JK

WORKSHOP

Part 1

History of JK framework

2017 (Helsinki) – I accidentaly blocked a system command by writing a script with the same name and giving it higher search priority

SOLUTION: call all my commands with unique first letter(s), e.g. JKcommand

```
$user: JK + <TAB>
JKCS0_copy          JKQC              JKfitxtb          JKlog2com          JKorcaDLPNO        JKsend8
JKCS1_prepare       JKTS              JKfor             JKlog2xyz          JKout2xyz          JKsubg16
JKCS2_explore       JKacdc            JKformation       JKmov2xyz          JKpython           JKtakebest
JKCS3_run           JKattachstr.py    JKforsend         JKmovetoorigin.py  JKqdelall          JKtar
JKCS4_collect       JKavg             JKg16.sh          JKname             JKqmuch            JKxyz2com
JKCS5_filter        JKcheck           JKgaussstat       JKnms              JKremovefiles      JKxyz2inp
JKCS8_clean         JKchmod           JKgoodvibes       JKoptimizer        JKsacct            JKxyz2minp
JKML                JKex              JKjxyz2xyz        JKorca             JKsend
```

2018 – JKCS = Jammy Key for Configurational Sampling

2021 (Aarhus) – JKQC = Jammy Key for Quantum Chemistry

2021 (Aarhus) – JKML = Jammy Key for Machine Learning

Install JK framework

MANUAL: jkcs.rtfd.io

SETUP:

```
git clone https://github.com/kubeckaj/JKCS2.1.git
cd JKCS2.1
sh setup.sh molas -r -qml -nn
source ~/.bashrc
```

LINK OTHER PROGRAMS:

```
vim ~/.JKCSusersetup.txt
```

TEST:

```
sh test.sh
```

TASK

PREPARE WORKSHOP WORKING FOLDER:

```
#mkdir <wrkdir>
cd <wrk-dir>
JKtutorial
#or Grendel people could do:
/home/kubeckaj/JKtutorial
```

LECTURE_1/TASK_00/JKbashrc

Your home folder hidden files

~/.bashrc

```
#history search
bind '"\e[A"':history-search-backward > /dev/null 2>&1
bind '"\e[B"':history-search-forward  > /dev/null 2>&1

#aliases
alias molden='module load molden; unalias molden; molden'

#mistake prevention
alias dc='cd'
alias ccd='cd'
alias cdd='cd'
alias sl='ls'
alias lls='ls'
alias vm=‘mv’

#exports variables/paths
export PATH=$PATH:/home/kubeckaj/Utilities
```

LECTURE_1/TASK_00/JKdircolorsrc

Your home folder hidden files

~/.vimrc

```
" show line numbers
set number
```

(see more at https://gist.github.com/simonista/8703722)

~/.moldenrc

```
background 15
oglbackground 13
```

~/.dircolorsrc

```
#ATR:00=none 01=bold 04=underscore 05=blink 07=reverse 08=concealed
#TC:30=black 31=red 32=green 33=yellow 34=blue 35=magenta 36=cyan 37=white
#BC:40=black 41=red 42=green 43=yellow 44=blue 45=magenta 46=cyan 47=white
.out 01;32
.dat 02;32;41
```

TERMINAL

- get well-familiar with shell commands:
  find, nl/cat/head/tail/more/less, bc/awk, sed, grep, xargs,
  touch, rm, cd, mkdir, ls, sort, wc, du, pwd, ln, echo,
  tar/zip, chmod/chown, man, for/if/while, seq, uniq, paste,
  who, time/date, cut, sort, dos2unix
- get familiar with some SLURM commands:
  = srun, squeue, scontrol, sacct, scancel, sinfo
  = -cpus-per-task, -nodes, -account, -time, -mem, -n,
  --array, --dependency
  = $SLURM_JOB_ID, $SLURM_CPUS_PER_TASK

LECTURE_1/TASK_01/*.xyz

ADVANCED:
-folder <path>
-collect xyz

Structures

What do you use to visualize molecules (Molden, VMD, GaussView)

I hope you know this?

```
cat *.xyz > movie.xyz
molden movie.xyz
```

JKQC can store xyz files too.

```
JKQC *.xyz
JKQC -collect xyz -out db.pkl
```

You can retrieve XYZ files as:

no need to run:

```
JKQC db.pkl -xyz
JKQC db.pkl -movie
```

LECTURE_1/TASK_01/db.pkl

ADVANCED:
-drop <column>
info.txt

The pickled file

JKQC can print some basic info:

```
JKQC db.pkl -info
```

You can use Python to read pickles.

```
JKpython
python
```

or

```
source ~/.JKCSusersetup.txt
program_PYTHON
```

Work in Python:

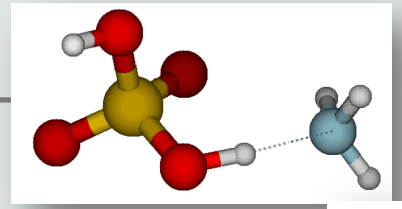
```
import pandas as pd
db = pd.read_pickle("pd.pkl")
print(db)
db. + TAB
help(db.at) #exit manual with <q>
str = db.at[0,("xyz", "structure")]
```

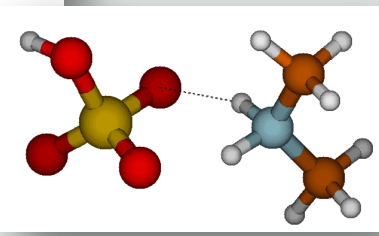
TODO: Calculate molecular mass of the 1. (0-th) structure.

LECTURE_1/TASK_02/*.log

ADVANCED:
-orcaext <str>
-turbomoleext <str>
-levels
-column X Y
ACDB 2.0

Output from QC program





Extract the electronic energy from all Gaussian log files.

```
grep "SCF Done" *.log
```

Create pickle

```
JKQC *.log -out db.pkl
```

TODO: Check the architecture of the pickle. (use: -info)

TODO: Print file basename and electronic energy. (use: -help PRINT)

TODO: Print cluster type + Gibbs free energy. (use: -help PRINT)

TODO: Print the binding/formation properties. (use: -help POST-C. + FORMATION)

TODO: Which cluster (sa_1am_1 or sa_1dma_1) is more stable? (use: brain)

LECTURE_1/TASK_03/db.pkl

ADVANCED:
-index 0 -out x.pkl
-ct vs -b
-pXYZ,-pLOG,-pOUT

Examine new file

Figure out:

TODO: How many structures are in the db.pkl file?

TODO: Visualize at least one molecule.

TODO: Print basename + el. energy of all structures. (use: -help PRINT)

TODO: Print only the lowest energy conformer (use: -help FILTERING)

TODO: Who is the author (username) who created these structures?

TODO: What is that username's password?

```
### ~/.bashrc ###
alias vim='echo LOOOSER!'
#also check: ls -a
```

LECTURE_1/TASK_03/db.pkl

ADVANCED:
-sort b
-filter_== lf nan
-reverse

Removing redundant structures

Filter high energy structures:

```
JKQC db.pkl -b -el -sort el -cut el -694.034 #-filter_lt
JKQC db.pkl -b -el -sort el -cutr el 3.0     #-rel_filter_lt
```

Filter structures with similar energy:

```
JKQC db.pkl -b -el -sort el –uniq el
```

Filter structures with similar energy and gyration radius:

```
JKQC db.pkl -b -el -sort el –uniq el,rg
JKQC db.pkl -b -el -sort el –uniq el,rg1 –noname -noex
```

Compare RMSD between all structures:

```
JKQC db.pkl -b -el -sort el –arbalign 0.3
```

ArbAlign: A Tool for Optimal Alignment of Arbitrarily Ordered Isomers Using the Kuhn–Munkres Algorithm

Berhane Temelso[*†∥], Joel M. Mabey[†], Toshiro Kubota[‡], Nana Appiah-Padi[†§], and George C. Shields[*†∥]

LECTURE_1/TASK_03/db.pkl

ADVANCED:
-extract 1sa*am
-extract 1sa**
-uniq vs -sample
-glob vs -globout
-sort g -select 1 vs -glob

Sample/Select from database

Uniform sampling based on rg,el:

```
JKQC db.pkl -b -el -sort el -sample 8 rg,el
```

Configurational Sampling of Noncovalent (Atmospheric) Molecular Clusters: Sulfuric Acid and Guanidine

Jakub Kubečka*, Vitus Besel, Theo Kurtén, Nanna Myllys, and Hanna Vehkamäki

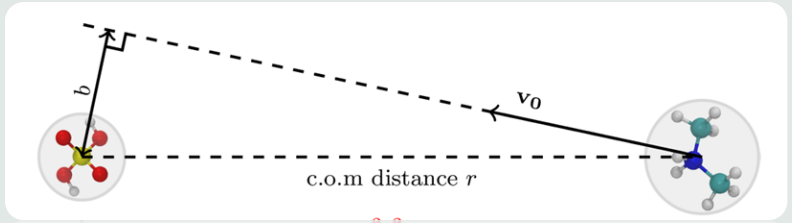
Index selection:

```
JKQC db.pkl -b -el
JKQC db.pkl -b -el -index 0:4
JKQC db.pkl -b -el -index 0:4 -shuffle
JKQC db.pkl -b -el -index 0:4 -shuffle -seed 69
JKQC db.pkl -b -el -index 0:4 -preshuffle
JKQC db.pkl -b -el -index 0:4 -preshuffle -seed 7
```

Extract cluster with specific name:

```
JKQC db.pkl -b -el -extract limonene_acy_TS_95 -noname
```

LECTURE_1/TASK_04/DFT.pkl
LECTURE_1/TASK_04/IMoS



ADVANCED:
-add <column> <file>
-addSP <file>

Combine pickle with your own data

Modeling approaches for atmospheric ion–dipole collisions: all-atom trajectory simulations and central field methods

Ivo Neefjes, Roope Halonen, Hanna Vehkamäki, and Bernhard Reischl

TODO: Print basenames+Gibbs free en.+population of structures in the DFT.pkl file.

TODO: What level of theory did Ivo use?

TODO: Extract Ivo's results [basename CCS] from IMoS/slurm.out to ccs.txt.

use: grep, awk '{print $X " " $Y}', >

TODO: Add the file as a new column to the pickle (use: -add CCS ccs.txt)

TODO: Explain the following command (use: -help, brain, and neighbor):

```
JKQC DFT.pkl -add CCS ccs.txt -ct -extra CCS -noname -bavg -filter_ne extra,CCS nan
```

LECTURE_1/TASK_05/DFTfreq.pkl

Examine lowest vibrational frequencies

TODO: Print basenames + lowest vib. frequencies.

TODO: Extract only the true minima to DFTminima.pkl

TODO: Extract only the failed calculations or structures with imaginary freq.

TODO: Filter only unique structures from DFTminima.pkl (unique based on default filtering for gyration radius, Gibbs free energy, and dipole moment) and save the structures to DFTfiltered.pkl (ArbAlign would take some time)

TODO: What is the probality/population of the lowest free energy minimum?

TODO: How does the population distribution change from 300 K to 30 K.

TODO: Compare gyration radius for the lowest free energy conformer and for Boltzmann average over the whole ensemble.

Vibrational frequencies

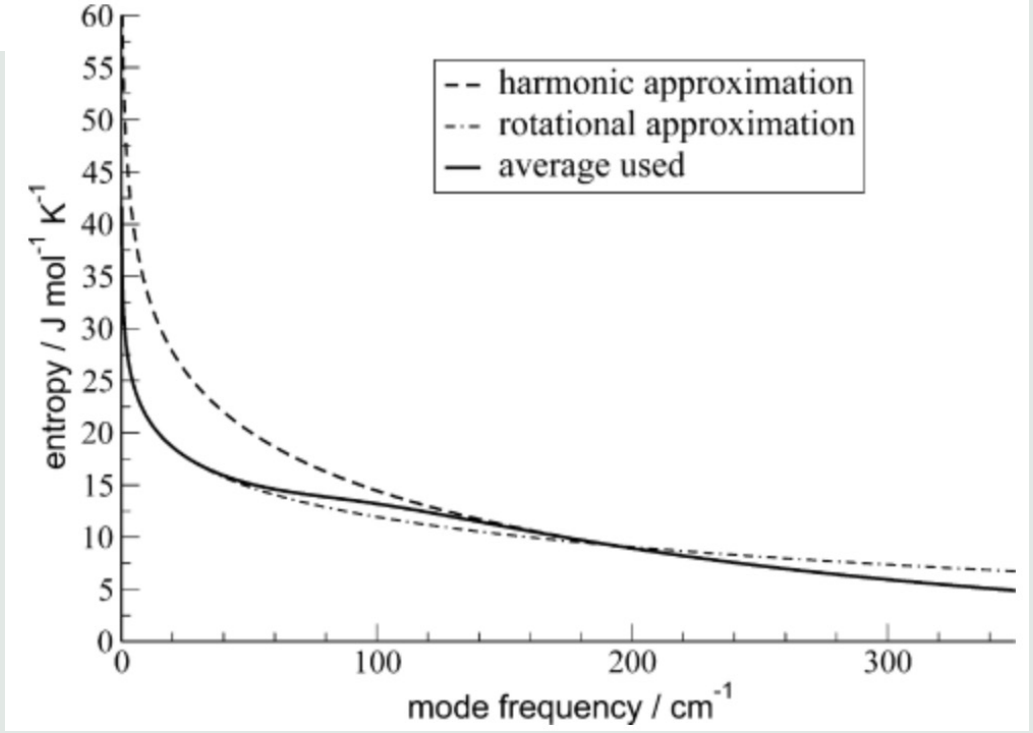
Supramolecular Binding Thermodynamics by Dispersion-Corrected Density Functional Theory

Prof. Dr. Stefan Grimme

Quasi-harmonic approximation (QHA)

Vibration frequency scaling

Precomputed vib. scaling factors:
https://cccbdb.nist.gov/vibscalejustx.asp



LECTURE_1/TASK_05/DFTminima.pkl

Adjust vibrational frequencies

TODO: Select one cluster of your choice and print its frequencies, e.g.:

```
JKQC DFTminima.pkl -extract 4sa3w-43_28_134 -noname -b -freq
```

TODO: Print its Gibbs free energy.

TODO: Print the Gibbs free energy corrected anharmonicity scaling factor of 0.996.

TODO: Print the Gibbs free energy corrected with quasi-harmonic approximation using the vibrational frequency threshold of 100 cm^{-1}.

TODO: Print the Gibbs free energy corrected for both at 278.15 K

TODO: Does anharmonicity scaling/QHA affect: electronic energy/ZPE/vibrational frequencies/enthalpy/entropy? Why yes/not?

LECTURE_1/TASK_06/*.out

Collecting forces

Storing all force components would take quite some amount of memory, so it is not done by default.

Collect forces:

```
JKQC -folder ./ -collect out -forces -out db_forces.pkl
```

We will need those later for ML.

LECTURE_1/TASK_07/DLPNO_DFT.pkl

FORMATION & SOLVATION

TODO: Extract $(SA)_{0-1}(W)_{0-5}$ clusters.

TODO: Print -ct and -g.

TODO: Print -formation

TODO: Add -unit. What is wrong with these values? (··· add XY)

TODO: Select global minimum.

TODO: Change to Boltzmann average.

TODO: Change the free energies to 250 K and properly Boltzmann average.

TODO: Print hydration distribution (-hydration).

TODO: Change humidity to 50%.

TODO: Print actual binding free energies at this humidity and SA concentration of 10^6 cm^{-3}

TODO: Use mass-correct actual binding free energies [e.g., Halonen, Vyslouzil]