Screening Molecular Candidates in Exoplanetary Atmospheres Using High-throughput Quantum Chemistry

Supplementary Material

Juan C. Zapata Trujillo[1], Maria M. Pettyjohn[1],, Laura K. McKemmish[1,*]

[1] *School of Chemistry, University of New South Wales, Sydney, NSW, 2052, Australia*

[*] l.mckemmish@unsw.edu.au

1 Molecular Candidates to SO_2 4.1 μm Spectral Line

In Section 6 in the main manuscript, we discuss how our quantum chemistry big data can provide powerful directions into recognising potential molecular candidates to unknown spectral signals. Specifically, we used the 4.1 μm SO_2 signal on the infrared spectrum of WASP-39b as a proof-of-concept for our approach.

We consider multiple criteria to screen down the list of potential molecular candidates to the 4.1 μm signal: (1) we constrained our search to molecules with predicted frequencies between 2320 and 2620 cm^{-1} (3.8–4.3 μm), (2) we excluded all molecules containing N–H and C–H functional groups, and (3) we only considered molecules with less than or equal to 6 total atoms including H; larger molecules are unlikely to be stable at the planet's temperature.

Our initial predicted fundamental frequency search in the CHNOPS2743-HARMONIC dataset did not provide promising molecular candidates. Instead, we ran VPT2 anharmonic calculations (input file template provided at the end of this document) to look for overtones and combination bands. The analysis of the most promising molecular candidates is outlined in Section 6 in the main manuscript. Here, in Table S1, we present the full list of potential molecular candidates along with their corresponding calculated frequencies and transition intensities.

2 Master csv File for the CHNOPS2743-HARMONIC Dataset

The file B971_def2TZVPD_GroundState.csv contains all the harmonic frequency and intensity values for all 2743 molecules in our dataset. Below is the description of all columns in the csv file in their corresponding order:

(i) **Formula:** Molecular formula.

(ii) **IUPAC_chemname:** IUPAC name for molecule.

(iii) **SMILES:** SMILES identifier.

(iv) **Atoms:** Total number of atoms in the molecule.

(v) **RawFreq [cm-1]:** Raw harmonic frequencies output by the Gaussian quantum chemistry package.

(vi) **ScaledFreq [cm-1]:** The raw harmonic frequencies scaled by the scaling factors reported in Zapata Trujillo & McKemmish (2023).

(vii) **Intensity:** Intensities of the harmonic frequencies (km/mole).

(viii) **mu_A [D]:** Dipole moment along the A principal axis (D).

(ix) **mu_B [D]:** Dipole moment along the B principal axis (D).

(x) **mu_C [D]:** Dipole moment along the C principal axis (D).

(xi) **mu_tot [D]:** Total dipole moment (D).

(xii) **A [cm-1]:** Rotational constant A (cm^{-1}).

(xiii) **B [cm-1]:** Rotational constant B (cm^{-1}).

(xiv) **C [cm-1]:** Rotational constant C (cm^{-1}).

Table S1. Statistical metrics for the B97-1/def2-TZVPD model chemistry in harmonic frequency calculations as described in Zapata Trujillo & McKemmish (2023). Note that the errors in are given at different wavelengths.

Formula	SMILES	Calculated Frequency		Band Type	Intensity
		in cm^{-1}	in μm		
H_2O_3S	O=S(O)O	2477	4.04	Overtone	0.69
COS	O=C=S	2609	3.83	CombBand	0.00
O_3	O=[O+][O-]	2513	3.98	Overtone	0.01
		2425	4.12	Overtone	0.12
		2458	4.07	CombBand	8.04
		2590	3.86	CombBand	0.01
		2421	4.13	CombBand	0.12
CN_2O_3	C(=O)(N=O)N=O	2564	3.90	CombBand	2.18
		2397	4.17	CombBand	0.64
		2491	4.01	CombBand	0.50
		2321	4.31	CombBand	0.04
		2384	4.19	CombBand	0.00
HOP	P=O	2473	4.04	Overtone	0.70
		2448	4.09	Overtone	63.82
C_2N_2S	N#CN=C=S	2588	3.86	CombBand	0.36
		2587	3.87	CombBand	0.12
		2551	3.92	CombBand	6.47
		2533	3.95	CombBand	0.16
		2379	4.20	CombBand	27.33
CN_2O_2	N#CN=O	2389	4.19	CombBand	0.43
		2483	4.03	CombBand	0.11
		2446	4.09	CombBand	0.34
C_3S_2	C(=C=S)=C=S	2610	3.83	CombBand	0.02
		2535	3.95	CombBand	0.00
		2535	3.95	CombBand	0.00
CHNO_1	OC#N	2380	4.20	Overtone	2.69
		2479	4.03	CombBand	0.10
		2463	4.06	CombBand	0.00
		2329	4.29	CombBand	1.17
C_3N_2O	O=C(C#N)C#N	2592	3.86	CombBand	0.10
		2549	3.92	CombBand	0.04
		2545	3.93	CombBand	0.16
		2509	3.99	CombBand	0.15
		2421	4.13	CombBand	0.26
O_3S	O=S(=O)=O	2418	4.14	CombBand	0.07
		2456	4.07	CombBand	2.40
		2458	4.07	CombBand	2.39
		2532	3.95	CombBand	0.23
		2439	4.10	CombBand	0.60
CN_2OS	N#CSN=O	2597	3.85	CombBand	0.26
		2581	3.87	CombBand	0.30
		2473	4.04	CombBand	0.16
		2394	4.18	CombBand	0.00
HO_2P	O=PO	2541	3.94	Overtone	0.86
		2390	4.18	CombBand	1.64
		2482	4.03	CombBand	26.49
		2431	4.11	CombBand	2.53
C_2N_2O	N#CN=C=O	2464	4.06	Overtone	1.47
		2332	4.29	CombBand	0.64
		2542	3.93	CombBand	0.11
		2422	4.13	CombBand	0.18
		2395	4.18	CombBand	0.20
HNO_3	O[N+]([O-])=O	2553	3.92	Overtone	2.28
		2601	3.84	CombBand	3.85
		2609	3.83	CombBand	0.71
		2484	4.02	CombBand	0.00
		2361	4.24	CombBand	0.13
H_2O_2	OO	2545	3.93	Overtone	0.54
		2333	4.29	CombBand	0.07
SO_2	O=S=O	2367	4.23	Overtone	0.29
		2564	3.90	CombBand	3.15

3 VPT2 Anharmonic Data csv File

The file SmallMolecules_VPT2Data.csv contains the calculated vPT2 anharmonic data for 77 small molecules (i.e., with less than 6 total atoms) in the CHNOPS2734-HARMONIC set. Below is the description of all columns in this csv file in their corresponding order:

(i) **Formula:** Molecular formula.

(ii) **SMILES:** SMILES identifier.

(iii) **Atoms:** Total number of atoms in the molecule.

(iv) **Frequencies [cm-1]:** List of calculated frequencies output by the Gaussian quantum chemistry package.

(v) **Freq_Kind:** List outlining the type of frequency calculated, i.e., harmonic, fundamental, overtone or combination band.

(vi) **Intensity:** Intensities of the harmonic frequencies (km/mole).

Note that we have no clear understating of the errors associated with this data as no thorough benchmarking of anharmonic approaches is currently available.

4 Sample Calculation Input

All calculations were run on the Katana Supercomputer located on the University of New South Wales, Kensington Campus in Sydney, New South Wales, Australia.

4.1 GFN2-xTB Cheap Geometry Optimisation

All molecules were optimised using the `xtb` program (Bannwarth et al. 2019) using the terminal bash command below:

```
xtb {molecule}.xyz --opt gfn2
```

where `{molecule}` is the name of the molecule geometry to be optimised. The `--opt` flag performs a singlepoint geometry optimisation on the input `xyz` file. The `gfn2` option performs the geometry optimisation with the GFN2-xTB method.

4.2 CREST and CENSO Calculations

For further details for setting up the CREST and CENSO algorithms, we recommend looking into the official documentation at https://xtb-docs.readthedocs.io/en/latest/CENSO_docs/censo_setup.html.

4.2.1 CREST Conformer Ensamble Generation

All output `xtb` molecule geometries are input into the `CREST` program (Pracht et al. 2020) to obtain a set of conformer energies and corresponding geometries with the following terminal bash command:

```
crest {molecule}.xyz > {molecule}_crest.out
```

where the `out` file contains the geometries of all conformers computed by `CREST`.

4.2.2 CENSO Conformer Refinement

All output `CREST` conformer geometries are input into the `CENSO` program (Grimme et al. 2021). The `CENSO` input file for one molecule is shown below.

```
#!/bin/bash

#PBS -N {molecule}_censo
#PBS -l select=1:ncpus=1:mem=4gb
#PBS -l walltime=04:00:00
#PBS -j oe
#PBS -m ae
```

```
cd $PBS_O_WORKDIR

module load python/3.7.4
module load openmpi/2.1.6
module load orca/5.0.3
module load intel/2020.0.088
module load xtb/6.4.0

### setup programs
## XTB
export OMP_NUM_THREADS=1
export MKL_NUM_THREADS=1
ulimit -s unlimited
export OMP_STACKSIZE=1000m

## TM
export PARA_ARCH=SMP
source /home/$USER/bin/.turbo751
export PARNODES=4  ## omp
export TM_PAR_FORK=1

### ORCA4.2.1
ORCAPATH="/apps/orca/5.0.3";
MPIPATH="/apps/openmpi/2.1.6/bin:";
MPILIB="//apps/openmpi/2.1.6/bin/lib64";
PATH=${ORCAPATH}:${MPIPATH}:$PATH
LD_LIBRARY_PATH=${ORCAPATH}:${MPILIB}:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=/software/intel/parallel_studio_xe_2017.1/
parallel_studio_xe_2017.4.056/compilers_and_libraries_2017/
linux/compiler/lib/intel64_lin:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=/software/intel/parallel_studio_xe_2017/
mkl/lib/intel64:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=${ORCAPATH}:${MPILIB}:$LD_LIBRARY_PATH

## PATH
PATH=/home/$USER/bin:$PATH
export PATH
### end programs + PATH

export HOSTS_FILE=$PBS_NODEFILE
cat $HOSTS_FILE>hosts_file

TMP_DIR=/home/$USER/temp_{molecule}
DIR1=$PWD

mkdir -p $TMP_DIR/$PBS_JOBID

#check file system access
if [ ! -d $TMP_DIR/$PBS_JOBID ]; then
echo "Unable to create $TMP_DIR/$PBS_JOBID  on $HOSTNAME. Must stop."
exit
fi

#check current location
if [ "$PWD" == "$HOME" ]; then
echo "Cowardly refusing to copy the whole home directory"
exit
```

```
fi

#copy everything to node (will NOT copy directories for safety reasons.
#Add an 'r' only if absolutely sure what you are doing)
#bwlimit limits bandwidth to 5000 kbytes/sec

rsync -q --bwlimit=5000 $DIR1/* $TMP_DIR/$PBS_JOBID/
rsync -rq --ignore-missing-args --bwlimit=5000 $DIR1/CONF* $TMP_DIR/$PBS_JOBID/
rsync -q --bwlimit=5000 $DIR1/.* $TMP_DIR/$PBS_JOBID/
cd $TMP_DIR/$PBS_JOBID

###############################################################################
#Gettimings

start=$(date +%s)
###############################################################################
#jobs start here (if you have no idea what this script does, only edit this part...)

echo "Calculation from $(date)" >> RUNTIME
export PYTHONUNBUFFERED=1

/path/to/censo -inp {molecule}_conformers.xyz > {molecule}_censo_conformers.out

#end of job     (....and stop editing here.)
###############################################################################
#Print timings to file
end=$(date +%s)
secs=$(expr $end - $start)
printf '%dh:%dm:%02ds\n' $(($secs/3600)) $(($secs%3600/60)) $(($secs%60)) >> RUNTIME
###############################################################################
#copy everything back that is smaller than 5 gbytes

rsync -rq --bwlimit=5000 --max-size=5G $TMP_DIR/$PBS_JOBID/* $DIR1/
rsync -q --bwlimit=5000 --max-size=5G $TMP_DIR/$PBS_JOBID/.* $DIR1/

#to be safe, get mos alpha and beta seperately.
#Note that the rsync syntax is strange; you need to first include everything,
#then exclude the rest ("*" includes subdirectories)

rsync -rq --bwlimit=5000 --include="*/" --include="mos" --include="alpha"
                                                       --include="beta"
--exclude=* $TMP_DIR/$PBS_JOBID/* $DIR1/

#if you want the large files as well, comment in the following

#rsync -r --bwlimit=1000 --min-size=5G $TMP_DIR/$PBS_JOBID/* $DIR1/

cd $DIR1
rm -r $TMP_DIR/$PBS_JOBI
```

A majority of the input file sets up supercomputer settings, file and directory paths, and loads modules. The `CENSO` program generates temporary directories in the location designated by the `TMP_DIR` variable, copying the temporary directory into your local directory once the calculation is complete. The most important line is

```
/path/to/censo -inp {molecule}_conformers.xyz > {molecule}_censo_conformers.out
```

where the {`molecule`}`_conformers.xyz` is output by the `CREST` program and contains all the computed conformer geometries

for the molecule with the name {molecule}. In addition to the output out file detailing the computation, the CENSO program also outputs a dat and xyz file containing the Boltzmann weights and geometries of the optimised conformers, respectively.

The global parameter file .censorc for both CREST and CENSO programs is shown below.

```
$CENSO global configuration file: .censorc
$VERSION:1.2.0

ORCA: /apps/orca/5.0.3
ORCA version: 5.0.3
GFN-xTB: /apps/xtb/6.4.0/bin/xtb
CREST: /apps/xtb/6.4.0/bin/crest
mpshift: /path/including/binary/mpshift-binary
escf: /path/including/binary/escf-binary

#COSMO-RS
ctd = BP_TZVP_C30_1601.ctd cdir = "/software/cluster/COSMOthermX16/COSMOtherm
/CTDATA-FILES" ldir = "/software/cluster/COSMOthermX16/COSMOtherm/CTDATA-FILES"
$ENDPROGRAMS

$CRE SORTING SETTINGS:
$GENERAL SETTINGS:
nconf: all                        # ['all', 'number e.g. 10 up to all conformers']
charge: 0                         # ['number e.g. 0']
unpaired: 0                       # ['number e.g. 0']
solvent: gas                      # ['gas', 'acetone', 'acetonitrile', 'aniline',
                                     'benzaldehyde', 'benzene', 'ccl4', '...']
prog_rrho: xtb                    # ['xtb']
temperature: 298.15               # ['temperature in K e.g. 298.15']
trange: [273.15, 378.15, 5]  # ['temperature range [start, end, step]']
multitemp: on                     # ['on', 'off']
evaluate_rrho: on                 # ['on', 'off']
consider_sym: off                 # ['on', 'off']
bhess: off                        # ['on', 'off']
imagthr: automatic                # ['automatic or e.g., -100    # in cm-1']
sthr: automatic                   # ['automatic or e.g., 50     # in cm-1']
scale: automatic                  # ['automatic or e.g., 1.0 ']
rmsdbias: off                     # ['on', 'off']
sm_rrho: alpb                     # ['alpb', 'gbsa']
progress: on                      # ['on', 'off']
check: on                         # ['on', 'off']
prog: orca                        # ['tm', 'orca']
func: r2scan-3c                   # ['b3-lyp', 'b3lyp', 'b3lyp-3c', 'b3lyp-d3',
                                     'b3lyp-d3(0)', 'b3lyp-d4', 'b3lyp-nl', '...']
basis: automatic                  # ['automatic', 'def2-TZVP', 'def2-mSVP',
                                     'def2-mSVP', 'def2-mSVP', 'def2-mSVP', '...']
maxthreads: 4                     # ['number of threads e.g. 2']
omp: 1                            # ['number cores per thread e.g. 4']
balance: off                      # ['on', 'off']
cosmorsparam: automatic           # ['automatic', '12-fine', '12-normal',
                                     '13-fine', '13-normal', '14-fine', '...']

$PART0 - CHEAP-PRESCREENING - SETTINGS:
part0: on                         # ['on', 'off']
func0: b97-d3                     # ['b3-lyp', 'b3lyp', 'b3lyp-3c', 'b3lyp-d3',
                                     'b3lyp-d3(0)', 'b3lyp-d4', '...']
basis0: def2-SV(P)                # ['automatic', 'def2-SV(P)', 'def2-TZVP',
                                     'def2-mSVP', 'def2-mSVP', 'def2-mSVP', '...']
```

```
part0_gfnv: gfn2                # ['gfn1', 'gfn2', 'gfnff']
part0_threshold: 4.0            # ['number e.g. 4.0']

$PART1 - PRESCREENING - SETTINGS:
# func and basis is set under GENERAL SETTINGS
part1: on                       # ['on', 'off']
smgsolv1: smd                   # ['alpb_gsolv', 'cosmo', 'cosmors', 'cosmors-fine',
                                   'cpcm', 'dcosmors', '...']
part1_gfnv: gfn2                # ['gfn1', 'gfn2', 'gfnff']
part1_threshold: 3.5            # ['number e.g. 5.0']

$PART2 - OPTIMIZATION - SETTINGS:
# func and basis is set under GENERAL SETTINGS
part2: on                       # ['on', 'off']
prog2opt: orca                  # ['tm', 'orca', 'prog', 'automatic']
part2_threshold: 2.5            # ['number e.g. 4.0']
sm2: smd                        # ['cosmo', 'cpcm', 'dcosmors', 'default', 'smd']
smgsolv2: smd                   # ['alpb_gsolv', 'cosmo', 'cosmors', 'cosmors-fine',
'cpcm', 'dcosmors', '...']
part2_gfnv: gfn2                # ['gfn1', 'gfn2', 'gfnff']
ancopt: on                      # ['on']
hlow: 0.01                      # ['lowest force constant in ANC
                                   generation, e.g. 0.01']
opt_spearman: on                # ['on', 'off']
part2_P_threshold: 99           # ['Boltzmann sum threshold in %. e.g. 95
                                   (between 1 and 100)']
optlevel2: automatic            # ['crude', 'sloppy', 'loose', 'lax', 'normal',
                                   'tight', 'vtight', 'extreme', '...']
optcycles: 8                    # ['number e.g. 5 or 10']
spearmanthr: -4.0               # ['value between -1 and 1, if outside
                                   set automatically']
radsize: 10                     # ['number e.g. 8 or 10']
crestcheck: on                  # ['on', 'off']

$PART3 - REFINEMENT - SETTINGS:
part3: on                       # ['on', 'off']
prog3: orca                     # ['tm', 'orca', 'prog']
func3: pw6b95                   # ['b3-lyp', 'b3lyp', 'b3lyp-3c', 'b3lyp-d3',
                                   'b3lyp-d3(0)', 'b3lyp-d4', 'b3lyp-nl', '...']
basis3: def2-TZVPD              # ['DZ', 'QZV', 'QZVP', 'QZVPP', 'SV(P)',
                                   'SVP', 'TZVP', 'TZVPP', 'aug-cc-pV5Z', '...']
smgsolv3: smd                   # ['alpb_gsolv', 'cosmo', 'cosmors', 'cosmors-fine',
                                   'cpcm', 'dcosmors', '...']
part3_gfnv: gfn2                # ['gfn1', 'gfn2', 'gfnff']
part3_threshold: 99             # ['Boltzmann sum threshold in %.
                                   e.g. 95 (between 1 and 100)']

$NMR PROPERTY SETTINGS:
$PART4 SETTINGS:
part4: off                      # ['on', 'off']
couplings: on                   # ['on', 'off']
progJ: prog                     # ['tm', 'orca', 'prog']
funcJ: pbe0                     # ['b3-lyp', 'b3lyp', 'b3lyp-3c', 'b3lyp-d3',
                                   'b3lyp-d3(0)', 'b3lyp-d4', 'b3lyp-nl', '...']
basisJ: def2-TZVP               # ['DZ', 'QZV', 'QZVP', 'QZVPP', 'SV(P)',
                                   'SVP', 'TZVP', 'TZVPP', 'aug-cc-pV5Z', '...']
sm4J: smd                       # ['cosmo', 'cpcm', 'dcosmors', 'smd']
```

```
shieldings: on                  # ['on', 'off']
progS: prog                     # ['tm', 'orca', 'prog']
funcS: pbe0                     # ['b3-lyp', 'b3lyp', 'b3lyp-3c', 'b3lyp-d3',
                                   'b3lyp-d3(0)', 'b3lyp-d4', 'b3lyp-nl', '...']
basisS: def2-TZVP               # ['DZ', 'QZV', 'QZVP', 'QZVPP', 'SV(P)',
                                   'SVP', 'TZVP', 'TZVPP', 'aug-cc-pV5Z', '...']
sm4S: smd                       # ['cosmo', 'cpcm', 'dcosmors', 'smd']
reference_1H: TMS               # ['TMS']
reference_13C: TMS              # ['TMS']
reference_19F: CFCl3            # ['CFCl3']
reference_29Si: TMS             # ['TMS']
reference_31P: TMP              # ['TMP', 'PH3']
1H_active: on                   # ['on', 'off']
13C_active: on                  # ['on', 'off']
19F_active: off                 # ['on', 'off']
29Si_active: off                # ['on', 'off']
31P_active: off                 # ['on', 'off']
resonance_frequency: 300.0  # ['MHz number of your experimental spectrometer setup']

$OPTICAL ROTATION PROPERTY SETTINGS:
$PART5 SETTINGS:
optical_rotation: off           # ['on', 'off']
funcOR: pbe                     # ['functional for opt_rot e.g. pbe']
funcOR_SCF: r2scan-3c           # ['functional for SCF in opt_rot e.g. r2scan-3c']
basisOR: def2-SVPD              # ['basis set for opt_rot e.g. def2-SVPD']
frequency_optical_rot: [589.0]  # ['list of freq in nm to evaluate
                                   opt rot at e.g. [589, 700]']
$END CENSORC
```

The `PART4`, corresponding to nuclear magnetic resonance properties, and `PART5`, corresponding to optical rotation, settings are turned off. The remaining settings are set to those suggested in the **?** and **?** papers to efficiently obtain the most accurate energies, i.e., the r2scan-3c and pw6b95 methods. Generally, we perform gas phase calculations with the `solvent` setting set to gas in a temperature range of 273.15 to 378.15 K. All `CREST` and `CENSO` calculations are performed with the ORCA cite quantum chemistry package.

4.3 Harmonic Frequency Calculations

The harmonic frequency calculations were run for each molecule with the Gaussian quantum chemistry package (Frisch et al. 2009) using the template input file below.

```
# B971/Gen NoSymm Opt=(Cartesian,CalcFC,Tight,MaxCycles=100) Int=Ultrafine
  Freq=VibRot output=pickett

Harmonic frequencies for {molecule}

{charge} {multiplicity}
{geometry}
{basis_data}
```

where `B971` corresponds to the method (Hamprecht et al. 1998) and `Gen` corresponds to the input basis set def2-TZVPD (Schäfer et al. 1994). The basis set information is added below the {`basis_data`} section. The groundstate conformer geometry was input below the {`geometry`} section because the def2-TZVPD basis set is not built-in in the current version of Gaussian (it needs to be added manually from `https://www.basissetexchange.org/`). The {`charge`} and {`multiplicity`} were set according to the molecule.

4.4 VPT2 Anharmonic Calculations

We ran VPT2 anharmonic calculations for a subset of small molecules in our dataset to calculate overtones and combination bands, which are forbidden wirthin the harmonic approximation. We used the input file below to run these calculations.

```
# B971/Gen NoSymm Opt=(Cartesian,CalcFC,Tight,MaxCycles=100) Int=Ultrafine
  Freq=(Anharmonic,ReadAnharm,VibRot) output=pickett

VPT2 Anharmonic Calculation for {molecule}

{charge} {multiplicity}
{geometry}
{basis_data}

PT2Model=GVPT2
```

BIBLIOGRAPHY

Bannwarth C., Ehlert S., Grimme S., 2019, Journal of Chemical Theory and Computation, 15, 1652
Frisch M. J., et al., 2009, Gaussian09 Revision D.01
Grimme S., Bohle F., Hansen A., Pracht P., Spicher S., Stahn M., 2021, Journal of Physical Chemistry A, 125, 4039â4054
Hamprecht F. A., Cohen A. J., Tozer D. J., Handy N. C., 1998, Journal of Chemical Physics, 109, 6264
Pracht P., Bohle F., Grimme S., 2020, Physical Chemistry Chemical Physics, 22, 7169
Schäfer A., Huber C., Ahlrichs R., 1994, The Journal of Chemical Physics, 100, 5829
Zapata Trujillo J. C., McKemmish L. K., 2023, The Journal of Physical Chemistry A, 127, 1715