The quantum chemical search for novel materials and the issue of data processing: The InfoMol project

Lüthi, Hans P ; Heinen, Stefan ; Schneider, Gisbert ; Glöss, Andreas ; Brändle, Martin P ; King, Rollin A ; Pyzer-Knapp, Edward ; Alharbi, Fahhad H ; Kais, Sabre

DOI: https://doi.org/10.1016/j.jocs.2015.10.003

The Quantum Chemical Search for Novel Materials and the Issue of Data Processing: The InfoMol Project

Hans P. Lüthi[1,2], Stefan Heinen[1], Gisbert Schneider[1], Andreas Glöss[3], Martin P. Brändle[4], Rollin A. King[5], Edward Pyzer-Knapp[6], Fahhad H. Alharbi[2,7], and Sabre Kais[2,7]

[1]Department of Chemistry and Applied Bioscience, ETH Zürich,Zürich, Switzerland, [2]Qatar Environment and Energy Research Institute, Doha, Qatar [3]Institute of Chemistry, University of Zürich, Zürich, Switzerland, [4]Zentrale Informatik, University of Zürich, Zürich, Switzerland, [5]Bethel University, St. Paul, Minnesota, USA, [6] Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts, USA, [7]Hamad Bin Khalifa University, Doha, Qatar

Revised Version, Sept 20, 2015

Abstract:
In the search for novel materials, quantum chemical modelling and simulation has taken an important role. Molecular properties are computed on the basis of first-principles methods and screened against pre-defined criteria. Alternatively, the results of these computations are used as source data to enhance the predictions of data-centric models. Whichever modelling strategy is being applied, data-intense steps are involved in the process. One key bottleneck in this regard is the lack of availability of machine-readable output for virtually all quantum chemistry codes. The results of computations need to be extracted manually or using scripts and parsers, instead of directly being written out in machine-readable format to be imported into a database for archival, analysis and exchange. We present two solutions implemented in two selected examples, the TURBOMOLE and PSI4 program packages. Next to the standard output, both codes generate Extensible Markup Language (XML) output files, but in two different ways. The generation of machine-readable output in a structured format can easily be implemented, and, as long as the data can be transformed, the choice of data format is secondary. The concept is illustrated for two different use cases from method benchmarking and drug design. A third illustration addresses the definition of a data processing and exchange protocol for screening libraries of light-harvesting compounds.

Keywords: quantum chemistry, modeling and simulation, materials design, drug design, data processing

1. Introduction

Quantum chemical modelling and simulation have played an important role in guiding the search ("discovery") for novel materials. Typically, we explore how a successful material ("lead") works, and based on the knowledge gained from this analysis, we propose new compounds ("prospects") and score these against pre-defined criteria ("virtual screening"). More recently, machine learning and other data-centric methods were introduced to expedite the search process. Whichever strategy is applied, data-intense steps are involved in the discovery process. However, in computational quantum chemistry, the processing (storage, analysis, and exchange) of large amounts of data is an issue that has not been thoroughly addressed until very recently.

Typically, the results of quantum chemical computations are analyzed and presented by means of visualization tools interfaced with a specific application code, and by collection of data which has either been extracted manually from the standard output ("cut-and-paste") or by use of scripts and parsers. This manual or semi-automatic transcription of the data is labor-intensive, and error-prone at the same time. Sooner rather than later, the application codes will have to deliver their results in a machine-readable format. Ideally, these data will be well structured to be imported into a database.

The choice of data structure and format presents a challenge. There are many choices, and numerous initiatives addressing the issue of machine-readable output formats for modelling in chemistry were observed. However, consensus has never been reached, and most examples of quantum chemical calculations delivering machine-readable output served a very narrow purpose such as the computation of the energies of conformers of large array of molecules. It is not until recently that more general data processing solutions were presented, which also showed some "new chemistry". The most striking example is the Harvard Clean Energy Project (CEP, http://cleanenergy.molecularspace.org/). As part of the CEP, about three million organic molecules were screened based on their quantum chemically computed properties to find prospects for highly efficient light-harvesting compounds to be used in photovoltaic devices [1].

In an effort to optimize the non-linear optical properties of donor-acceptor functionalized linearly pi-conjugated polymers, some of the present authors (HPL, MPB) developed a "computing infrastructure", which allowed the processing of the results of a few thousand compounds using statistical analysis to establish relationships between molecular properties [2]. The output of the quantum chemical code (Gaussian03, [3]) was transformed to Chemical Markup Language format (CML, [4]), an Extensible Markup Language "dialect" (XML, [5]), using a parser (JumboMarker, [6]). The CML output files were imported into a document database (Xindice, [7]). Later, the authors entered a developer agreement with the TURBOMOLE GmbH (http://turbomole.com) to extend their quantum chemistry code to generate XML output during execution time, obviating parsing of the standard output [8]. More recently, one of the authors (RAK) was also able to output the results of computations with the PSI4 code in an XML format during run-time (v.i).

The goal of these authors is to develop an infrastructure for the analysis of the information obtained from quantum chemical computations to support the search for novel materials. The results of these

computations will be archived in databases, to enable fast and efficient queries of the data. The focus of this project named "InfoMol" is not necessarily on virtual screening, but much more on the establishment of relationships between the (molecular) properties of these compounds and their chemical activity. From this, we expect to find the chemical information contained in the molecule which is responsible for a particular property. These "descriptors" will also enter the scoring functions to screen an array of prospects against a set of target properties. At the same time, the information obtained from these quantum chemical computations can serve as input for machine learning and other data-centric modelling tools.

The information archived needs to be general and easy to query. The queries should not only allow the rapid extraction of specific data, but should also support complex searches. Finding those molecules within a large array, which show maximal values for a certain target property, but also fulfill a number of other specifications, needs to be possible.

The main features of the InfoMol infrastructure, along with three illustrations, two of which are use cases, are presented in this paper. The third illustration addresses the issue of finding efficient scoring functions for very large arrays of compounds such as those available from the Harvard CEP project.

2. InfoMol Tools: TURBOMOLE / PSI4-XML-eXist

TURBOMOLE is a commercially available quantum chemistry application package to solve the Schrödinger equation for molecules in gas phase or in a polarizable continuum using wave function and density functional methods (www.turbomole.com). The code, like many other electronic structure theory packages, consists of different programs performing specific tasks ("calculations"). The programs are interconnected by shell scripts ("driver").

To obtain machine-readable output, in principle, each single write statement could be replicated, the original one creating standard output and the new one creating the machine-readable output. In TURBOMOLE (Version 6.3), however, an output layer was constructed which collects and converts the relevant information to XML format to be archived in a database (see Figure 1). For this conversion we are using FoXLib (FORTRAN to XML library; see http://www1.gly.bris.ac.uk/~walker/FoX/ or https://github.com/andreww/fox). This version of the TURBOMOLE program package is still under development and not yet part of the official distribution.

Figure 1: Workflow

PSI4 is a widely used open-source quantum chemical program package (www.psicode.org). Though built upon earlier versions of PSI, version 4 introduced a top-level python driver. The major

programs or modules in PSI4 are written in C++ and are called by the driver. The input file syntax is modified Python (dubbed “Psithon”), which allows the user to mingle standard Python with typical chemical parameters in the input file (see Figure 2). Most significant for the present paper, the underlying modules return key results back to the driver routine, resulting in several advantages. First, the driver can easily collect the desired information for subsequent, alternative output formats such as XML. Second, as new capabilities are added to the PSI4 program package, e.g., a new wave function type resulting in new energy values, no changes are necessary to the C++ source code to support its XML output. The data exported to XML can be a standard set of data, or specified explicitly by the user in the input file. Finally, the decision to include or exclude certain data in the XML output could even be made at runtime, if that were desirable. This version of PSI4 is still under development and there are no use cases available yet.

Figure 2: Psithon

In both cases, the machine-readable output is generated during execution of the programs. This does require interventions in the application program code (TURBOMOLE) or in the driver (PSI4), but comes with a number of advantages. (i) there is no parser required, (ii) there is no need to maintain the parser (response to changes in standard output format, and, most importantly, (iii) there is no dependence on standard output (availability of a particular data item and numerical precision of the information).

The data to be stored for the longer term (i.e. duration of the project or beyond) are those one would typically find in the final output of a calculation. These data can be split into three groups, namely metadata, input data, and results (“output data”). Based on the metadata (provenance of the computation, i.e. information such as hardware and operating system; application program version and libraries at compile time) and the input data, a calculation can be fully reproduced. The results cover energies and properties (structural and electronic). The archived data thus are light-weight (scalars, vectors, tensors, small matrices); still measured in Kbytes or Mbytes per calculation.

Table 1: XML-CML

If necessary, larger data items, such as the wave function or the electron density, can either be stored or be re-computed on the fly. If stored, for reasons of efficiency, this should be in binary format. Often these data are needed for different programs to interoperate (multi-scale modelling, for example) rather than for the analysis of the computed data. C5/D5Cost [9], developed for code interoperability, allows reading/writing information in an HDF5 format (www.hdfgroup.org) at high throughput, and has been used to connect different ab initio codes or ab initio codes with post-processing tools.

The Extensible Markup Language (XML) has the advantage that it is both human- and machine-readable, and that it is widely used as a data-exchange format supported by the World Wide Web consortium (http://www.w3.org/XML/). XML allows expressing content, structure and logic, and the schema can always be adapted to serve a particular application or purpose. The data model in XML databases is built on hierarchial documents archived in collections, different from relational

databases that are defined by tables. It offers a good balance between structure and flexibility. Flexibility is an issue, as not every single calculation will deliver the same set of data; the data to be archived will strongly depend on the kind of problem: the optimization of a molecular geometry or the response of this same molecule to an external field (electric or magnetic; static or time-dependent) will create very different sets of outputs. This balance of structure and flexibility has been harnessed by the NoSQL community to develop powerful, yet flexible database platforms such as MongoDB (www.mongodb.org), which is now also being used by the Harvard CEP. Note that XML can be easily adapted to be fed directly into a MongoDB.

XML also comes with a large number of tools for analysis, transformation and data archival. Chemical Markup Language (CML) is an established "XML dialect" for chemistry, and its schema covers most of the semantics needed, even for computational quantum chemistry [4]. More recently, some of those authors published CompChem, a CML-based convention for computational chemistry [10]. CompChem has a working CML and convention validator code. Unfortunately, as of 2015, CompChem has seen very limited adoption, has only tentative basis-set and array specifications, and a published extension of CML to the NWChem program [23] deviated from the original CompChem schema [11].

The TURBOMOLE XML output documents are validated against an extension of the Chemical Markup Language (CML) Schema, and imported into an eXist, an open source native XML database (www.exist-db.org). In the first release of the software (TURBOMOLE-XML-eXist 1.0), all XML files went through a full validation process against the extended CML schema (see also use case 1). In the most recent version of the software infrastructure (Version 1.1), the validation is on correct XML syntax only (use case 2), in response to the observation made, that all calculations which failed (aborted, incomplete) were identified already at this level.

Within PSI4, as a computation is running the data are collected in a simple, custom Python class designed analogously to the data structures in the CompChem convention. Whilst this storage structure in no way restricts the possible ultimate output formats, we have found the CompChem convention to describe a reasonable, useful way to organize the information. At the end of the computation, the data is loaded into an object from the standard Python library ElementTree. Using the ElementTree library, the XML output is easily generated. Outputs from a variety of different types of computations have passed the test of the CML/CompChem validator.

We use the eXist database to archive the XML results files in various collections, each one representing a series of (related) calculations. Queries are performed either using XQuery or XPath language. By default, both languages deliver XML output, which then can be transformed into the required final format (input for statistics or visualization packages, LaTeX tables, etc.). This is commonly done using XSLT (Extensible Stylesheet Language Transformation; see Figures 1 and 2). XSLT as well as XQuery can both deliver text or graphical output (SVG format) directly.

Figure 3: XQuery

The simplest way to query (a collection of) XML documents is with XPath. This language allows addressing a specific node of an XML document to retrieve the value of its elements and attributes. XPath also allows conditional searches along with simple arithmetic. XQuery, often referred to as the "SQL of XML", on the other hand, is a more complex language. It is built on XPath, but allows one to perform FLWOR expression (For-Let-Where-Order-Return) based searches. The output of the query can be text or graphical (see Figure 3). In the use cases presented here, most of the searches were performed using XQuery (for an illustration of a query see. Figure 4). It should be noted that writing a complex XQuery search is more demanding, certainly if the programmer is new to the declarative programming paradigm. It should also be noted that the XML framework is ideal for building APIs, and hence can be considered to be language agnostic – allowing the user to choose a preferred query language if so desired.

Figure 4: XQuery-Outputs

The InfoMol infrastructure, installed on a virtual server accessible to all members of the project, is operated by the ETH Zurich IT services.

3. InfoMol Illustrations and Use Cases

The TURBOMOLE-XML-eXist infrastructure was applied to solve "real chemistry" problems, each of which is, or will be, published separately in the corresponding specialized literature. The first illustration or use case presented covers a typical and very straightforward method benchmark study. The second use case is about incorporating quantum chemically derived electronic structure information into drug design, an area where data-centric modelling is very established. Finally, the third illustration is about the complexity of establishing a protocol for screening libraries of light-harvesting organic materials to be used in organic heterojunction photovoltaic devices. It also illustrated the increasingly close interaction between first-principles and data-centric modelling.

Illustration 1: Quantum Chemical Method Benchmark

In molecular modelling and simulation as well as in method and program development, method benchmarks represent a frequent and repetitive task that calls for automation. The RI-MP2-F12 method implemented in TURBOMOLE is particularly suited for the calculation of the weak intermolecular interaction energies. The method, however, relies on four different basis sets, A (main basis), B, C, and D (auxiliary basis sets), each of which needs to be optimized. Also, the basis sets are interdependent, and both, the result and the stability of the method relative to a choice of quartet of basis sets needs to be monitored. For an array of weakly interacting compounds (twelve different dimers of ethylene; Figure 5), 100 quartets of basis sets ABCD were explored. The interaction energies of the twelve ethylene dimers are within a narrow window (5 kJ/mole), and the correct ordering (weakest to strongest interaction energy) therefore requires high accuracy in the calculation. For this numerical experiment, we started from a high quality main basis A (triple zeta), allowing the three auxiliary basis sets to be (nominally) equally good (also triple zeta) or better (quadruple and quintuple zeta). For all calculations, counterpoise corrections were performed in order to have an estimate for the basis set superposition error, i.e. the error observed for the dimer,

where each of the two ethylene molecules will profit from the presence of the basis set of its neighbor. This spurious stabilizing effect disappears, once the two ethylene molecules are separated (dissociated). The effect scales with the size of the basis, and will disappear with very large basis sets.

Figure 5: Ethylene

In this use case, the XML output data of more than 5,000 calculations were archived, and the analysis based on XQueries gave a clear picture on the basis set dependence of the target observable, the interaction energy of an ethylene dimer. In Figure 6, the relationship between the basis set quality, the compute time, and the quality of the computed interaction energy is shown as a 2D plot. The main result is that the error in the computed energy strongly responds to the quality of auxiliary basis set D, whereas an improvement of the auxiliary bases B and C beyond the quality of the main basis (triple zeta T) has no impact, and only makes the computation more expensive. Also, the experiment shows that very large basis sets are required to achieve the accuracy targeted.

Figure 6: Method-Performance

From the technical perspective, this use case shows that with the database solution the human labor involved in the analysis of the data in method benchmarking is marginalized, and that it is no longer the time determining step in this task. At the same time, the frequency of errors is drastically reduced. For more detail about this project see [8].

Illustration 2: Exploring Electronic Structure-Activity Relationships in Biochemically Active Compounds (“drugs”) Using Quantum Chemical Data.

In this use case, an array of 337 biochemically active compounds was explored quantum chemically in an attempt to relate their electronic structure with their activity using machine learning tools to expedite the screening process [12] [13]. The goal behind this study is to explore a particular representation of the electronic structure of these compounds as a descriptor for their interaction with different types of macromolecules (“targets”), similar to the approach used in [14]. Whereas in biomedical research, the in silico drug design has a longstanding tradition, it is not until recently that quantum chemically derived information is being used as a source for the definition of descriptors.

The initial set of molecular structures for these compounds was extracted from the Protein Data Bank (PDB, [15]). A second set of structures was obtained using the 3D structure generator software CORINA (https://www.molecular-networks.com/products/corina) [16] frequently used in drug design to obtain the (Cartesian) atomic coordinates of a molecule directly from its constitution, i.e. from a one-dimensional representation of the chemical formula. Finally, departing from the PDB and CORINA geometries as an initial guess, the molecular structures were optimized using the quantum chemical model selected. The quantum chemically optimized structures for a given

molecule will not necessarily be identical as the method will search the minimum energy structure nearest to the initial (PDB, CORINA) geometry.

The parameter space to be explored is determined by the size of the array (337 molecules), the number of quantum chemical models (2; BP86/6-31G* and BP86/TZVP), the number of atomic charge models (3; Mulliken, Löwdin, Natural Atomic Population Analysis (NPA)), and the number of types of molecular structures considered (6; PDB, CORINA, quantum chemically optimized molecular geometries starting from the PDB and from CORINA structures for both basis sets). About half of the 14,000 atoms present in the 337 compounds are hydrogens. Experimental information about their position in 3D space is rarely available. There are three options to treat the charges of these hydrogen atoms: they are (i) ignored, (ii) projected onto the atom the hydrogen is bound to, or (iii) considered explicitly. Therefore, in total, for this array we will have to perform and analyze the results of 337*2*3*3= 6,066 atomic charge evaluations for each of the six arrays of structures.

To illustrate the usefulness of the TURBOMOLE-XML-eXist (Version 1.1) system, we focus on one particular aspect of the study, namely the analysis of the differences among the structures in the original PDB and CORINA arrays, and the two quantum chemically optimized structures, but for just one of the two models used (BP86/6-31G*)). In Figure 7 we see histograms of all interatomic distances between any two atoms (left) and all interatomic distances between carbon atoms, but now shown for two different sources (PDB and CORINA). The interatomic distances are grouped into separate "bins" with a width of 1 Angström (Å). Whereas the first bin in the carbon-carbon distances histogram (right) is empty, there is a small population in the corresponding bin on the left hand side. These contain O-H and N-H distances, all smaller than 1.0 Å.

Figure 7: Interatomic-distances

The panel on the right hand side of Figure 7 also shows that there are differences in the carbon-carbon distances between the CORINA and the PDB set. If we "zoom" into the range of covalent carbon-carbon bond lengths, i.e. the range between 1.1 and 1.7 Å (Figure 8), we see that there are no triple bonds (typically about 1.20 Å long) in this array, but that there is a small population of double bonds (about 1.33 Å). Most distances correspond to single bonds (about 1.50 Å) or aromatic (benzene-type) bonds (about 1.40 Å). We observe that the CORINA code uses a narrow spectrum of (parametrized) distances when representing a given type of bond. The spread of experimental bond lengths (PDB) is much larger.

Figure 8: CC-distances

Finally, the comparison between the initial (CORINA, PDB) and the respective quantum chemically optimized structures (Table 2) shows that one particular effect of the quantum chemical structure optimization is the formation of new intramolecular hydrogen bonds. In the original PDB set, a total of nine hydrogen bonds between oxygen atoms are observed. After quantum chemical optimization, another 30 hydrogen bonds are established. The same trend is observed for the CORINA structures. Obviously, the formation of additional intramolecular hydrogen bonds leads to stabilization of the

structure. The quantum chemical optimization was performed in vacuum (i.e. without interaction with any other molecules).

Table 2: HydrogenBonds

As in illustration 1, the human labor time for data analysis is drastically reduced. At the same time, the fact that all data are available in a structured format will allow to address other issues as they may come up during the study, thus providing better control, better (chemical) insight and depth of analysis.

Illustration 3: Searching libraries for light-harvesting organic materials for use in photovoltaic devices: about the difficulty of establishing a data processing protocol

In a joint venture with the Harvard CEP, we started to explore the properties of thousands of organic compounds, extracted from a pool of more than 3 million compounds which were scored according to their quantum chemically determined light-absorption properties (Scharber descriptor model; [18]). The Scharber descriptor model is not very selective, and the goal of this joint research is the development of new and more elaborate scoring functions predicting the power conversion efficiency of a given prospect much more reliably.

However, the process of generating an electric current from the absorption of solar light is a complex multi-step process, not yet fully understood in all detail. Even if it were understood, first-principles based modelling of the creation of the exciton ("electron and hole pair") by incident light, the dissociation of the exciton and the transport of the separated electric charges to the anode and cathode will be computationally extremely demanding and therefore prohibitively expensive. Thus, pure first-principle based screening is not realistic. To reduce the pool of "candidate compounds", we will still need to enhance the descriptor model by improvement of the underlying physical model, but we will need to incorporate methods to convert between these calculated compound properties, and their experimentally observed analogs. This can be achieved robustly by utilizing techniques from the area of machine learning, and it the subject of an upcoming paper [19].

The original Scharber model predicts the power conversion efficiency of a photovoltaic cell based on the first (longest wavelengths) absorption frequency of the compound, which relates to the voltage of the electric current triggered. All other cell performance factors, including the electric current generated by the cell, are represented in terms of (tunable) parameters. In a recent study, some of the present authors developed a more elaborate Scharber-type descriptor model [20]. It takes into account the specific light-absorbing properties of every molecule, thus addressing a distinctive feature of each individual compound. This will further reduce the set of prospects. Still, additional screening steps will be required in order to identify the top candidates within the initial array [21].

Part of the problem is that despite the great interest in the topic of photovoltaic devices, there is no central collection of data to relate theoretical and experimental findings: the information is spread

over many research groups. For that matter, the Harvard Organic Photovoltaic Dataset (HOPV15), soon to be made available to the public, was created [22]. Based on this collection, the Harvard group noticed that there is a significant discrepancy between computed and experimental information, and that the theoretical data need to be calibrated. Calibration based on machine learning tools helped improve the predictions made by the Scharber model for the HOPV15 compounds at least in view of the predicted voltage [19]. The prediction of the electric current generated remains difficult as it depends on numerous factors, including the morphology of the material and the ease of electron transport through the amorphous material.

Figure 9: HOPV15

This example clearly shows that brute-force virtual screening based on first-principle models is not an option here. In order to find the top prospects from millions of compounds, first-principles and data-centric prediction methods need to be combined (Figure 9). In order for this approach to be generally undertaken, it will be necessary for there to be a 'low-barrier' method for data-exchange between the computational and experimental collaborating teams – something which is not easily achievable with currently used methods.

4. Conclusions

In view of archival, analysis and exchange, but also in view of program interoperability, the computational quantum chemistry codes should provide output in machine-readable form. As long as the output document is structured, well defined (schema), and therefore transformable to other formats, the choice of actual data format is a secondary issue. Clearly, it would be nice to have at least a community *de facto* data standard, similar to the quantum chemical models. Recent history, however, has shown that it will be difficult to reach agreement across the entire discipline. The computational quantum chemistry community should start "learning by doing", and give evolution a chance.

We present one solution to the problem implemented in two different ways in two different program packages: TURBOMOLE and PSI4. The solution, XML using the semantics of CML and CompChem, is easily implemented: either the XML output is generated using the FORTRAN to XML conversion library and service (TURBOMOLE), or it is the code driver collecting the desired information for the machine-readable output (PSI4). In both cases, there is no need to parse the standard output and the interventions in the source code or the code driver are minimal.

In view of data analysis, the early experience shows that both the efficiency as well as the depth of analysis are both positively affected. In routine applications, such as a method benchmarks or a method validation against a given reference, the bottleneck is clearly shifted from the time it takes to analyze the data to the time to run the calculations (execution time; see use case 1). Illustration 2 shows that the possibility to look at your data from different angles at low effort provides more insight, also encourages to look at the data more closely. This way the depth of analysis will benefit, and ultimately support the "discovery process". Illustration 3, on the other hand, illustrates the

difficulty of mapping a complex process such as the generation of an electrical signal from solar radiation mediated by a light-harvesting compound. As shown based on illustrations 2 and 3, in order to develop potent descriptors and selective scoring functions, first-principles based molecular design will have to adapt data-centric modelling tools and *vice versa*.
From a technical perspective, working with a database using a declarative language such as XQuery marks a new challenge for this community. To address this challenge, APIs allowing the user to "speak" a procedural language such as Python when querying the database are one possible solution..

Since modelling and simulation is becoming more data-intense, computational quantum chemistry, like many other fields of computational science, will have to address the data processing bottleneck more vigorously. The first and most urgent step in this direction is to have the application codes generate output in machine-readable format.

5. Acknowledgements

This project was supported by the "InfoMol" Project of the Qatar Environment and Energy Research Institute (QEERI). RAK acknowledges a visiting professorship of the Department of Chemistry and Applied Biosciences of ETH Zürich, and HPL thanks for the opportunity to visit the research group of. Prof. Alàn Aspuru-Guzik, Harvard CEP, in April and June 2014.

References:

1. J. Hachmann, et al. The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid. J. Phys. Chem. Lett. 2 (2011) 2241–2251.
2. A. Elsener, C.C.M. Samson, M.P. Brändle, P. Bühlmann, H.P. Lüthi, Statistical analysis of quantum chemical data using generalized XML/CML archives for the derivation of molecular design rules,
   Chimia 61 (2007) 165-168
3. Gaussian 98, Revision A.9, Gaussian, Inc., Pittsburgh PA, M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, V.G. Zakrzewski, J.A. Montgomery Jr., R.E. Stratmann, J.C. Burant, S. Dapprich, J.M. Millam, A.D. Daniels, K.N. Kudin, M.C. Strain, O. Farkas, J.Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, J.A. Petersson, P.Y. Ayala, Q. Cui, K. Morokuma, D.K. Malick, A.D. Rabuck, K. Raghavachari, J.B. Foresman, J. Cioslowski, J.V. Ortiz, A.G. Baboul, B.B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R.L. Martin, D.J. Fox, T. Keith, M.A. Al-Laham, C.Y. Peng, A. Nanayakkara, M. Challacombe, P.M.W. Gill, B. Johnson, W. Chen, M.W. Wong, J.L. Andres, C. Gonzalez, M. Head-Gordon, E.S. Replogle, and J.A. Pople, 1998
4. P. Murray-Rust, H. Rzepa, Chemical markup, xml, and the world wide web. 4. cml schema, Chem. Inf. Comput. Sci. 43 (2003) 757-772.
5. XML, www.w3.org/TR/xml
6. JUMBOMarker, Generic Text Document Parser, Revision V 0.4, P. Murray-Rust.
7. Xindice, Revision 1.1b4, The Apache Software Foundation, 2004.
8. A. Glöss, M.P. Brändle, W. Klopper, H.P. Lüthi, The MP2 binding energy of the ethene dimer and its dependence on the auxiliary basis sets: a benchmark study using a newly developed infrastructure for the processing of quantum chemical data,
   Mol. Phys., 110 (2012) 2523-2534
9. E. Rossi, S. Evangelisti, A. Lagan, A. Monari, S. Rampino, M. Verdicchio, K.K. Baldridge, G.L. Bendazzoli, S. Borini, R. Cimiraglia, C. Angeli, P. Kallay, K. Ruud, J. Sanchez-Marin, A. Scemama, P.G. Szalay, A. Tajti, H.P. Lüthi, Code interoperability and standard data formats in quantum chemistry and quantum dynamics: the Q5/D5cost data model, J. Comp. Chem. 35 (2014) 611-621
10. W. Phadungsukanan, M. Kraft, J.A. Townsend, P. Murray-Rust, The semantics of Chemical Markup Language (CML) for computational chemistry: CompChem, Journal of Cheminformatics 4:15 (2012)
11. W.A. de Jong, A.M. Walker, M.D. Hanwell, From data to analysis: linking NWChem and Avogadro with the syntax and semantics of Chemical Markup Language, J. of Cheminformatics 5:25 (2013)
12. G. Schneider, De novo design - hop(p)ing against hope, Drug Discovery Today Technol. 10, (2013) e453-e460.
13. D. Reker, G. Schneider, Active learning strategies in computer-assisted drug discovery. Drug Discovery Today 20 (2013) 458-465
14. S. Renner, C.H. Schwab, J. Gasteiger, G. Schneider, Impact of conformational flexibility on three-dimensional similarity searching using correlation vectors, J Chem Inf Model., 46 (2006) 2324-32
15. P.W Rose, C. Bi, W.F. Bluhm, C.H. Christie, D. Dimitropoulos, S. Dutta, R.K. Green, D.S. Goodsell, A. Prlic, M. Quesada, G.B. Quinn, A.G. Ramos, J.D. Westbrook, J. Young, C. Zardecki, H.M. Berman, P.E. Bourne, The RCSB Protein Data Bank: new resources for

research and education, Nucleic Acids Res. 2013 Jan;41(Database issue):D475-482.
16. J Gasteiger, C Rudolph, J Sadowski, Automatic generation of 3D-atomic coordinates for organic molecules, Tetrahedron Computer Methodology 3 (6), 537-547
17. R: A language and environment for statistical computing, Version 2.13.0, R Foundation for Statistical Computing (www.r-project.org/), Vienna 2011
18. M. C. Scharber, D. Mühlbacher, M. Koppe, P. Denk, C. Waldauf, A. J. Heeger, and C. J. Brabec, Design rules for donors in bulk-heterojunction solar cells - Towards 10 % energy-conversion efficiency, Advanced Materials, 18 (2006) 789–794
19. G. Simm, E.O. Pyzer-Knapp, A. Aspuru-Guzik, A. Bayesian Calibration of Quantum Chemical Calculations to Experimental Observations: Application to Organic Photovoltaics. Manuscript in preparation (2015).
20. F.A. Alharbi, S.N. Rashkeev, F. El-Mellouhi, H.P. Lüthi, N. Tabet, S. Kais, An Efficient Descriptor Model for Designing Materials for Solar Cells, npj Comp. Materials, in press
21. J. Hachmann, R. Olivares-Amaya, A. Jinich, A. L. Appleton, M.A. Blood-Forsythe, L R. Seress, C. Román-Salgado, K. Trepte, S. Atahan-Evrenk, S. Er, S. Shrestha, R. Mondal, A. Sokolov, Z. Baod, A. Aspuru-Guzik, Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry – the Harvard Clean Energy Project, Energy Environ. Sci., 7 (2014) 698-704.
22. E.O. Pyzer-Knapp, G. Simm, T. Lutzow, K. Li, L. Serres, and A. Aspuru-Guzik, The HOPV15 Dataset, Manscript in preparation
23. NWChem: M. Valiev, E.J. Bylaska N. Govind, K. Kowalski, T.P. Straatsma, H.H.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations. Comput Phys Commun 181 (2011) 1477–1489.

Figure Captions

Figure 1 (Workflow): The flow of a typical TURBMOLE-XML-eXist computation. Each program module performs a specific task in the computation and generates, next to the standard output and the input for the next step, an XML output file which, after validation, is imported in the eXist database.

Figure 2 (Psithon): Schematic of the user-customizable collection of job data and its export in CML format within PSI4 along with (symbolic) "psithon" input example.

Figure 3 (XQuery): At the heart of InfoMol: the XQuery of the XML documents, created by TURBOMOLE and PSI4 (QC Programs) and imported into the eXist database, returns output in different formats (XML (default), plain text, Scalable Vector Graphics (SVG)). This return is either final, or can be transformed using tools such as Extensible Style Sheet Language Transformation (XSLT) to create input for post-processing and analysis applications (visualization, statistical analysis), or for presentation purposes (LaTeX tables, etc.). These documents may be uploaded to a file server for data exchange.

Figure 4 (XQuery-Output): Searching for relationships between the molecular properties within an array of compounds: an example of a simple XQuery. The query contains a FLWOR expression (*For-Let-Where-Order-Return*). The *for* loop is over the collection of TURBOMOLE CML outputs binding the value of the dipole moment of each compound to a variable called "dipole" through a *let* statement. Results are returned only if the HOMO-LUMO gap falls below a given threshold, and if the molecule is uncharged (*where* statements).

Figure 5 (Ethylene): Members K and L of the array of the twelve ethylene dimers. K marks the dimer with the strongest, and K the dimer with the weakest interaction energy (taken from the RI-MP2-F12/aug-cc-pV5Z reference calculation and measured in kJ/mole; values in parenthesis are basis set superposition corrected)

Figure 6 (Method-Performance): Basis set error versus compute-time evaluated for all possible quadruples of basis sets ABCD. Each spike represents one calculation. Red spikes refer to triple zeta quality for the main basis set A; blue spikes refer to quadruple zeta quality for A. The reference is the all-quintuple zeta calculation (only one single calculation possible under the given rules for basis set combinations). Only the spikes below the horizontal bar (dotted line) point at calculations with acceptable accuracy for this purpose. Spikes with one particular auxiliary basis set varied (X = B,C, or D)), show that the choice of basis D (green line) has substantial influence of the result obtained, whereas using better quality for B and C only increases the computational cost without having much impact on the accuracy of the calculation. "Best price/performance is obtained with the TQT5 combination of basis sets (blue square). Technical note: the data for this graph were extracted from the database using XQuery.

Figure 7 (Interatomic-distances): The number ("population") of interatomic distances found in each bin (PDB structures). All bins are 1.0 Å wide, i.e. the first (leftmost) bin is covering all pairs of atoms separated by 1.0 Å maximally (left). On the right hand side, only carbon-carbon distances are shown (PDB and CORINA structures). Technical note: the data were extracted using XQuery; the histograms were made using the program package R [17].

Figure 6 (CC-distances): Zooming into bin number 2, with a focus on all carbon-carbon distances found in the PDB and CORINA arrays in the range of 1.0 to 1.7 Å. Aromatic (benzene-like) and

single carbon-carbon bonds are most common in this range of interatomic distances. Technical note: the data were extracted using XQuery; the histograms were made using the program package R [17].

Figure 9 (HOPV15): Data-centric- and first-principle-based modeling interoperating in order to enrich (finding new lead prospects) and expedite (better descriptors) the screening process.

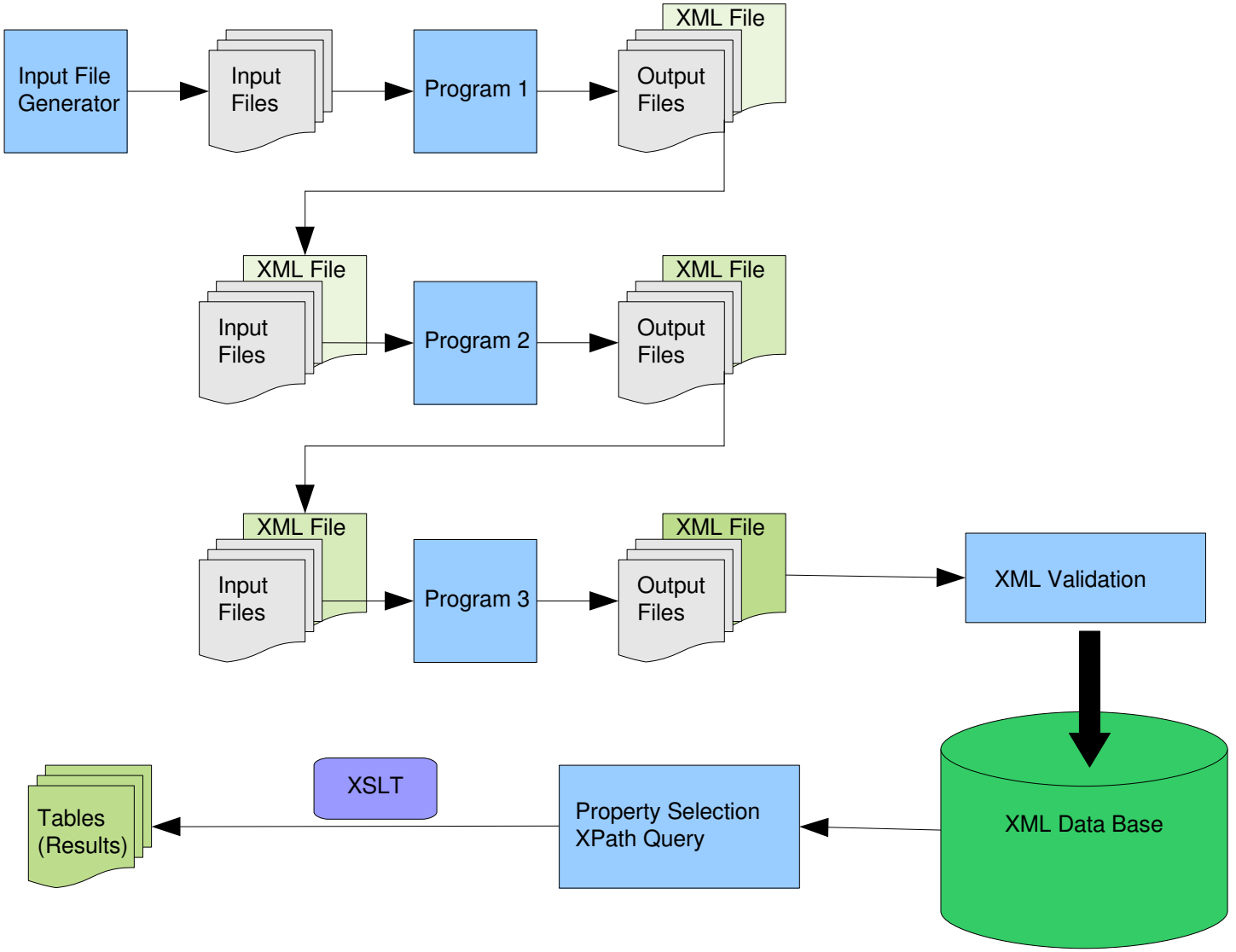
Table 1: The data items collected from each TURBOMOLE calculation (overview; most important items only). Items marked with an asterisk were introduced with version 1.1 of the system and are not yet defined in CML dictionaries.

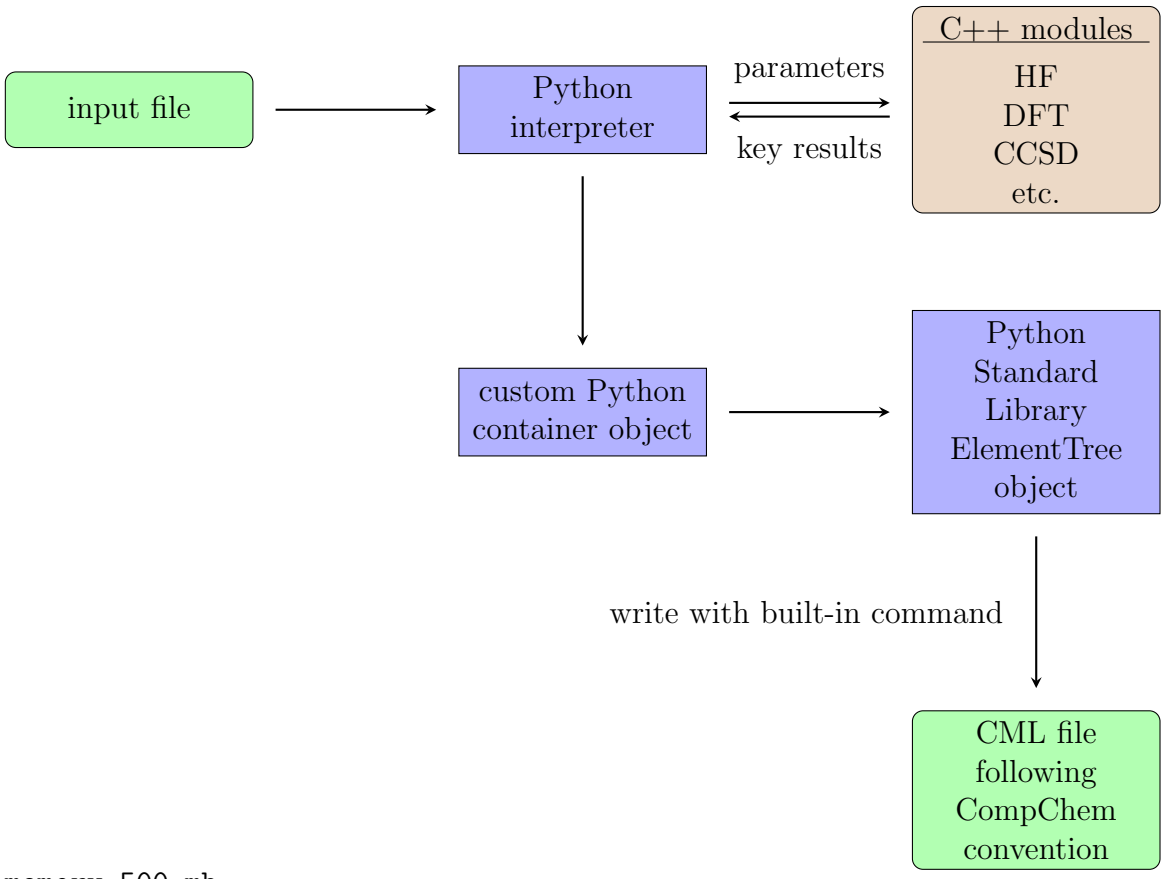
Source and Type of Information	**Data Items**
From Input of Calculation	
Molecular System	InChI code[2] AtomArray[1]: elementType[1], Cartesian coordinates[1], basRefs[3] (link to basis sets for this atom), atom ID[1] symmetry[1]
Basis Sets	basisSet[3]: elementType[3], basisSetName[3], basisSetType[3], contractionType[3]
From Output of Calculation	
Meta-information	UUID calculation identifier[2,3] Program-name[2,3], -version[2,3], -mode[2,3] (serial, parallel), OS[2,3], host name[2,3]
Execution time	Start and end time of calculation (wall time) [2,3]
Energies	Total energies and components (potential, kinetic)[3,4], RI-MP2-[R12/F12] pair energies (single point) [3,4], orbital energies and orbital occupation numbers[4]*, HOMO/LUMO energy difference[4]*
Molecular properties	charge[3,4], dipole[3,4], quadrupole[3,4], quadrupole anisotropy[3,4] Mulliken[4], Löwdin[4] and NPA atomic charges[4]*

[1] CML element or attribute
[2] used in CML element *metadata*
[3] for details see [8]
[4] used in CML element *property*

Table 2: Exploring the computed information at greater detail: the result of a query on the number of hydrogen bonds formed as a result of the quantum chemical optimization process. Only OHO bonds were searched for in these two collections.

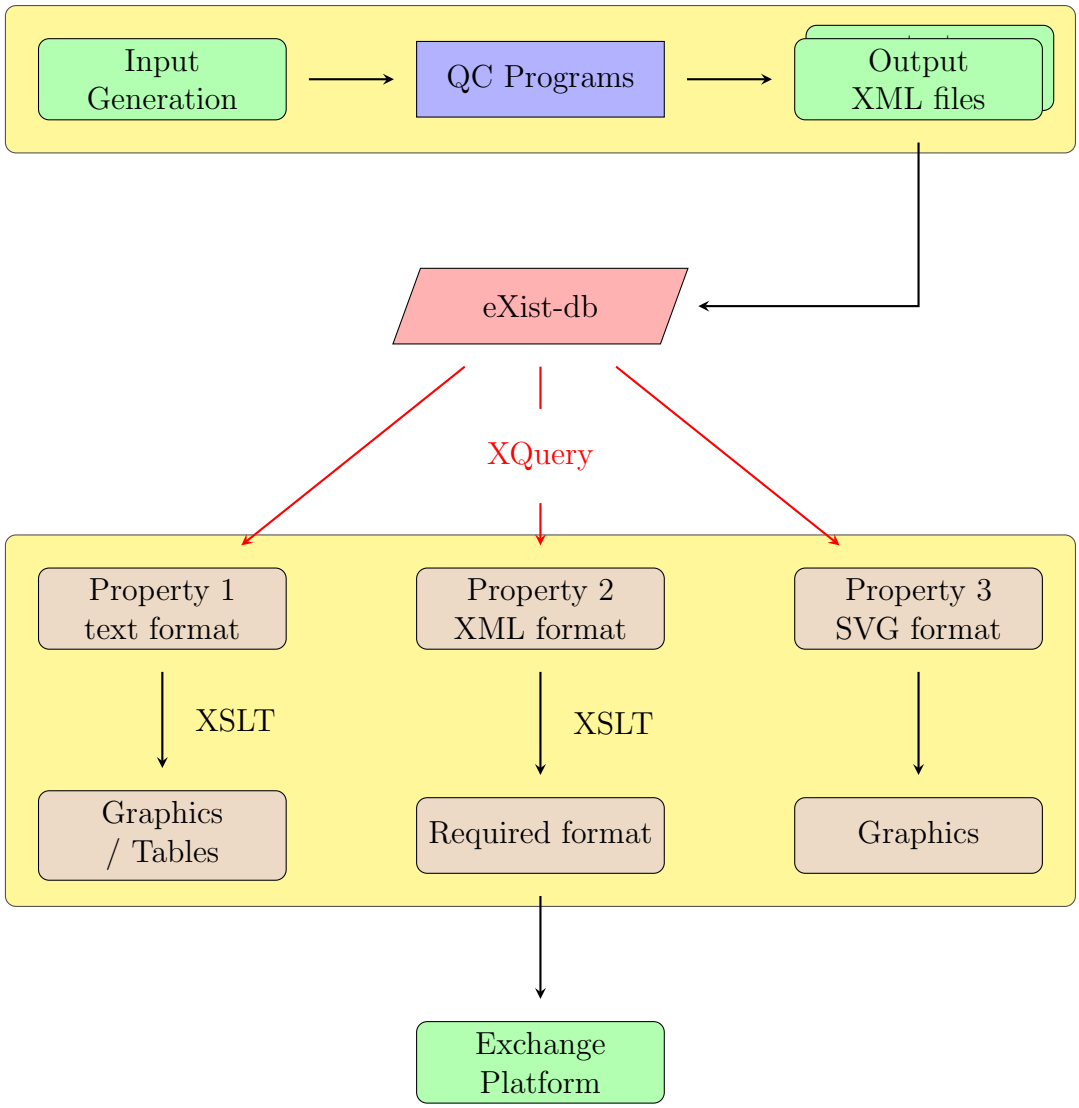
Structure Array	Number of Hydrogen Bonds
PDB	9
PDB OPT	39
CORINA	1
CORINA OPT	42


Input File Generator
Input Files
Program 1
XML File
Output Files
XML File
Input Files
Program 2
XML File
Output Files
XML File
Input Files
Program 3
XML File
Output Files
XML Validation
XML Data Base
Property Selection XPath Query
XSLT
Tables (Results)

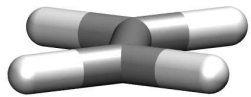


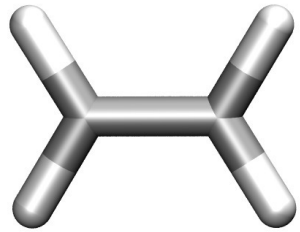

```
memory 500 mb
molecule h2o {
0 1
H
O 1 1.0
H 2 1.0 1 110.0
}

set { basis cc-pvdz }

from xml_psi import *

### Initialization
# Create jobList and job.
xml_psi_setup("optimization")

# Add parameters as desired.
xml_psi_init_add_string("psi:basis",
"cc-pvtz")

# Store geometry.
xml_psi_init_add_molecule( psi4 )
```

```
# Do calculation(s) and store results
#(### Finalization).
optimize('scf')

.
.
.

### Write out results
xml_psi_output()
```


Input Generation
QC Programs
Output XML files
eXist-db
XQuery
Property 1 text format
Property 2 XML format
Property 3 SVG format
XSLT
XSLT
Graphics / Tables
Required format
Graphics
Exchange Platform

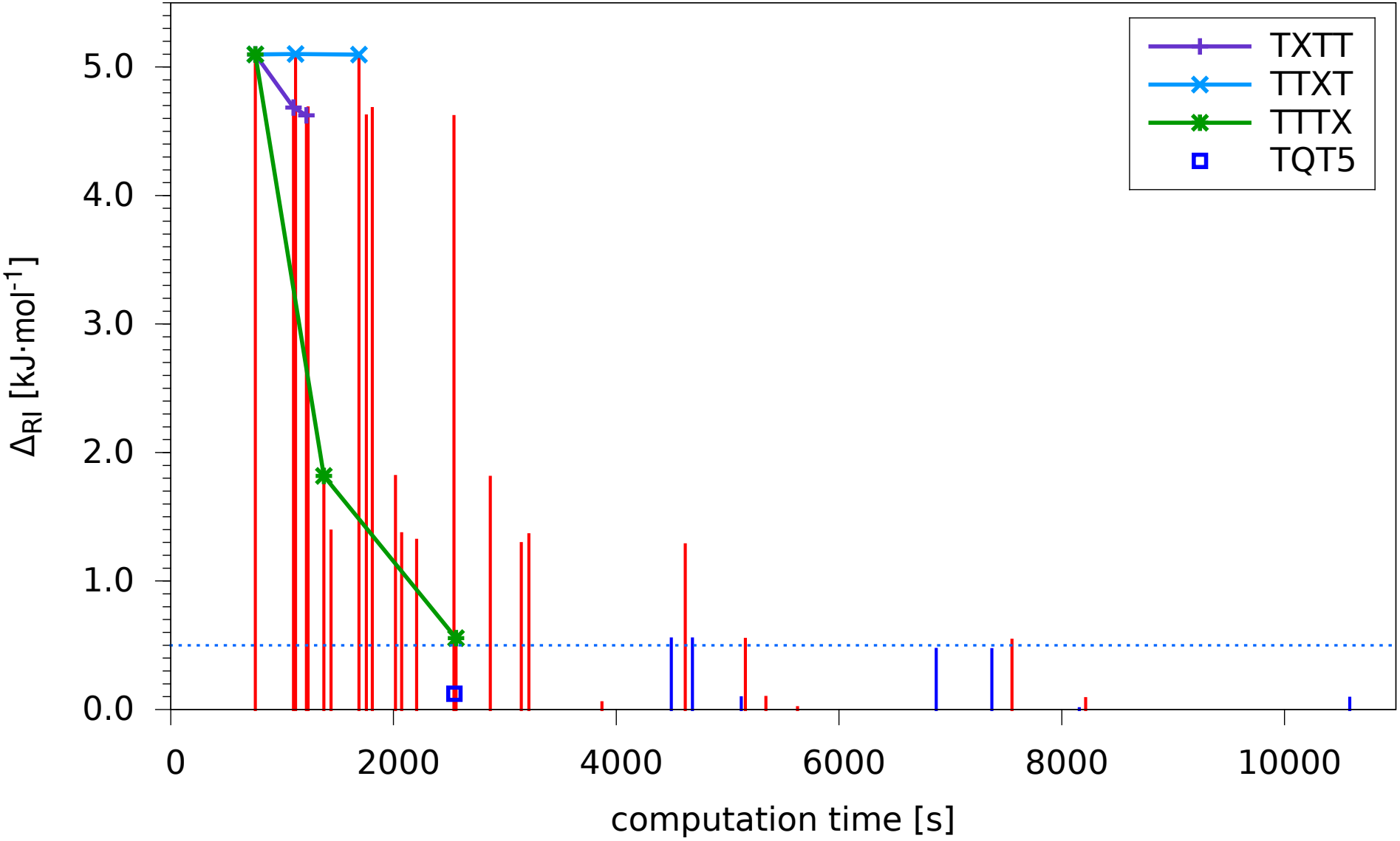
```
xquery version "3.0";

(: Find the molecules from the collection biomed_pdb with the largest electrostatic dipole moments.
   Restrict the search to neutral (uncharged) molecules with small HOMO-LUMO gap (0.2 au or less);
   print dipole moment (largest first), gap and InChi code of the molecule :)

declare default element namespace "http://www.xml-cml.org/schema";

for $i in collection("/db/biomed_pdb")/cml

let $code :=  xs:string($i/module/molecule/metadataList[@title="molecular system identifier" ]/metadata/@content)
let $dipole := xs:float($i//propertyList[@title = "electrostatic moments"]/property[@title = "dipole value"]/scalar)
let $charge := xs:float($i//propertyList[@title = "electrostatic moments"]/property[@title = "total charge"]/scalar)
let $gap := xs:float($i//propertyList[@title = "homo lumo gap" ]/property[@title = "HOMO/LUMO_GAP"]/scalar)

where $gap < 0.20 and abs($charge) < 1.e-10

order by $dipole descending

return
        <result>
        {$dipole, $gap, $code}
        </result>
```

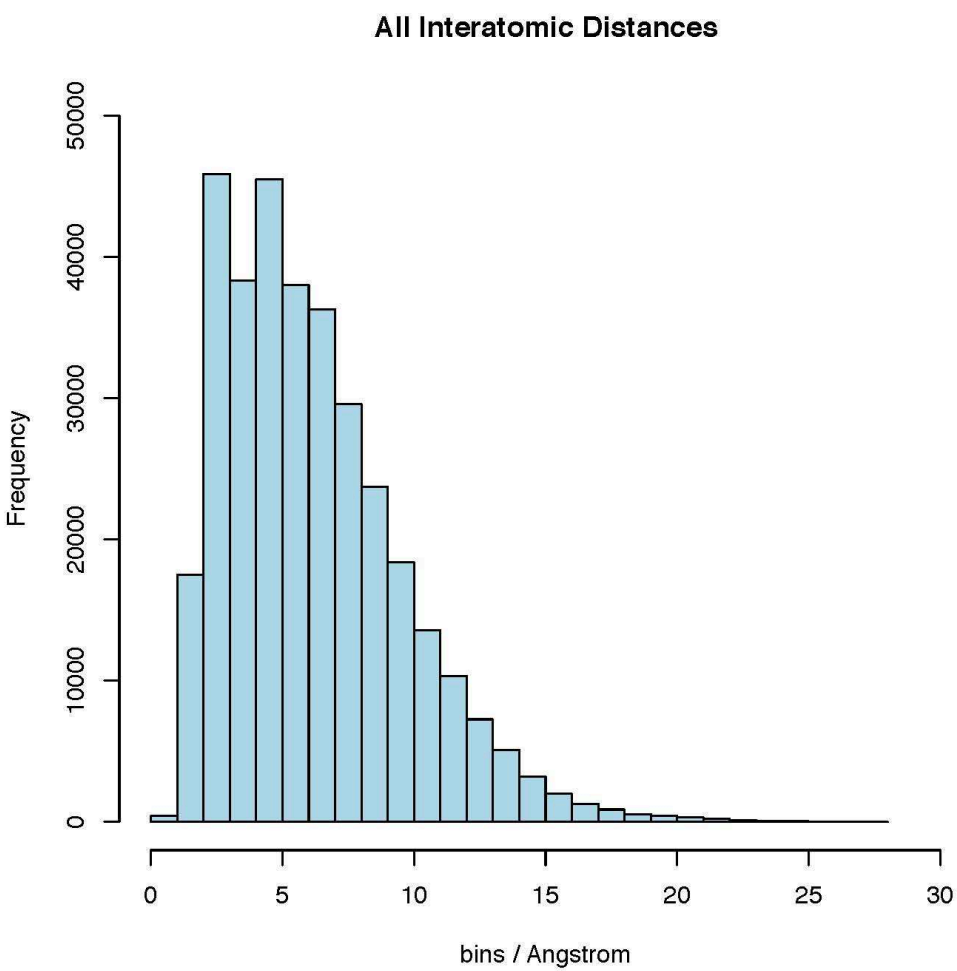


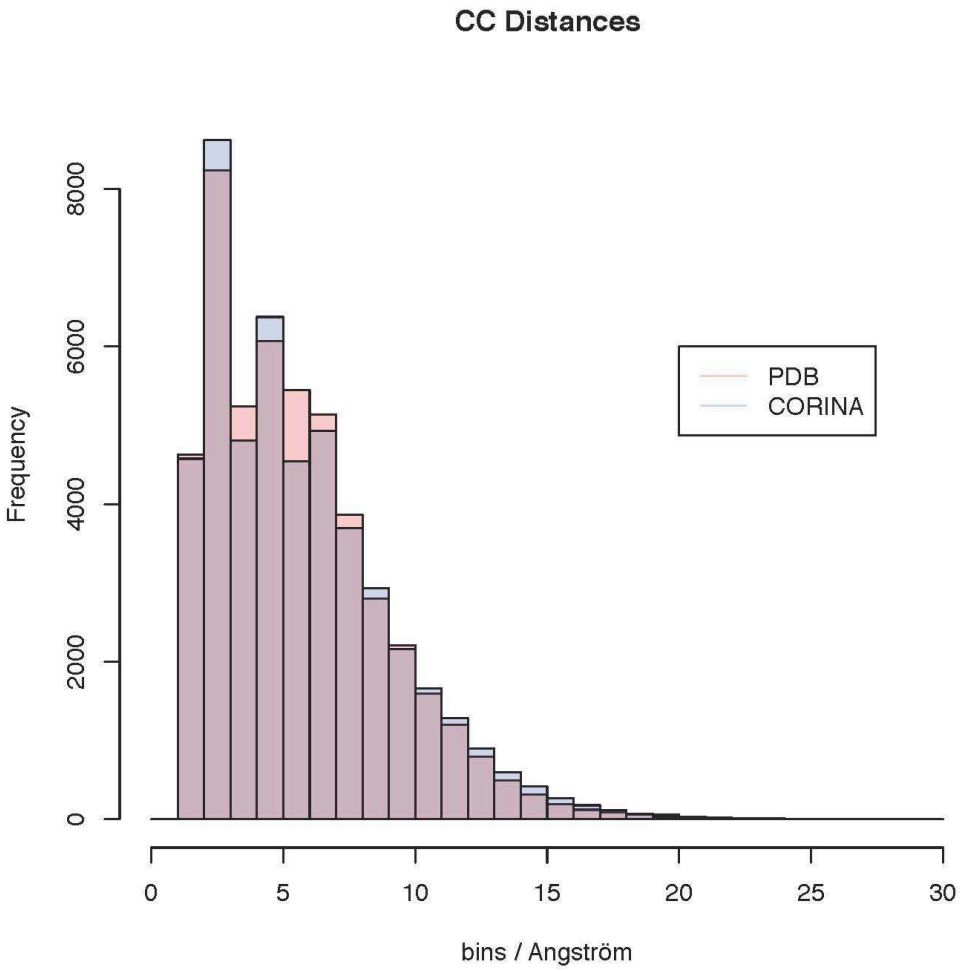


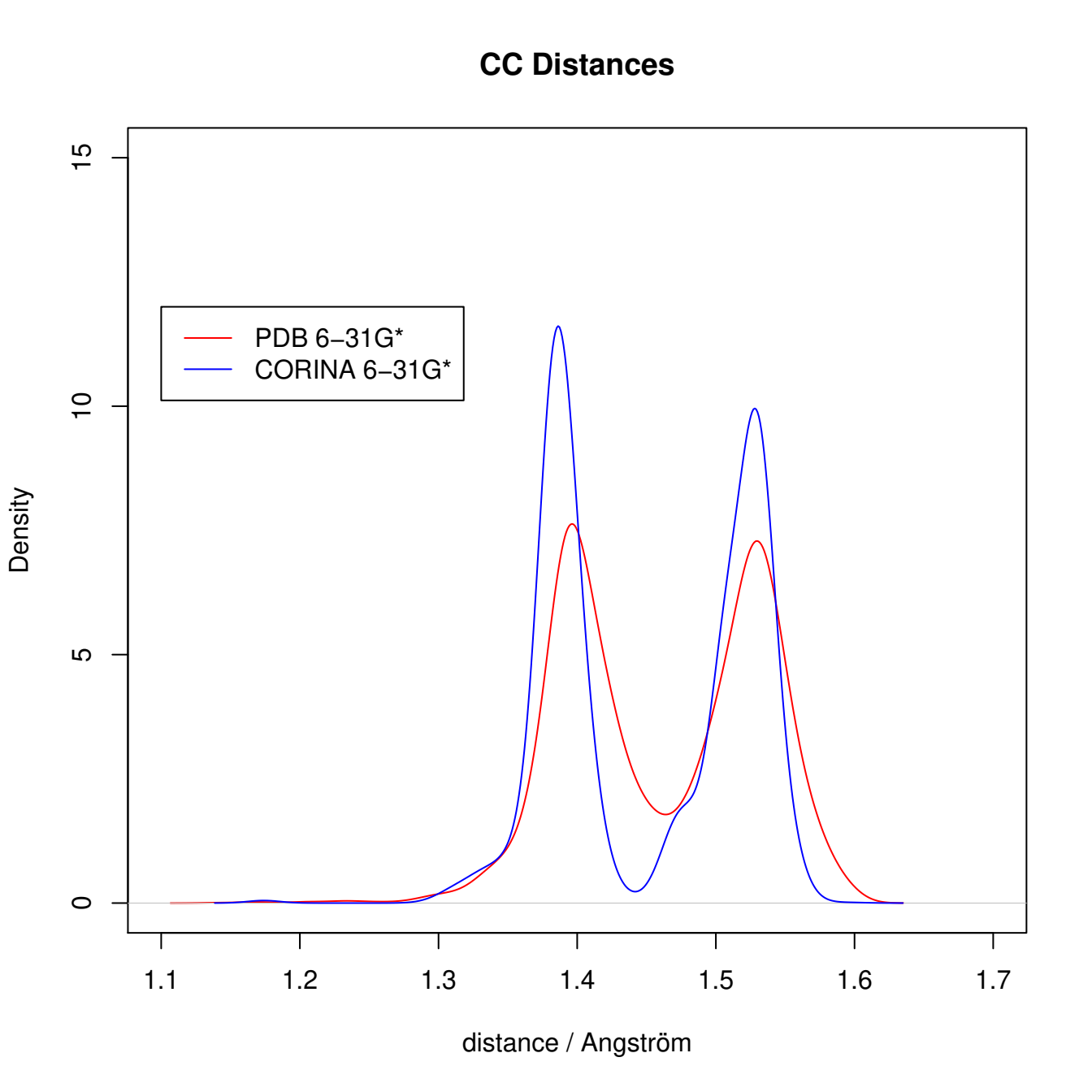
L -0.697 (-0.668)

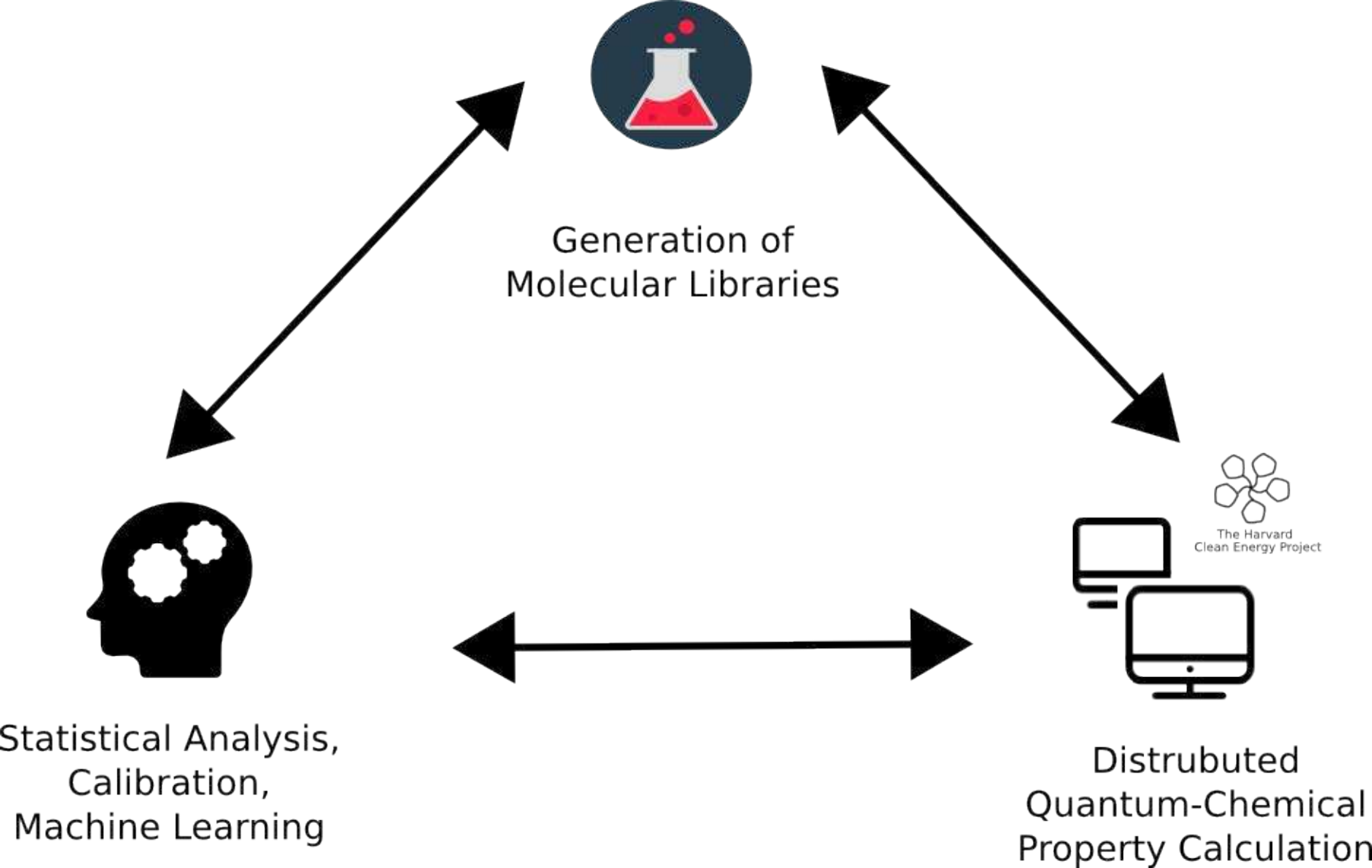
K -6.566 (-6.517)


TXTT
TTXT
TTTX
TQT5
5.0
4.0
3.0
2.0
1.0
0.0
Δ_{RI} [kJ·mol^{-1}]
0
2000
4000
6000
8000
10000
computation time [s]


All Interatomic Distances
Frequency
0
10000
20000
30000
40000
50000
0
5
10
15
20
25
30
bins / Angstrom


CC Distances
Frequency
0
2000
4000
6000
8000
PDB
CORINA
0
5
10
15
20
25
30
bins / Angström


CC Distances
PDB 6–31G*
CORINA 6–31G*
Density
0
5
10
15
1.1
1.2
1.3
1.4
1.5
1.6
1.7
distance / Angström


Generation of
Molecular Libraries
The Harvard
Clean Energy Project
Statistical Analysis,
Calibration,
Machine Learning
Distrubuted
Quantum-Chemical
Property Calculation