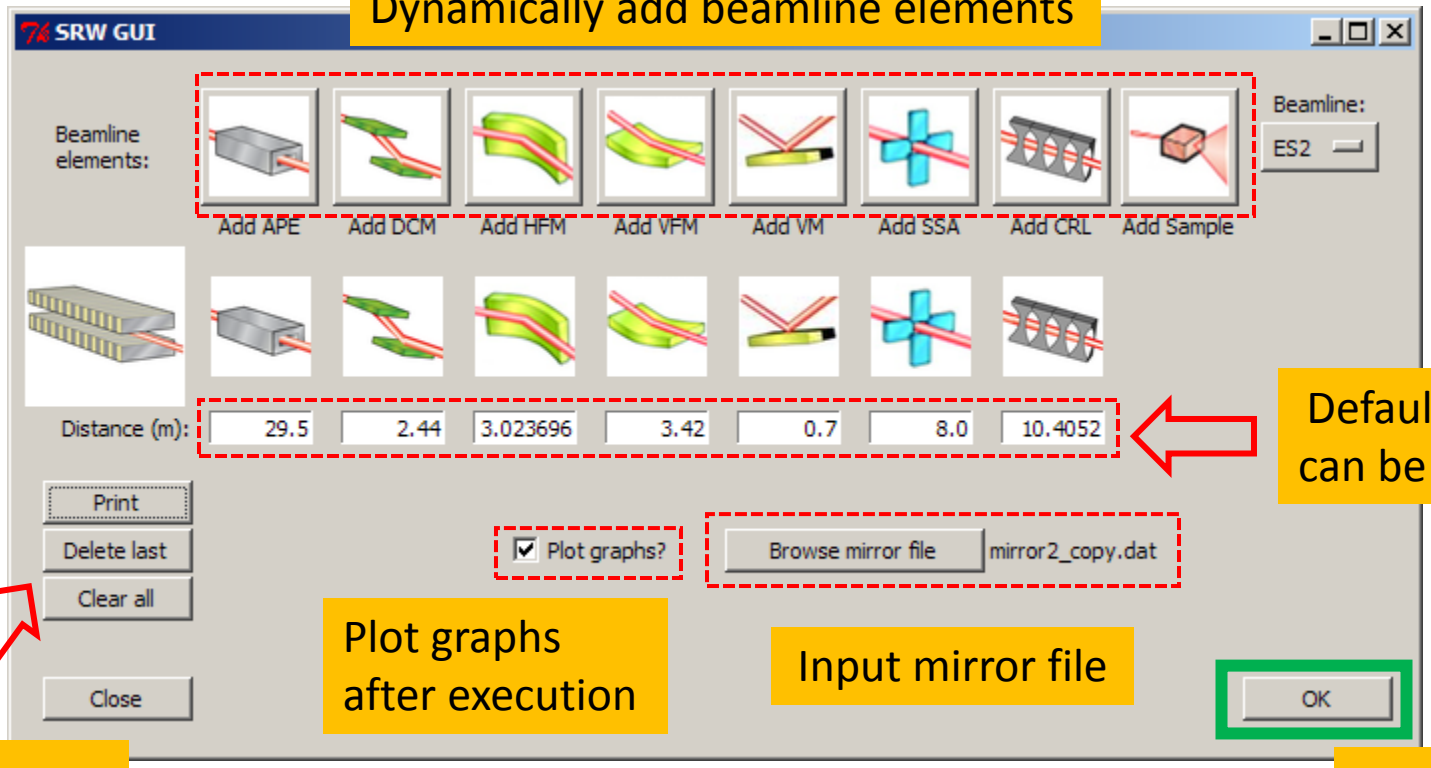
- Python module performs calculation of the surface energies and creates a set of vertices, indices, colors.
- Another Python module creates an html file from templates with data understandable by the *three.js* library.
- http://han.ess.sunysb.edu/crystal_form/ – web interface for the Python program: crystal form from POSCAR or CIF files!



Programming exercise

Programming exercise

Dynamically add beamline elements



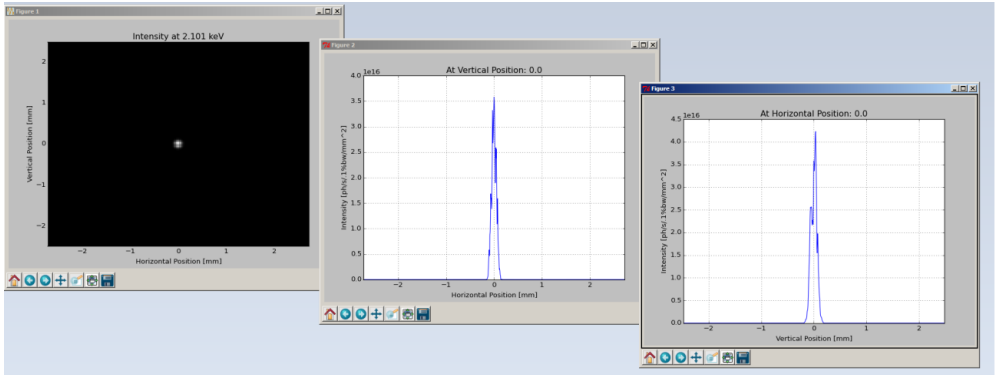
Default values, can be changed

Remove elements one by one or all at once

Plot graphs after execution

Input mirror file

Execute SRW!

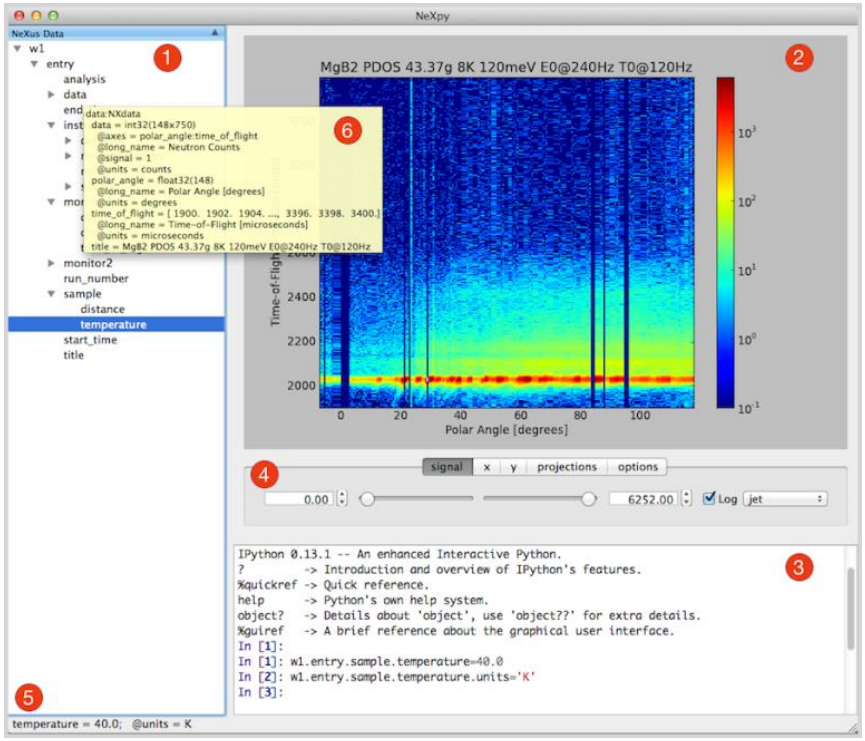


SRW example is adapted for SMI beamline (private communication with M. Zhernenkov)

Demonstration

Different implementations

- Implementation of GUI based on more advanced library PyQT instead of TkInter
- Integration of console with GUI, like in NexPy: IPython + PyQT + Matplotlib:



GUI interface together with command line environment

- Implementation of NumPy arrays to store data
- Rewriting C++ and IgorPro code to Python to avoid heterogeneous code

Conclusion

- Two examples of scientific modelling of materials and their properties based on the first principles/DFT calculations:
  - We discovered novel phase of BeF_2 under high pressure;
  - We successfully implemented first principles modeling of proteins secondary structure using an example of alanine polypeptide.

- Implementation of Python programming of USPEX code for structure prediction.

- Examples of modern programming technics (e.g., TkInter/PyQT, API, data mining, supercomputing, etc.) which can also be implemented at NSLS-II.


Dank u wel
Mahalo
Khawp khun
Juspaxar
Spasibo
Efcharisto
Salamat
Thank
Toda raba
Cám ơn
Dhanyawaad
Köszi
Pidamayado
Tusen takk
Dankon
Terima kasih
you
Merci
Doh je
Shukria
Grazie mille
Tashakkur
Shukuria
bolzïn
suksama
Graciasa
Biyan
Kamsahamnida
Maake
Komapsumnida
Tingki
gozaimashita
atu
Tesekkurler
Ekhmet
Mehrbani
Yaqhanyelay
Efcharisto
Arigato
Obrigado
Dankscheen