

ConStruQt[®] High throughput quantum chemistry
for molecular design
by ChemAlive



Quantum chemical accuracy (>10x improvement)

5,000 or more molecules per minute

Conformational, tautomeric and stereoisomeric analysis

October, 2019
Beta Release
(ConStruQt-API beta 1.0)
info@chemalive.com

1	INTRODUCTION TO CONSTRUQT-API	3
1.1	MOTIVATION.....	3
1.2	TECHNICAL DESCRIPTION	3
1.2.1	<i>ConStruQt Chemistry</i>	4
1.2.2	<i>ConStruQt Software/Infrastructure</i>	7
2	CONSTRUQT-API INSTRUCTIONS.....	7
2.1	ACCESS AND USE OF THE API (BRANDING).....	7
2.1.1	<i>Obtaining an API Key</i>	7
2.1.2	<i>Restrictions</i>	7
2.2	API CALLS	8
2.2.1	<i>API-Submit</i>	8
2.2.2	<i>API-Read</i>	10
2.3	INTERPRETING THE JSON.....	11
2.3.1	<i>Structure of the JSON Output</i>	11
2.3.2	<i>Analysis Example: Uracil</i>	19
3	TERMS OF SERVICE	20
3.1	ACCESS TO OUR APIS	20
3.2	RULES	20
3.3	TRANSPARENCY AND REPORTING.....	21
3.4	PRIVACY POLICY	21
3.5	RIGHT TO SUSPEND ACCESS AND AUDIT.....	21
3.6	OWNERSHIP AND PROPERTY RIGHTS	21
3.7	TERMINATION.....	22
3.8	REPRESENTATIONS; DISCLAIMER OF WARRANTIES	22
3.9	LIMITATIONS OF LIABILITY	22
3.10	APPLICATION OF CONSUMER LAW	22
3.11	YOUR INDEMNIFICATION OF US.....	23
3.12	LIMITATIONS ON INDEMNIFICATIONS	23
3.13	SURVIVAL	23
3.14	GENERAL PROVISIONS	23
3.15	MODIFICATION TO THE CONTRACT.....	24
3.16	WAIVER	24
3.17	SEVERABILITY.....	24
3.18	ASSIGNMENT.....	24
3.19	ENTIRE AGREEMENT	24
4	APPENDIX	24
4.1	WEB LINKS	24

1 Introduction to ConStruQt-API

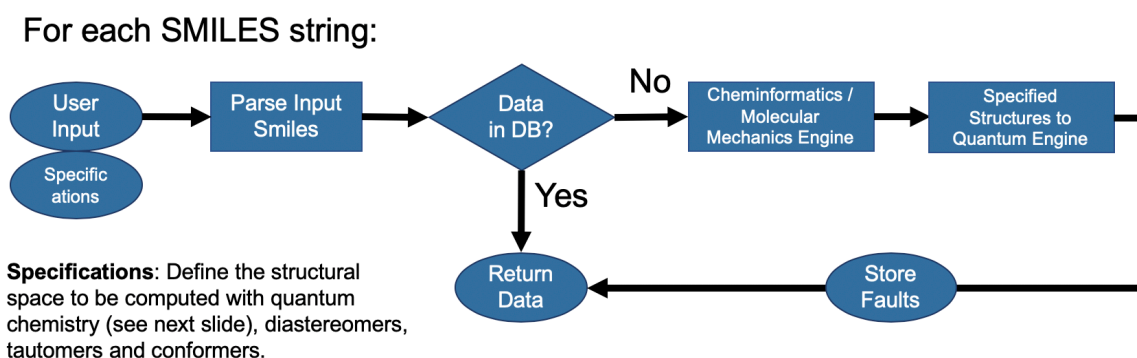
1.1 Motivation

ConStruQt-API is a member of the Qt series of APIs under development by ChemAlive. The Qt series delivers fully automated hassle-free web access to advanced quantum chemical calculations for small to medium-sized organic molecules. The APIs are designed for high throughput applications suited for analysis of large chemical libraries and for extracting trend and statistically meaningful data from collections of high-value target compounds. The Qt series aims to displace classical mechanics and rules-based cheminformatics with quantitative predictive analytics based on modern physics and quantum chemistry.

With high throughput quantum chemistry, accurate chemical structures, molecular energies and electronic properties become available yielding a sea-change in predictability while accessing a big data regime for advanced analytics of self-consistent computed datasets. ConStruQt-API delivers 10x more accurate molecular structures and energetics compared to classical mechanics (force field) approaches and allows the sorting and prioritizing chemical structures at a scale of thousands of molecules.

1.2 Technical Description

ConStruQt-API is a molecular structure management tool that allows quantitative prioritization of chemical structures including tautomers, stereoisomers and conformers based on quantum chemical energetics. It connects web-based cheminformatics tools to advanced quantum chemistry routines, cloud infrastructure and the world's largest database of quantum chemical data, called Qontext. Directly from SMILES descriptions. ConStruQt-API manages the 2D and 3D structural description of new molecules, launches quantum chemical calculations, verifies their outcomes and stores the data. For molecules already existing in Qontext, the API returns the stored data. The beta version delivers only semi-empirical PM6 data, but more advanced QM methods are available by request (info@chemalive.com). The top-level workflow is shown in Scheme 1.



Scheme 1. Top-level workflow of ConStruQt Engine.

The input SMILES is processed by the Cheminformatics/Molecular Mechanics and Quantum Engines according to user Specifications and quality controlled by each engine's algorithm (described below). Inputs are checked for redundancy with existing data in the ChemAlive database, Qontext.

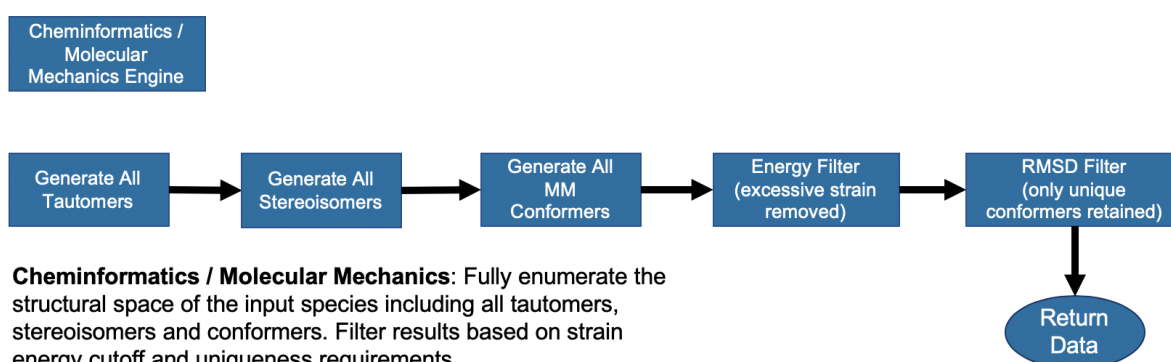
1.2.1 ConStruQt Chemistry

The **ConStruQt** default structure management routine (cheminformatics/mechanics) follows the following main steps:

- (1) Conversion of input SMILES to canonical form, rejection of false inputs (e.g. salts or complexes, metal containing molecules).
- (2) Rejection of large macromolecules (current default limit is 200 atoms).
- (3) Enumeration of tautomeric SMILES and generation of InChI for comparison.
- (4) Enumeration of stereoisomeric SMILES, removal of identical meso SMILES.
- (5) Generation of conformations of each unique SMILES at the Universal Force Field (UFF) level.
- (6) Removal of very strained conformations, removal of identical conformations.

These steps are represented in Scheme 2, below.

For each SMILES string:



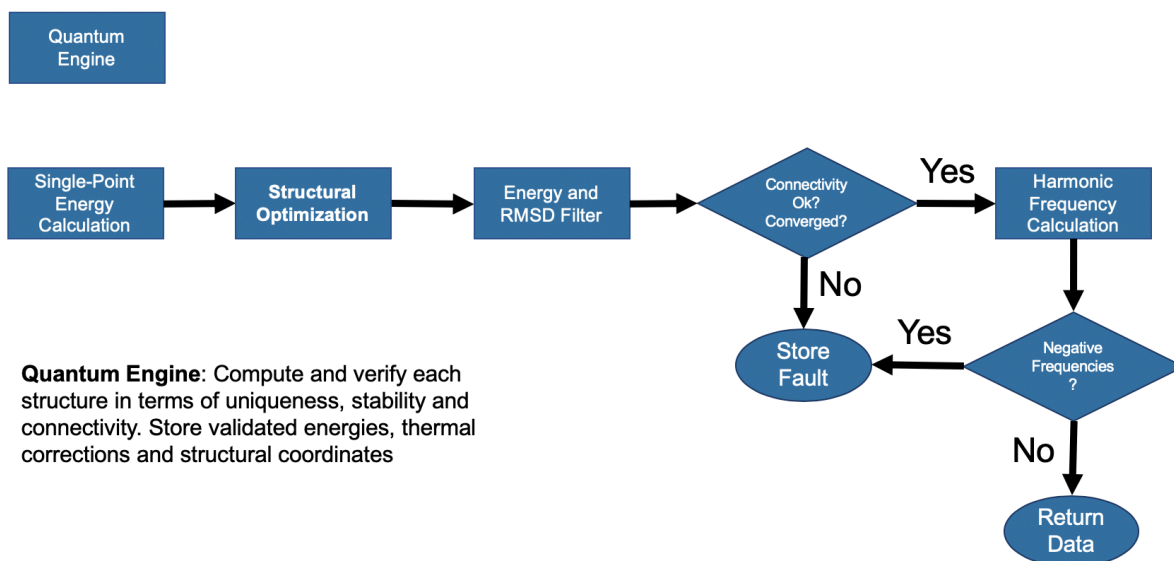
Scheme 2. Cheminformatics / Molecular Mechanics Engine workflow.

The **ConStruQt** default Quantum Engine routine (quantum mechanics) follows the following main steps:

- (1) Single-point calculation at the PM6 semi-empirical level on the input UFF structure.
- (2) Structural optimization at the PM6 level.
- (3) Removal of identical 3D structures.
- (4) Validation that the molecular connectivity of the input structure is the same as the output structure.
- (5) Harmonic frequency calculation at the PM6 level to obtain thermal corrections (zero-point energy, enthalpy, entropy).
- (6) Validation that the obtained structure is a minimum on the potential energy surface.

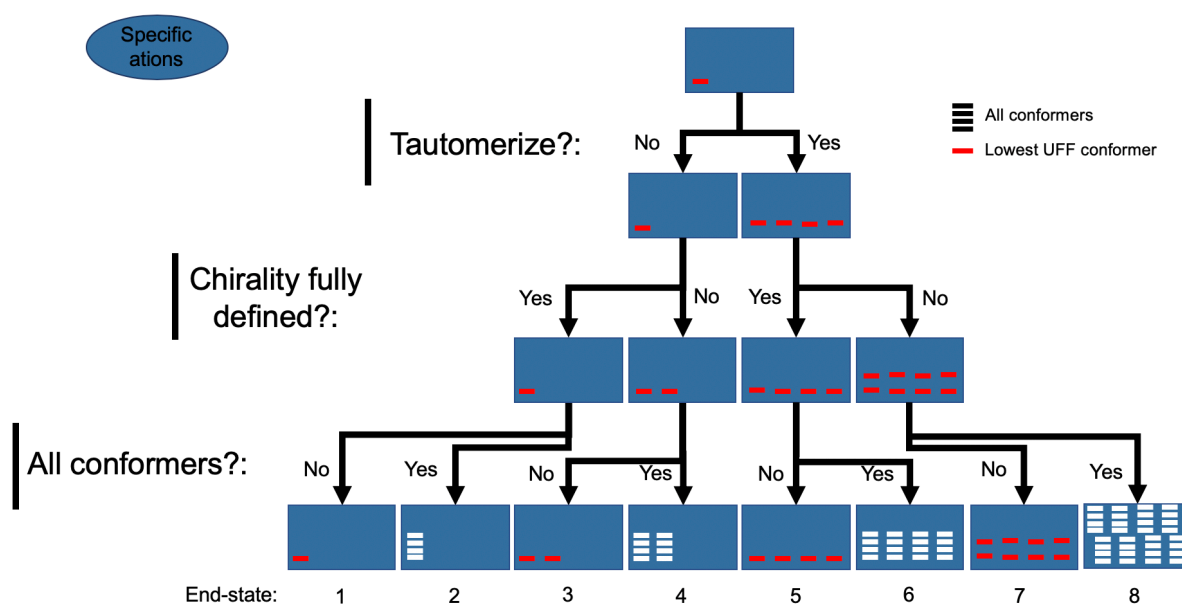
The workflow is represented in Scheme 3, below.

For each structure:



Scheme 3. Quantum Engine workflow.

Once these structures are enumerated/generated the API calls manage the set of structures that are passed on to the Quantum Engine (see API calls) according to user Specifications. This allows the user to control the latency of the query and to select the most relevant information to process, which is necessary due to the slower speed of quantum chemical calculations. Eight End-states are accessible through two user-controlled toggles (allTautomers / allConformers) and an automatic stereoisomer check. The decision tree is as follows in Scheme 4.



Scheme 4. User specification decision tree to arrive at End-states 1 - 8.

The default behavior is to arrive at End-state 1 – a single structure calculated with quantum chemistry (red line). Only the user inputted molecule is computed. No tautomers are computed with QM. Assuming the user has fully specified all chirality in the molecule, no other stereoisomers are computed with QM. Finally, the lowest energy conformer, as determined by the universal force field (UFF), is selected for further quantum chemical calculation.

Stereoisomers are treated automatically by the **ConStruQt** routine. It is left up to the user to fully specify all chiral centers of their input molecules. Any ambiguity will be treated as intentional, i.e. by not specifying chirality you are asking the engine to compute all stereoisomers of the molecule.

If the user specifies the on-state of the allConformer toggle End-state 2 will result, where all UFF conformers are passed to the Quantum Engine.

End-state 3 is the default behavior when the user has entered a SMILES string with an unspecified chiral center (allConformers and allTautomers are off). The lowest energy conformer for each according to UFF is returned.

End-state 4 is the result of the allConformer toggle switch on for a chirality unspecified molecule.

End-state 5 is the result of the on-state for the allTautomer toggle with off-state for the allConformer toggle with a fully specified (or achiral) input SMILES string.

End-state 6 results from both the allTautomer and allConformer toggle switched on for a fully specified stereoisomer (or achiral molecule).

End-state 7 is arrived at by the on-state for allTautomer toggle, off-state for the allConformer toggle and an unspecified chirality for input molecule.

End-state 8 results from an unspecified chiral molecule as input and the allTautomer and allConformer toggles switched.

Once all data is computed and stored it is possible to continue with further quantum chemical calculations using NWCHEM software that gives access to advanced electronic structure theory and properties. For bespoke follow-on routines contact info@chemalive.com.

1.2.2 ConStruQt Software/Infrastructure

The remote **ConStruQt**-API runs over HTTP using JSON-RPC. It connects to a backend (Scala) that manages API implementation, distributed computation scheduling and user accounts. Submitted jobs run on a 6000+ core Apache Spark computational cluster (Amazon Web Services) using spot computing nodes. The engine (C++) manages a number of open-sources softwares that handle molecular constitution and 3D structure using cheminformatics, molecular mechanics and quantum chemistry. Validated results are stored on an RDS volume in a relational (PostgreSQL) database currently serving 200 M+ structures and 3.5 M+ molecules. The components are summarized in Figure 1.

Remote-API	JSON-RPC over HTTP
Backend	Remote API implementation, distributed computation scheduling, user management (Scala)
Engine	Molecular constitution and 3D structure management using cheminformatics, molecular mechanics and quantum chemistry (C++)
Database	Relational (PostgreSQL) serving 200 M+ structures and 3.5 M molecules
Infrastructure	6000+ core Apache Spark computational cluster (Amazon Web Services) + 4 x 72 core development servers

Figure 1. Components of ConStruQt-API.

The Engine wraps a number of open-source softwares that contribute a diverse set of routines that allow **ConStruQt**'s final functionality. These include RDKit, Openbabel, Cp2k and NWChem.

2 ConStruQt-API Instructions

2.1 Access and Use of the API (Branding)

ConStruQt is currently operated in a restricted freemium configuration. If you decide to deploy the API we expect that proper referrals are made to ChemAlive through a clickable company logo found here at www.chemalive.com with the subtext "For larger scale projects, more advanced calculations or bespoke projects please contact info@chemalive.com." The link should refer to the company main page, www.chemalive.com.

2.1.1 Obtaining an API Key

Access to the API is through a supplied APIkey. These keys are distributed on a case-by-case basis. Get in touch at info@chemalive.com for further information.

2.1.2 Restrictions

Currently the API is restricted to reading or submitting libraries of no more than 1,000 molecules. Submissions are restricted to molecules with less than 200 atoms. For best results organic molecules of 30-75 heavy atoms are recommended (think fragment libraries).

2.2 API Calls

2.2.1 API-Submit

API-Submit submits a set of molecules for computation. It expects six parameters that control access, what is submitted, how it is submitted and what data to compute. To make an API-Submit request of default routine data, one needs to send an HTTP request as follows:

- * URL: <https://api.chemalive.com>
- * Method: POST
- * Headers: Content-Type: application/json-rpc

* Body:

```
{
  "id": "0",
  "method": "submit",
  "params": {
    "apiKey": "APIKEY",
    "substances": ["SMILES"],
    "interactive": false,
    "force": false,
    "allTautomers": false,
    "allConformers": false
  },
  "jsonrpc": "2.0"
}
```

The APIKEY is a required parameter and will be provided upon request by contacting info@chemalive.com. In your e-mail you may already provide your own API key by generating it here: <https://randomkeygen.com/>

Which molecules are calculated is determined by the required "substances" parameter as a sequenced list of SMILES. Note that "substances" is plural (with an s) as it expects a list of SMILES using the JSON list format with '[']. The list limit is 1,000 molecules.

There are two optional parameters useful for testing the API, The "interactive" mode is set by a Boolean that turns on (true) or off (false) the submission of molecules to the front of the queue (DO NOT ABUSE: This is only for testing purposes when there is a lot of other user activity). Next, "force" is a second Boolean that forces the computation of a molecule (true) whether it is in the database already or not. This is also useful for testing. Both parameters default to 'false' if they are excluded from the query.

To control what type of data is calculated for the input molecules there are two parameters: "allTautomers" and "allConformers." Both are Booleans where turning off (false) and turning on (true) the passage of all tautomers or all conformers to the Quantum Engine is set. See section 1.2.1 for further description. Both parameters default to 'false' if they are excluded from the query.

Here is how to perform this operation via a common CURL tool from a terminal:

```
curl -d '{"id": "0", "method": "submit", "params": {"apiKey": "APIKEY", "substances": ["SMILES"], "interactive": false, "force": false, "allTautomers": false, "allConformers": false}, "jsonrpc": "2.0"}' -H "Content-Type: application/json-rpc" -X POST https://api.chemalive.com
```


This command is asking for a single PM6 calculation of the lowest UFF conformer of the molecule SMILES in its input tautomeric form.

The API response will be a block text of JSON formatted data indicating the status of the query molecule: exists, running or failed like this:

```
{"jsonrpc":"2.0","result":["running"],"id":"0"}
```

or,

```
{"jsonrpc":"2.0","result":["exists"],"id":"0"}
```

or,

```
{"jsonrpc":"2.0","result":["failed"],"id":"0"}
```

As API-Submit queries the status of a molecule, it is recommended that the submit routine be always the initial point of entry for querying a molecule. If the molecule does not yet exist in the database it will be calculated, if it does, API-Read can be invoked to read the data (see below). A list of existing molecules for testing is provided in Table 1 below.

Table 1. Example molecules already in Qontext for testing of API-Read.

Name	SMILES
Nitroglycerin	<chem>C(C(CO[N+](=O)[O-])O[N+](=O)[O-])O[N+](=O)[O-]</chem>
Aspirin	<chem>CC(=O)OC1=CC=CC=C1C(=O)O</chem>
Serotonin	<chem>C1=CC2=C(C=C1O)C(=CN2)CCN</chem>
Nicotine	<chem>CN1CCCC1C2=CN=CC=C2</chem>
Clofenotane (DDT)	<chem>C1=CC(=CC=C1C(C2=CC=C(C=C2)Cl)C(Cl)(Cl)Cl)Cl</chem>
Camphor	<chem>CC1(C2CCC1(C(=O)C2)C)C</chem>
Glyphosphate	<chem>C(C(=O)O)NCP(=O)(O)O</chem>
Glycine	<chem>C(C(=O)O)N</chem>
Ascorbic Acid (Vitamin C)	<chem>C(C(C1C(=C(C(=O)O1)O)O)O)O</chem>
Folic Acid	<chem>C1=CC(=CC=C1C(=O)NC(CCC(=O)O)C(=O)O)NCC2=CN=C3C(=N2)C(=O)N=C(N3)N</chem>
Lenalidomide	<chem>C1CC(=O)NC(=O)C1N2CC3=C(C2=O)C=CC=C3N</chem>
Penicillin	<chem>CC1(C(N2C(S1)C(C2=O)NC(=O)COC3=CC=CC=C3)C(=O)O)C</chem>
Morphine	<chem>CN1CCC23C4C1CC5=C2C(=C(C=C5)O)OC3C(C=C4)O</chem>
Progesterin	<chem>CC1=CC2C(CCC3(C2CCC3(C(=O)C)OC(=O)C)C)C4(C1=CC(C4)OC(=O)CCC5CCCC5)C</chem>

To use a CURL command for multiple queries using part of the list above like this:

```
curl -d '{"id": "0", "method": "submit", "params": {"apiKey": "APIKEY",
"substances":["CC(=O)OC1=CC=CC=C1C(=O)O","C1=CC2=C(C=C1O)C(=CN2)CCN","CN1CCCC1
C2=CN=CC=C2","C1=CC(=CC=C1C(C2=CC=C(C=C2)Cl)C(Cl)(Cl)Cl)","CC1(C2CCC1(C(=O)C2
)C)C","C(C(=O)O)NCP(=O)(O)O"]', "interactive":false, "force": false, "allTautomers": false,
"allConformers": false}, "jsonrpc": "2.0"}' -H "Content-Type: application/json-rpc" -X POST
https://api.chemalive.com
```

The API response would be:

```
{"jsonrpc":"2.0","result":["exists","exists","exists","exists","exists","exists"],"id":"0"}
```

Default values are assumed when an optional parameter is missing.

2.2.2 API-Read

API-Read reads pre-existing information from the ChemAlive database, named **Qontext**. To make an API-Read request of default routine data, one needs to send an HTTP request as follows:

```
* URL: https://api.chemalive.com
* Method: POST
* Headers: Content-Type: application/json-rpc
* Body:
{
  "id":"0",
  "method":"read",
  "params":{
    "apiKey":"APIKEY",
    "substances":["SMILES"],
    "procedure": "UFF-PM6",
    "sdf": false
  },
  "jsonrpc":"2.0"
}
```

Where, again, the APIKEY will be provided upon request and where 'substances' is a list of molecules in SMILES format. The "procedure" value lets the user specify which procedure data is returned by JSON (see below). If this value is not included, then it will default to UFF-PM6. The current acceptable strings are:

"UFF-PM6" : This indicates PM6 data derived from UFF guess structures.

"UFF" : This indicates only the UFF data.

Here is how to perform this operation via a common CURL tool on Linux:

```
curl -d '{"id": "0", "method": "read", "params": {"apiKey": "APIKEY", "substances": ["SMILES"]},
"procedure": "UFF-PM6", "sdf": false}, "jsonrpc": "2.0"}' -H "Content-Type: application/json-rpc" -X
POST https://api.chemalive.com
```

Do not begin queries to ConStruQt-API with a call to API-Read unless the molecules are known to exist, e.g. they are pre-computed or sample molecules (as in Table 1). To check a molecule's existence in the database use API-Submit (above).

The API-Read response will be a block text of JSON formatted data (discussed below). Post-processing of API-read JSON data can put the structural data in any desired format. Lookout for future API calls with support for other common formats. In the meantime, for SDF format choose 'true' for the "sdf" parameter, which returns the JSON block text but also a block text of the SD format of each computed structure for more facile reading of structural data into common web-based tools (e.g. JSMOL). If left out, sdf will default to 'false.'

2.3 Interpreting the JSON

The API block text that is returned is in JSON format. The structure of this data and definitions of terms are provided in this section.

2.3.1 Structure of the JSON Output

What follows is a detailed description of the JSON output for a set of queries for Uracil. From a terminal, the following curl command is inputted:

```
curl -d '{"id": "0", "method": "read", "params": {"apiKey": "APIKEY", "substances":
["C1=CNC(=O)NC1=O"], "procedure": "UFF-PM6", "sdf": false}, "jsonrpc": "2.0"}' -H "Content-
Type: application/json-rpc" -X POST https://api.chemalive.com
```

The JSON response has been formatted from its block form using standard JSON tools as below. The response delivers the coordinates of the lowest energy UFF conformer of the input tautomer:

```
[
{
  "charge": 0,
  "comment": "",
  "id": 34260220821,
  "inchi": "InChI=1S/C4H4N2O2/c7-3-1-2-5-4(8)6-3/h1-
2H,(H2,5,6,7,8)",
  "molecularFormula": "C4H4N2O2",
  "nAtoms": 12,
  "nHeavyAtoms": 8,
  "nRadicals": 0,
  "tautomers": [
    {
      "achiralSmiles": "Oc1nccc(=O)[nH]1",
      "id": 557162239394,
      "molecules": [
        {
          "chiralSmiles": "Oc1nccc(=O)[nH]1",
          "conformers": {
            "PM6": [
              {
                "atoms": [
                  {
                    "element": "O",
                    "partialCharge": -0.439611,
                    "valence": 0,
                    "x": 2.3973,
                    "y": -0.4596,
                    "z": -0.6018
                  },
                ],
              }
            ]
          }
        }
      ]
    }
  ]
}
```

This Section contains basic molecular information

"charge" is the molecular total charge
"comment" is a free form text field
"id" is the UID for the substance from inchi
"inchi" is the InChI
"molecularFormula" is that
"nAtoms" is the number of total atoms
"nHeavyAtoms" is the number of heavy atoms (all except hydrogen)
"nRadicals" is the number of radical centers and thus the expected molecular spin.

Stereoisomers of the achiral SMILES are Molecules

"chiralSmiles" may be the same as achiralSmiles
"conformers" section containing the structures for this chiralSmiles
"PM6" is the procedure name indicating the level of theory used to produce the structure and energy.

Atomic properties and coordinates

"atoms" section containing the coordinates for each conformer.
"partialCharge" is the atomic partial charge

```
{
  "element": "C",
  "partialCharge": -0.47886,
  "valence": 0,
  "x": 1.0963,
  "y": -0.319,
  "z": -0.245
},
{
  "element": "N",
  "partialCharge": 0.563538,
  "valence": 0,
  "x": 0.4098,
  "y": -1.3619,
  "z": 0.2535
},
{
  "element": "C",
  "partialCharge": 0.23248,
  "valence": 0,
  "x": -0.9184,
  "y": -1.1198,
  "z": 0.5904
},
{
  "element": "C",
  "partialCharge": -0.547212,
  "valence": 0,
  "x": -1.514,
  "y": 0.1012,
  "z": 0.4326
},
{
  "element": "C",
  "partialCharge": 0.663561,
  "valence": 0,
  "x": -0.806,
  "y": 1.2438,
  "z": -0.1002
},
{
  "element": "O",
  "partialCharge": -0.536305,
  "valence": 0,
  "x": -1.166,
  "y": 2.378,
  "z": -0.3056
},
{
  "element": "N",
  "partialCharge": -0.517108,
  "valence": 0,
  "x": 0.597,
  "y": 0.957,
  "z": -0.4474
},
{
  "element": "H",
  "partialCharge": 0.365833,
  "valence": 0,
  "x": 2.7202,
  "y": -1.4054,
  "z": -0.4374
},
{
  "element": "H",
  "partialCharge": 0.168088,
  "valence": 0,
  "x": -1.4468,
  "y": -1.9942,
  "z": 0.994
},
{
  "element": "H",
  "partialCharge": 0.218175,
  "valence": 0,
```

```

"x": -2.5561,
"y": 0.279,
"z": 0.7003
},
{
"element": "H",
"partialCharge": 0.307421,
"valence": 0,
"x": 1.1869,
"y": 1.701,
"z": -0.8335
}
],
"bonds": [
{
"firstAtom": 1,
"secondAtom": 2,
"stereo": 0,
"type": 1
},
{
"firstAtom": 1,
"secondAtom": 9,
"stereo": 0,
"type": 1
},
{
"firstAtom": 2,
"secondAtom": 3,
"stereo": 0,
"type": 2
},
{
"firstAtom": 2,
"secondAtom": 8,
"stereo": 0,
"type": 1
},
{
"firstAtom": 3,
"secondAtom": 4,
"stereo": 0,
"type": 1
},
{
"firstAtom": 4,
"secondAtom": 5,
"stereo": 0,
"type": 2
},
{
"firstAtom": 4,
"secondAtom": 10,
"stereo": 0,
"type": 1
},
{
"firstAtom": 5,
"secondAtom": 6,
"stereo": 0,
"type": 1
},
{
"firstAtom": 5,
"secondAtom": 11,
"stereo": 0,
"type": 1
},
{
"firstAtom": 6,
"secondAtom": 7,
"stereo": 0,
"type": 2
},
{
"firstAtom": 6,

```

Bond properties

"bonds" section containing the bond matrix of the molecule as determined by a bond sitance library.
"type" 1 = single, 2 = double, 3 = triple

```

      "secondAtom": 8,
      "stereo": 0,
      "type": 1
    },
    {
      "firstAtom": 8,
      "secondAtom": 12,
      "stereo": 0,
      "type": 1
    }
  ],
  "chiral": false,
  "cpuTime": 1742671,
  "energy": -34273.362916165606,
  "enthalpy": 53.4443608491264,
  "gibbsEnergy": 32.2953356389875,
  "id": 532775367176,
  "molecularDipole": [
    -1.681254,
    2.209447,
    -0.10197
  ],
  "sourceEnergy": -34255.51865697295,
  "zeroPointEnergy": 49.079736424908305
}
]
};
"id": 954644588460
}
],
"nRotationBonds": 1
}
],
"weight": 112
}
]

```

Further information and energies

“chiral” tells us if the molecule is chiral or not.
 “cpuTime” how much cpu time was consumed for the whole query
 “energy” the electronic energy (SCF energy) resulting from the optimization of the conformer in kcal/mol.
 “enthalpy” the energetic correction in kcal/mol derived from the harmonic frequency analysis to compensate for enthalpy. Add this value to “energy” to get the enthalpy of the conformer.
 “gibbsEnergy” the energetic correction in kcal/mol derived from the harmonic frequency analysis to compensate for entropy at 298.15 K. Add this value to “energy” to get the Gibbs energy of the conformer.
 “id” is the unique id for this conformer.
 “molecularDipole” x,y,z vector of molecular dipole in wit magnitude in Debye.
 “sourceEnergy” a single point PM6 energy in kcal/mol on the input coordinates from UFF assuming default routine.
 “zeroPointEnergy” the energetic correction in kcal/mol derived from the harmonic frequency analysis to compensate for thermal vibrations at zero Kelvin. Add this value to the value “energy” to get the ZPE corrected energy of the conformer.

Notice the response is for a single tautomer (input tautomer) and one conformer. The input SMILES is not in canonical form. The value in the JSON has been converted to canonical form. To query all the tautomers of Uracil we input the CURL as below with the “allTautomers” parameter set to ‘true.’

```

curl -d '{"id": "0", "method": "read", "params": {"apiKey": "APIKEY", "substances": ["C1=CNC(=O)NC1=O"], "procedure": "UFF-PM6", "sdf": false, "allTautomers": true}, "jsonrpc": "2.0"}' -H "Content-Type: application/json-rpc" -X POST https://api.chemalive.com

```

The “atoms” and “bonds” information in the JSON output below has been removed for clarity and length.

```

[
  {
    "charge": 0,
    "comment": "",
    "id": 34260220821,
    "inchi": "InChI=1S/C4H4N2O2/c7-3-1-2-5-4(8)6-3/h1-2H,(H2,5,6,7,8)",
    "molecularFormula": "C4H4N2O2",
    "nAtoms": 12,
    "nHeavyAtoms": 8,
    "nRadicals": 0,
    "tautomers": [
      {
        "achiralSmiles": "Oc1nc(=O)cc[nH]1",
        "id": 479429729245,
        "molecules": [
          {
            "chiralSmiles": "Oc1nc(=O)cc[nH]1",
            "conformers": {
              "PM6": [
                {
                  "atoms": [REMOVED for CLARITY]
                  "bonds": [REMOVED for CLARITY]
                }
              ]
            }
          }
        ]
      }
    ]
  }
]

```

First Tautomer

```

    "chiral": false,
    "cpuTime": 1718278,
    "energy": -34252.87000366132,
    "enthalpy": 52.9681485176079,
    "gibbsEnergy": 30.2962447583472,
    "id": 103026086028,
    "molecularDipole": [
      -4.588171,
      -7.594547,
      -2.016014
    ],
    "sourceEnergy": -34233.47876180404,
    "zeroPointEnergy": 48.0667628996721
  }
}
},
{id": 363743268117
}
],
"nRotationBonds": 1
},
{
  "achiralSmiles": "Oc1cc[nH]c(=O)n1",
  "id": 547299972436,
  "molecules": [
    {
      "chiralSmiles": "Oc1cc[nH]c(=O)n1",
      "conformers": {
        "PM6": [
          {
            "atoms": [REMOVED for CLARITY]
            "bonds": [REMOVED for CLARITY]
            "chiral": false,
            "cpuTime": 1717644,
            "energy": -34272.22751418219,
            "enthalpy": 53.6198409753339,
            "gibbsEnergy": 32.5265127172239,
            "id": 542322670579,
            "molecularDipole": [
              0.003117,
              -4.597602,
              1.114817
            ],
            "sourceEnergy": -34256.62138876162,
            "zeroPointEnergy": 49.2880963107204
          }
        ]
      }
    }
  ],
  "id": 1028242032074
}
],
"nRotationBonds": 1
},
{
  "achiralSmiles": "Oc1ccnc(n1)O",
  "id": 885488039700,
  "molecules": [
    {
      "chiralSmiles": "Oc1ccnc(n1)O",
      "conformers": {
        "PM6": [
          {
            "atoms": [REMOVED for CLARITY]
            "bonds": [REMOVED for CLARITY]
            "chiral": false,
            "cpuTime": 1723164,
            "energy": -34261.91838355477,
            "enthalpy": 52.6111054941543,
            "gibbsEnergy": 31.557396386910003,
            "id": 576213982852,
            "molecularDipole": [
              1.844828,
              1.361369,
              -0.561827
            ],
            "sourceEnergy": -34250.223569940346,

```

Second Tautomer

Third Tautomer

```

    "zeroPointEnergy": 48.2681581894407
  }
]
},
"nRotationBonds": 1
},
{
"achiralSmiles": "O=c1cc[nH]c(=O)[nH]1",
"id": 164913420633,
"molecules": [
{
"chiralSmiles": "O=c1cc[nH]c(=O)[nH]1",
"conformers": {
"PM6": [
{
"atoms": [REMOVED for CLARITY]
"bonds": [REMOVED for CLARITY]
"chiral": false,
"cpuTime": 1700412,
"energy": -34286.918522335356,
"enthalpy": 54.4892178781446,
"gibbsEnergy": 33.2334119987322,
"id": 367117424913,
"molecularDipole": [
3.915668,
2.3918,
-0.136423
],
"sourceEnergy": -34269.38758887185,
"zeroPointEnergy": 50.1084576737175
}
]
},
"nRotationBonds": 1
},
{
"achiralSmiles": "O=c1ccc[nH]c(=O)[nH]1",
"id": 557162239394,
"molecules": [
{
"chiralSmiles": "O=c1ccc[nH]c(=O)[nH]1",
"conformers": {
"PM6": [
{
"atoms": [REMOVED for CLARITY]
"bonds": [REMOVED for CLARITY]
"chiral": false,
"cpuTime": 1742671,
"energy": -34273.362916165606,
"enthalpy": 53.4443608491264,
"gibbsEnergy": 32.2953356389875,
"id": 532775367176,
"molecularDipole": [
-1.681254,
2.209447,
-0.10197
],
"sourceEnergy": -34255.51865697295,
"zeroPointEnergy": 49.079736424908305
}
]
},
"nRotationBonds": 1
},
{
"achiralSmiles": "O=c1ccnc(=O)[nH]1",
"id": 536621173303,

```

Fourth Tautomer

Fifth Tautomer


```

"molecules": [
  {
    "chiralSmiles": "Oc1ccnc(=O)[nH]1",
    "conformers": {
      "PM6": [
        {
          "atoms": [REMOVED for CLARITY]
          "bonds": [REMOVED for CLARITY]
          "chiral": false,
          "cpuTime": 1743631,
          "energy": -34267.16262730918,
          "enthalpy": 53.0716504763826,
          "gibbsEnergy": 31.7182717879467,
          "id": 979813169178,
          "molecularDipole": [
            -5.120868,
            0.885976,
            -0.105637
          ],
          "sourceEnergy": -34251.877839644185,
          "zeroPointEnergy": 48.5968884563697
        }
      ]
    },
    "id": 644742955358
  },
  {
    "nRotationBonds": 1
  },
  {
    "weight": 112
  }
]

```

Sixth Tautomer

Each tautomer is listed with its lowest energy conformer at PM6 level and corresponding energies. Setting “allConformers” parameter to ‘true’ will list all the conformers of the input tautomer via the following CURL.

```

curl -d '{"id": "0", "method": "read", "params": {"apiKey": "APIKEY", "substances":
["C1=CNC(=O)NC1=O"], "procedure": "UFF-PM6", "sdf": false, "allConformers": true}, "jsonrpc":
"2.0"}' -H "Content-Type: application/json-rpc" -X POST https://api.chemalive.com

```

```

[
  {
    "charge": 0,
    "comment": "",
    "id": 756805887804,
    "inchi": "InChI=1S/C4H4N2O2/c7-3-1-2-5-4(8)6-3/h1-2H,(H2,5,6,7,8)",
    "molecularFormula": "C4H4N2O2",
    "nAtoms": 12,
    "nHeavyAtoms": 8,
    "nRadicals": 0,
    "tautomers": [
      {
        "achiralSmiles": "Oc1nccc(=O)[nH]1",
        "id": 783433348043,
        "molecules": [
          {
            "chiralSmiles": "Oc1nccc(=O)[nH]1",
            "conformers": {
              "PM6": [
                {
                  "atoms": [REMOVED for CLARITY]
                  "bonds": [REMOVED for CLARITY]
                  "chiral": false,
                  "cpuTime": 1739850,
                  "energy": -34273.362916165606,
                  "enthalpy": 53.4443608491264,
                  "gibbsEnergy": 32.2953356389875,
                  "id": 385015450777,
                  "molecularDipole": [
                    -1.681254,
                    2.209447,

```

First Conformer

```

-0.10197
],
"sourceEnergy": -34255.51865697295,
"zeroPointEnergy": 49.079736424908305
},
{
"atoms": [REMOVED for CLARITY]
"bonds": [REMOVED for CLARITY]
"chiral": false,
"cpuTime": 1686186,
"energy": -34273.362916165606,
"enthalpy": 53.4443608491264,
"gibbsEnergy": 32.2953356389875,
"id": 42236847811,
"molecularDipole": [
-1.681254,
2.209447,
-0.10197
],
"sourceEnergy": -34255.51865697295,
"zeroPointEnergy": 49.079736424908305
},
{
"chiral": false,
"cpuTime": 1800012,
"energy": -34266.27002524961,
"enthalpy": 53.143173530533794,
"gibbsEnergy": 31.7066128490754,
"id": 395780616356,
"molecularDipole": [
-2.011103,
-0.313505,
0.102397
],
"sourceEnergy": -34248.87469644967,
"zeroPointEnergy": 48.6334227649179
},
{
"atoms": [REMOVED for CLARITY]
"bonds": [REMOVED for CLARITY]
"chiral": false,
"cpuTime": 1730751,
"energy": -34266.27002524961,
"enthalpy": 53.143173530533794,
"gibbsEnergy": 31.7066128490754,
"id": 811533851233,
"molecularDipole": [
-2.011103,
-0.313505,
0.102397
],
"sourceEnergy": -34248.87469644967,
"zeroPointEnergy": 48.6334227649179
}
]
},
"nRotationBonds": 0
},
"weight": 112
}
]

```

Second Conformer

Third Conformer

Three conformers are listed. If allTautomers and allConformers are both set to 'true' the JSON output will include the super set. If a molecule with chiral ambiguity is inputted, then the stereoisomers will be generated. Here is a random example to try:

```
curl -d '{"id": "0", "method": "read", "params": {"apiKey": "APIKEY", "substances": ["CCCC(C)CC"], "procedure": "UFF-PM6", "sdf": false, "allConformers": true}, "jsonrpc": "2.0"}' -H "Content-Type: application/json-rpc" -X POST https://api.chemalive.com
```

The JSON output now contains two SMILES, CCC[C@@H](CC)C and CCC[C@H](CC)C. If either of these is used instead for the CURL then only that stereoisomer will be returned.

2.3.2 Analysis Example: Uracil

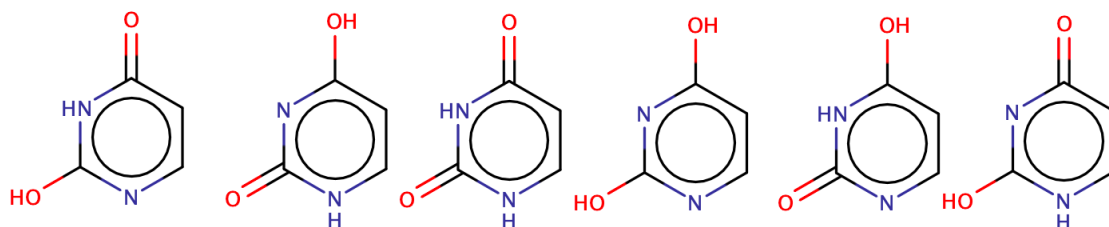
So, what do we do with all this data? **ConStruQt-API** is the only game in town for automated accurate energetic assessment of tautomers and conformers on the web. We will analyze Uracil.

2.3.2.1 Uracil Tautomers

In the example above, Uracil produced 6 “chiralSmiles” listed now in Table 2, below.

Table 2. Tautomeric data extracted from the JSON

chiralSmiles	energy	gibbsEnergy	energy + gibbsEnergy = Free-energy	Relative Free- Energy
<chem>Oc1nccc(=O)[nH]1</chem>	-34273.36296	32.29779069	-34241.06517	12.61
<chem>Oc1cc[nH]c(=O)n1</chem>	-34272.22744	32.5279814	-34239.69946	13.98
<chem>O=c1cc[nH]c(=O)[nH]1</chem>	-34286.91865	33.23979291	-34253.67886	0.00
<chem>Oc1ccnc(n1)O</chem>	-34261.91822	31.56413815	-34230.35408	23.32
<chem>Oc1ccnc(=O)[nH]1</chem>	-34269.14405	31.71863757	-34237.42541	16.25
<chem>Oc1nc(=O)cc[nH]1</chem>	-34261.74206	31.90388754	-34229.83818	23.84



Note that Uracil does not have any chiral centers and thus, there are also 6 achiralSmiles that are identical. To extract all unique structures enumerated we look for chiralSmiles (irrespective of the existence of chirality). Each of these unique structures has undergone full conformationally analysis with the lowest structure according to UFF used to pass on to the PM6 stage. The “energy” and gibbsEnergy” values listed above are taken for the conformer with a more negative (more stable) energy value when more than one exists. To obtain the free-energy, which is corrected for entropy effects, we simply add the “energy” to “gibbsEnergy” as in column 4. To make more sense of the data we normalize against the most negative resulting value to get the Relative Free-energy in column 5. The zero value is the most stable tautomer, O=c1cc[nH]c(=O)[nH]1, according to quantum mechanical gas-phase estimates using the semi-empirical PM6 method taking into account all enumerated tautomers and conformational structures. The most stable tautomer is the structure most likely to exist in a practical setting. From the magnitude of the difference we can conclude that this tautomer would represent nearly 100% of the molecules. The large kcal/mol differences (next structure is 12.61 kcal/mol higher) are well above the kT value at room temperature and thus these other structures essentially do

not exist under the model conditions. Of course, this could change in a solvent, or when embedded in a protein. Such assessments will be added to **ConStruQt** later on.

2.3.2.2 Uracil Conformers

Now let's take a look at the conformations. The first tautomer in Table 1 is Oc1nccc(=O)[nH]1. It has two conformations shown in Figure 1.

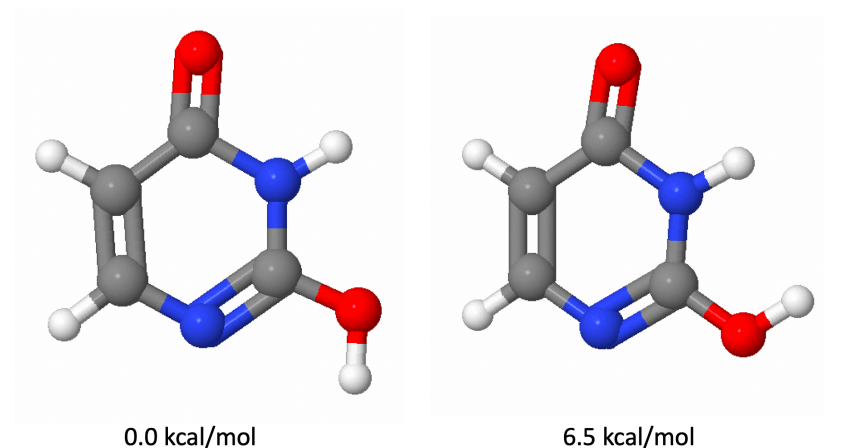


Figure 1. Relative Free-energies of the two conformers of the first tautomer of Uracil.

As uracil is conformationally simple, there are only two reasonable conformers for this tautomer, OH 'down' (left, 0.0 kcal/mol) and OH 'up' (right, 6.5 kcal/mol). OH 'up' is significantly less stable than OH 'down' mostly due to the 1,3-H---H unfavorable interaction versus the 1,3-H---N: favorable interaction. Now you know. Also, now you see why conformational analysis can be important. If we had chosen the OH 'up' conformer to represent this tautomer, we would have predicted that this tautomer was the fourth highest in energy in the set instead of the second. (12.61 + 6.5).

3 Terms of Service

3.1 Access to Our APIs

Subject to the restrictions below, we grant you a non-exclusive, worldwide, non-transferable (subject to the section titled "Assignment"), limited license to access our APIs and documentation only as necessary to develop, test and support an integration of your application (an "**Application**" or "**App**") with the Services. You may charge for your Application; however, you may not sell, rent, lease, sublicense, redistribute, or syndicate access to any of our APIs.

3.2 Rules

Your license to access our APIs and documentation is limited and subject to compliance with the Brand Guidelines above in section 2.1 where use of our APIs should refer to and link to our company website. Further, you will not: (A) access our APIs or documentation in violation of any law or regulation; (B) access our APIs in any manner that (i) compromises, breaks or circumvents any of our technical processes or security measures associated with the Services, (ii) poses a security vulnerability to customers or users of the Services, or (iii) tests the vulnerability of our systems or networks; (C) access our APIs or documentation in order to replicate or compete with the Services; (D) attempt to reverse engineer or otherwise derive

source code, trade secrets, or know-how of our APIs or Services; or (E) attempt to use our APIs in a manner that exceeds rate limits, or constitutes excessive or abusive usage.

3.3 Transparency and Reporting

If you offer your Application for use by others outside your organization, you must maintain a user agreement and privacy policy for your Application, which is prominently identified or located where users download or access your Application. Your privacy policy must meet applicable legal standards and describe the collection, use, storage and sharing of data in clear, understandable and accurate terms. You must promptly notify us in writing via email to info@chemalive.com of any breaches of your user agreement or privacy policy that impact or may impact customers or users of the Services.

3.4 Privacy Policy

The **ConStruQt**-API connects to a single tenancy database fully owned and operated by ChemAlive. Molecules submitted by a user may be linked to that user to allow easy navigation of passed submissions, but this linkage is not publicly available. ChemAlive will collect and use data relating to the use and performance of our API and will store all chemical data. We strongly suggest that no user submit intellectual property sensitive molecules to our API as their structure will be integrated within our database. If the submission of sensitive molecules should become necessary, please contact info@chemalive.com and we will arrange for a private database and engage in a side contract. WE WAIVE ALL RESPONSIBILITY FOR BREACHES OF INTELLECTUAL PROPERTY SECURITY that may occur from the use of our API – consider it as connected to an open database.

3.5 Right to Suspend Access and Audit

If we believe that there is a violation of the Contract that can simply be remedied by your modification or update of your Application, we will, in most cases, ask you to take direct action rather than intervene. In such instance, we may use your name, address and other contact details to contact you or provide this contact information to any third party that reasonably, in ChemAlive's sole determination, claims that you do not possess all of the necessary intellectual property rights. In some instances, we may directly step in and take what we determine to be appropriate action if you are not responsive, or if we believe there is a credible risk of harm to us, the Services, our customers or users or any third parties. ChemAlive also reserves a right to audit your application to ensure it does not violate our terms and policies. You agree that you will cooperate with inquiries related to such an audit and provide us with proof that your application complies with our terms and policies.

3.6 Ownership and Property Rights

You retain your ownership rights in your Application and we own and will continue to own our APIs, documentation and Services, including all related intellectual property rights therein. All of our rights not expressly granted by the Contract are hereby retained.

The more suggestions made, the better our APIs become. If you send us any feedback or suggestions regarding the APIs or documentation, there is a chance we will use it, so you grant us an unlimited, irrevocable, perpetual, sublicensable, transferable, royalty-free license to use any such feedback or suggestions for any purpose without any obligation or compensation to you. If we choose not to implement the suggestion, please don't take it personally. We appreciate it nonetheless.

3.7 Termination

You may terminate the Contract by discontinuing use of our APIs. We may terminate the Contract with or without cause, and without notice to you. Upon termination of the Contract, all rights and licenses granted to you will terminate immediately. You understand that any APIs or documentation that are not made generally available but that are otherwise made available to you are the confidential information of ChemAlive. Upon termination of the Contract, you will promptly destroy copies of any documentation and any other ChemAlive information in your possession or control that was received under the Contract.

3.8 Representations; Disclaimer of Warranties

You represent and warrant that you have validly entered into the Contract and have the legal power to do so.

EXCEPT AS EXPRESSLY PROVIDED FOR HEREIN, THE APIS, DOCUMENTATION AND ALL RELATED COMPONENTS AND INFORMATION ARE PROVIDED BY US ON AN "AS IS" AND "AS AVAILABLE" BASIS WITHOUT ANY WARRANTIES OF ANY KIND, AND WE EXPRESSLY DISCLAIM ANY AND ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. YOU ACKNOWLEDGE THAT WE DO NOT WARRANT THAT THE APIS WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE.

3.9 Limitations of Liability

IN NO EVENT WILL OUR LIABILITY ARISING OUT OF OR RELATED TO THE CONTRACT (WHETHER IN CONTRACT OR TORT OR UNDER ANY OTHER THEORY OF LIABILITY) EXCEED CHF 100.

IN NO EVENT WILL WE HAVE ANY LIABILITY TO YOU OR TO ANY THIRD PARTY FOR ANY LOST PROFITS OR REVENUES OR FOR ANY INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, COVER OR PUNITIVE DAMAGES HOWEVER CAUSED, WHETHER IN CONTRACT, TORT OR UNDER ANY OTHER THEORY OF LIABILITY, AND WHETHER OR NOT YOU OR THE THIRD PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING DISCLAIMER WILL NOT APPLY TO THE EXTENT PROHIBITED BY APPLICABLE LAW.

The limitations under this "Limitation of Liability" section apply with respect to all legal theories, whether in contract, tort or otherwise, and to the extent permitted by law. The provisions of this "Limitation of Liability" section allocate the risks under the Contract between the parties, and the parties have relied on these limitations in determining whether to enter into the Contract.

3.10 Application of Consumer Law

Our APIs and documentation are intended for use by businesses and organizations and not for consumer purposes. To the maximum extent permitted by law, you hereby acknowledge and agree that consumer laws do not apply. If however any consumer laws (e.g., in Australia, the Competition and Consumer Act 2010 (Cth)) do apply and cannot otherwise be lawfully excluded, nothing in these API Terms will restrict, exclude or modify any statutory warranties, guarantees, rights or remedies you have, and our liability is limited (at our option) to the replacement or repair of the APIs.

3.11 Your Indemnification of Us

You will defend us from and against any and all third party claims, actions, suits, proceedings, and demands arising from or related to your violation of the Contract or your violation of your user agreement or privacy policy (a "**Claim Against Us**"), and will indemnify the ChemAlive Indemnified Parties for all reasonable attorney's fees incurred and damages and other costs finally awarded against a ChemAlive Indemnified Party in connection with or as a result of, and for amounts paid by a ChemAlive Indemnified Party under a settlement you approve of in connection with, a Claim Against Us. We must provide you with prompt written notice of any Claim Against Us and allow you the right to assume the exclusive defense and control, and cooperate with any reasonable requests assisting your defense and settlement of such matter. This section states your sole liability with respect to, and the ChemAlive Indemnified Parties' exclusive remedy against you for, any Claim Against Us.

3.12 Limitations on Indemnifications

Notwithstanding anything contained in the preceding section, (a) we will always be free to choose our own counsel if we pay for the cost of such counsel; and (b) no settlement may be entered into by you, without our express written consent (such consent not to be unreasonably withheld), if: (i) the third party asserting the claim is a government agency, (ii) the settlement arguably involves the making of admissions, (iii) the settlement does not include a full release of liability, or (iv) the settlement includes terms other than a full release of liability and the payment of money.

3.13 Survival

The sections titled "Our Rights to Suspend Access and Audit," "Ownership & Proprietary Rights," "Termination," "Representations; Disclaimer of Warranties," "Limitation of Liability," "Your Indemnification of Us," "Limitations on Indemnifications," and "Survival," as well as all of the provisions under the general heading "General Provisions," will survive any termination or expiration of the Contract.

3.14 General Provisions

- (1) You grant us the right to use your company name and logo as a reference for marketing or promotional purposes on our website and in other public or private communications with our existing or potential customers, subject to your standard trademark usage guidelines as provided to us from time-to-time.
- (2) Neither we nor you will be liable by reason of any failure or delay in the performance of its obligations on account of events beyond the reasonable control of a party, which may include denial-of-service attacks, a failure by a third party hosting provider or utility provider, strikes, shortages, riots, fires, acts of God, war, terrorism, and governmental action.
- (3) The parties are independent contractors. The Contract does not create a partnership, franchise, joint venture, agency, fiduciary or employment relationship between the parties. There are no third party beneficiaries to the Contract.
- (4) Except as otherwise set forth herein, all notices under the Contract will be by email. Notices will be deemed to have been duly given the day after they are sent.
- (5) ChemAlive is still evolving, and so we need the flexibility to occasionally make changes to our APIs, including backwards incompatible changes. We will try to give notice of these changes through email and updates to this document. Also, parts of our API are undocumented, including certain methods, events, and properties. Given that these undocumented aspects of our APIs may change at any time, you should not rely on their behaviors.

3.15 Modification to the Contract

As our business evolves, we may change these API Terms and the other components of the Contract. If we make a material change to the Contract, we will provide you with reasonable notice prior to the change taking effect, either by emailing the email address associated with your account or by messaging you through the Services. You can review the most current version of the API Terms at any time by downloading this document and by visiting the most current versions of the other pages that are referenced in the Contract. The materially revised Contract will become effective on the date set forth in our notice, and all other changes will become effective upon posting of the change. If you access our APIs after the effective date, that access will constitute your acceptance of any revised terms and conditions.

3.16 Waiver

No failure or delay by either party in exercising any right under the Contract will constitute a waiver of that right. No waiver under the Contract will be effective unless made in writing and signed by an authorized representative of the party being deemed to have granted the waiver.

3.17 Severability

The Contract will be enforced to the fullest extent permitted under applicable law. If any provision of the Contract is held by a court of competent jurisdiction to be contrary to law, the provision will be modified by the court and interpreted so as best to accomplish the objectives of the original provision to the fullest extent permitted by law, and the remaining provisions of the Contract will remain in effect.

3.18 Assignment

Neither party may assign or delegate any of its rights or obligations hereunder, whether by operation of law or otherwise, without the prior written consent of the other party (not to be unreasonably withheld). Notwithstanding the foregoing, either party may assign the Contract in its entirety, without consent of the other party, to a corporate affiliate or in connection with a merger, acquisition, corporate reorganization, or sale of all or substantially all of its assets. Any purported assignment in violation of this section is void. A party's sole remedy for any purported assignment by the other party in breach of this section will be, at the non-assigning party's election, termination of the Contract upon written notice to the assigning party. Subject to the foregoing, the Contract will bind and inure to the benefit of the parties, their respective successors and permitted assigns.

3.19 Entire Agreement

The Contract, including these API Terms constitutes the entire agreement between the parties and supersedes all prior and contemporaneous agreements, proposals or representations, written or oral, concerning its subject matter.

4 Appendix

4.1 Web Links

Example JSON Outputs for the eight end states can be found for download here:

<https://www.chemalive.com/construqt-api-docs/>