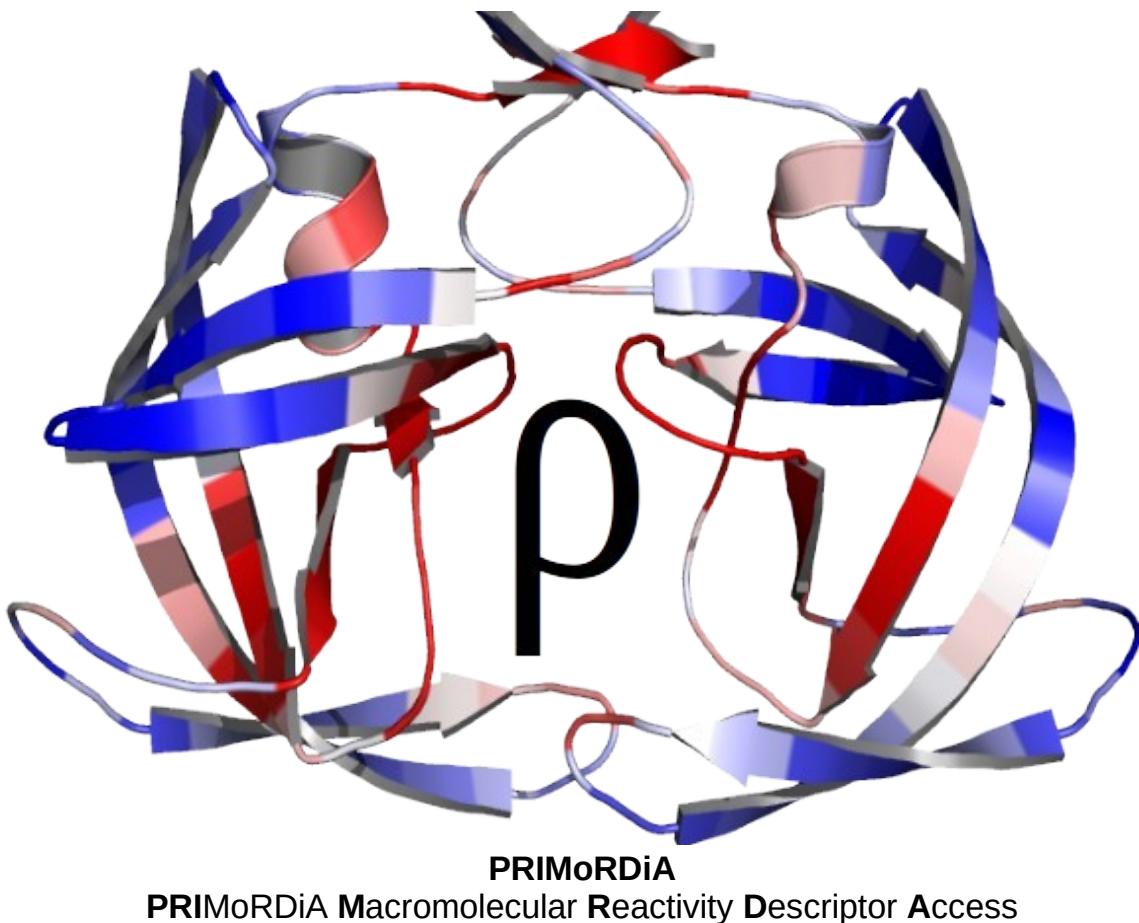

PRIMoRDiA Tutorials

Igor Barden Grillo
(PRIMoRDiA.software@gmail.com)
github.com/igorChem

January 29, 2023



Contents

1 Tutorial 1: Frozen Orbital Approximation	6
1.1 Context	6
1.2 Preparation of Files and Software Execution	7
1.3 Analysis Results	9
1.3.1 Condensed Local Descriptors	11
1.3.2 Volumetric Local Descriptors	12
2 Tutorial 2: Finite Differences Approximations	18
2.1 Context	18
2.2 File Preparation and Execution	18
2.3 Results Analysis	19
3 Tutorial 3: Band Descriptors	26
3.1 Context	26
3.2 File Preparation and Execution	26
3.3 Analysis Results	29
3.3.1 Condensed descriptors	30
3.3.2 Volumetric Representation	34
3.4 Spacial representation for residues and the Density of State	40
4 Tutorial 4: Acid strength analysis	43
4.1 Context	43
4.2 Required Files and Execution	43
4.3 Results Analysis	44
5 Tutorial 5: Enzymatic Reaction Analysis	51
5.1 Context	51
5.2 Required Files and Execution	52
5.3 Results Analysis	54
6 Tutorial Extra: Google Colabs	65

Introduction

This document contains tutorials for the PRIMoRDiA program. PRIMORDIA (**P**RIMoRDiA **M**acromolecular **R**eactivity **D**escriptors **A**ccess) is software written in *C++*, developed for post-processing of the results of computational chemistry packages, producing a wide variety of quantum descriptors: electronic properties and reactivity of chemical systems.

PRIMoRDiA implements the main reactivity descriptors of the conceptual DFT theory, including the mathematical definitions for Fukui's frontier electrons and Pearson's Hard and Soft Acid and Base theories (HSAB). PRIMoRDiA was developed with a focus on the calculation of descriptors and other more common electronic properties for large molecules that are relevant to biological processes, and therefore there are descriptors specifically modified to meet the particularities of these systems.

We also offer a user guide with general information about the theory and details about the use and interpretation of the results generated by the program. PRIMoRDiA was developed at the Laboratory of Quantum and Computational Chemistry of the Federal University of Paraíba.

To download the data from the repository you can use the following git command:

```
computer@user:$ git clone https://github.com/igorChem/PRIMoRDiA1.Ov.git
```

Through this command, the repository folder will be downloaded in the directory where your terminal is open, and can be easily updated with the command:

```
computer@user:$ git pull
```

Also, on the site itself it is possible to download a '.zip' containing all the data from the repository. However, it is strongly recommended the download of the last stable version. These versions are guaranteed to have all their features properly tested.

PRIMoRDiA is a program that is under constant development and cloning the repository at any point other than those marked as stable can generate errors in use.

Link to latest version: <https://github.com/igorChem/PRIMoRDiA1.Ov/releases/tag/v1.25>

More specific information on compiling, installing and using the already compiled binaries, see the user guide provided in the repository.

About the tutorials in this document, they cover practically all the

features of the program, but above all, what to do with the results of PRIMoRDiA, since the assembly of the input and execution normally only requires two commands in a terminal.

PRIMoRDiA works with three descriptor calculation modes, classified by the type of approximation performed. The first three tutorials deal exactly with these three specific ways of obtaining the descriptors, tutorials 4 and 5 work applications in possible objects of study and tutorial 6 teaches how to run the program on Google's cloud computing platform to overcome the lack of a version program for Windows or for users with limited computing resources.

The list of tutorials is just below:

1. Frozen Orbital Approximation:
2. Finite Difference Approximation:
3. Band Descriptors
4. Acid Strength Analysis
5. Enzyme Reaction Analysis
6. Use of Google Colabs

The first tutorial is very important to make, it introduces a lot of information about the input assembly and basic executions in the Pymol software to visualize the results. Also, in this same tutorial, the interpretation of global descriptors and the effect of electronic structure methods on these quantities is developed, information that is relevant for all methods and calculation modes.

The second has little difference in terms of input assembly and program execution, so we took the opportunity to provide more details about the use of Pymol. The third is the main tutorial for those who want to work with macromolecules, such as proteins/enzymes, introducing how to use the two implemented methods for the calculation of local descriptors and other various parameters for the analysis of results.

The fourth tutorial intends to demonstrate the use of descriptors within the context of a research question, bringing as an example the investigation of the acidic strength for different amino acids, showing how the descriptors can be used and combined to extract information from quantum calculations made for these systems.

The fifth tutorial brings a very advanced application of the software, resembling the analyzes we make in our recent publications. This tutorial applies Band descriptors, which are the third type of calculation mode uniquely implemented in PRIMoRDiA for dealing with macromolecules, by analyzing a trajectory that corresponds to an enzymatic catalysis reaction path.

In the last tutorial, we replicated the first one on Google's cloud computing platform. The idea is to provide an example of using the platform. From this first example, all other tutorials can be run with these resources.

1 Tutorial 1: Frozen Orbital Approximation

In this tutorial it will be demonstrated how to use the program to obtain the reactivity descriptors using the information from molecular boundary orbitals. As this is the first tutorial, it will be exemplified how to obtain the descriptors for acrolein molecule with the electronic structure calculated with the four computational chemistry packages that are supported by PRIMoRDiA.

1.1 Context

Theories of reactivity have always been much more explored within organic chemistry through the study of frontier molecular orbitals, the infamous HOMO and LUMO. This study comprises the analysis of the special distribution of these orbitals in molecules and their energy values.

And why are these orbitals important? First, molecular orbitals serve to describe the probability density of finding an electron in three-dimensional space; Second, during chemical reactions, electrons are transferred between molecules, reorganizing chemical bonds and therefore geometries. These boundary orbitals, such as HOMO (Highest Occupied Molecular Orbital) and LUMO (Lowest Unoccupied Molecular Orbital), are the ones that describe the spatial distribution of electrons most likely to be transferred, or the levels of virtual energies most likely to receive electrons.

When reactivity theories were demonstrated through mathematical treatment based on Density Functional Theory (DFT), frontier molecular orbitals became an valid approximation for local functions and energy values for reactivity descriptors. This is known as the Frozen Orbital Approximation (FOA). The entire theoretical part of this approach, the conceptual part and derivation of the reactivity descriptors can be found in our User's Guide.

This approach has some important advantages, such as the need for a faster calculation protocol than finite differences, and the consolidation of concepts based on molecular orbitals. The main disadvantage is that sometimes the HOMO and LUMO orbitals are not the ones that result in the most accurate representation of the system's reactivity.

In this tutorial, we will show you how to calculate reactivity/quantum descriptors using FOA, how to generate and analyze the results with Pymol graphics software, and the interpretation of tables with values for each atom. Also, as this is the first tutorial, more attention will be given to technical details about the input and scripts generated by the program, interpretation of descriptors and explanation of concepts that will be useful for the next tutorials.

1.2 Preparation of Files and Software Execution

Basically, what PRIMoRDiA does is extract the electronic properties from output files of quantum chemistry method packages, and with this information do the mathematical treatment necessary to obtain the reactivity descriptors. These descriptors can be global, representing a property for the entire molecular system considered in the quantum calculations, or local, which in turn describe a property for a given point in the three-dimensional molecular space.

The local descriptors for atoms are called condensed, where the part of the local function calculated for the descriptors is assigned to the atoms individually, making this quantity bound to the coordinates of the nucleus of that atom.

To exemplify the operation and use of the program, the electronic structure of the acrolein molecule was calculated using the four quantum chemistry packages that are supported by PRIMoRDiA (up to the present version), GAMESS, MOPAC, ORCA and Gaussian, each with a different calculation method. The graphical representation of the molecular arrangement of acrolein is shown just below in Figure 1.

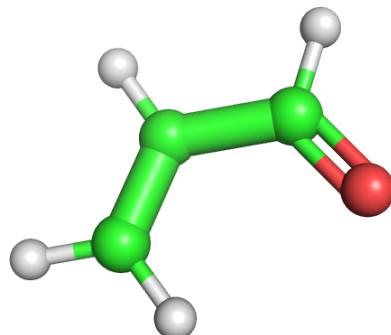


Figure 1: Ball-and-sticks representation for the acrolein molecular structure.

The files that PRIMoRDiA needs to be in the Tutorials_Files compressed folder, which in turn has the folders for each tutorial. In addition to these files, a text file is needed to tell PRIMoRDiA what to do, the famous input file. To generate this file from the electronic structure files that are in the folder, just run the program with the "-input" flag and the specific parameters for the calculation you want to execute. Let's run the command below, inside the directory with the mentioned files.

```
computer@user:$  
/path/to/PRIMoRDiA/PRIMoRDiA_1.25v -input -op 1 -p mopac -grid 40
```

In this case, PRIMORDiA will look for files with the extension ".aux" in the folder, as local descriptor calculations are required for mopac output files. After executing this command, two new files will appear in the directory: primordia.input and primordia.log. If this does not occur as described or there is an error message, check if the path to the executable is correct. Remembering that it is necessary to change the "/path/to/primordia/" to the address of the folder where the executable is located. Open the file "primordia.input", the result should be as shown in the following text box (Listing 1).

Listing 1: Generated Input by the command in PRIMoRDiA.

```
#RT normal
#PR eband 1 extrard pymols
1 acrolein.aux true 10 mopac
```

This command is useful when there are multiple files to be calculated in the folder. Let's understand the options used together with the flag

- *-op:*
Indicates the calculation option to be passed in the next argument.
- *1:*
Calculation option passed, it indicates that the descriptors will be obtained with FOA.
- *-p:*
Indicates the source program of the output files.
- *mopac:*
Is the keyword that indicates that the output files in the folder came from the MOPAC quantum chemistry package.
- *-grid:*
Flag that serves to indicate the grid resolution for the results of descriptors in volumetric representation. The higher the value more resolution, but also higher computational cost.
- *40:*
Grid resolution.

However, there are other output files in the folder from other computational chemistry packages, and to include them you need to manually edit the file so that it looks like the following text box (Listing 2).

Listing 2: Edited input for the tutorial execution.

```
#RT normal
#PR extrard pymols
1 acrolein.aux true 40 mopac
1 acrolein_orca.out true 40 orca
1 acrolein_gam.log true 40 gamess
1 acrolein_gauss.fchk true 40 gaussian
```

The first line of input must always start with "*#RT*", which by default is followed by the keyword "*normal*", which serves to indicate the Run Type (calculation type). The second line "*#PR*" indicates that the next parameters will be to control general aspects of the calculations.

The "*pymols*" parameter indicates that we want the program to write scripts to automate the visualization of results in Pymol. All lines in the input that start with "#" will not be considered, thus you can write comments in those lines. The lines that start with the integers, 1, 2 or 3, will be considered for calculation using the corresponding option.

To run PRIMoRDiA is very simple, from our input created and edited, just run the following command

```
computer@user:$ /path/to/PRIMoRDiA/PRIMoRDiA_1.25v -f primordia.input
```

The "-f" flag indicates that the next argument is an input file formatted for the program. The input file can have any name and does not need to have the ".input" extension. After running this command the program should indicate the total execution time, and in the same directory where the data and input are located, several new files of different types should appear. We will explain them to analyze in the next part of the tutorial.

1.3 Analysis Results

Reactivity descriptors are calculated as global quantities, system-wide properties, and locations, which is a mapping of reactivity over the topology of the molecular system. PRIMoRDiA writes files with the extension ".GRD" for the global descriptors of each entry in the system, and ".lrd" for the condensed locations and ".cube" for the volumetric locations. It also writes a file ending with ".global", where all the calculated global descriptors are gathered.

If we open this file, *primordia.global*, we will find data as shown in Figure 2. This file is already formatted to have the ideal spacing to be copied to a spreadsheet, as shown in Figure 3.

```

GRD HOMO_E LUMO_E T_Energy Energy_CAT Energy_AN IP EA ECP Hardness Softness Electrophilicity GAP N_MAX HOF
acrolein -10.25700000 -0.37600000 -26.34006077 0.00000000 0.00000000 10.25700000 0.37600000 -5.31650000 4.94050000 0.20240866 2.86055786 9.881
acrolein_orca -10.72560000 1.49950000 -5192.03767000 0.00000000 0.00000000 10.72560000 -1.49950000 -4.61305000 6.11255000 0.16359784 1.7406999
acrolein_gam -10.74306072 1.49390586 -8032.83699098 0.00000000 0.00000000 10.74306072 -1.49390586 -4.62457743 6.11848329 0.16343920 1.74771389
acrolein_gauss -7.00216526 -1.85905344 -191.90932743 0.00000000 0.00000000 7.00216526 1.85905344 -4.43060935 2.57155591 0.38886963 3.81681362

```

Figure 2: Global descriptors results for the run input.

GRD	HOMO_E	LUMO_E	T_Energy	Energy_CAT	Energy_AN	IP	EA	ECP	Hardness	Softness	Electrophilicity	GAP	N_MAX
acrolein	-10.257	-0.376	-26.34006077	0	0	10.257	0.376	-5.3165	4.9405	0.20240866	2.86055786	9.881	2.15221131
acrolein_orca	-10.7256	1.4995	-5192.03767	0	0	10.7256	-1.4995	-4.61305	6.11255	0.16359784	1.7406999	12.2251	1.50937007
acrolein_gam	-10.74306072	1.49390586	-8032.836991	0	0	10.74306072	-1.49390586	-4.62457743	6.11848329	0.16343920	1.74771389	12.23696658	1.51167445
acrolein_gauss	-7.00216526	-1.85905344	-191.90932743	0	0	7.00216526	1.85905344	-4.43060935	2.57155591	0.38886963	3.81681362	5.14311182	3.44585886

Figure 3: Global descriptors results imported to a sheet.

For the Ionization Potential (IP), the values were around 10 eV, with the exception of the results calculated from the GAUSSIAN output, which used the DFT method and presented 7 eV. This value is the energy required to extract an electron from the system and is the basis for other reactivity descriptors along with electron affinity (EA), which is the change in energy when the system receives an electron.

Electronic Chemical Potential (ECP) is the conceptual DFT descriptor that measures the propensity of a molecular system to donate electrons, which is very similar between the *ab initio* and DFT methods used here for acrolein, showing the molecule more likely to donate electrons when descriptors come from semi-empirical calculus.

The ECP is often a negative value that comes from the derivative of electronic energy in relation to the number of electrons, that is, systems require energy to donate electrons. Thus, we interpret these results as the least negative, as they require less energy to donate electrons and thus react through the charge transfer process.

Hardness is the second derivative of electronic energy with respect to the number of electrons and therefore the derivative of the ECP with respect to the number of electrons. These descriptors indicate the system's resistance to donating electrons, and a system with higher hardness is more chemically stable.

The highest hardness values calculated for acrolein were with the Hartree-Fock methods and the lowest with DFT. Thus, we can conclude that the DFT method indicated the acrolein with a greater propensity to donate its electrons and participate in a chemical reaction through this process. The hardness being the derivative of the ECP indicates how it will change as the electron density moves through the system. Thus, a system may have a high ECP, but the high hardness will make electron density transfer more difficult.

Stabilization energy of a system to receive all possible electrons is measured by the total electrophilicity. The corresponding maximum

number of electrons that the system can receive is given by the descriptor nMax and, finally, the difference between the boundary molecular orbitals, HOMO and LUMO, 'gap' is the last descriptor in the list. For the FOA method, the gap and the hardness are the same quantity, although for the finite difference method this is not true and the two pieces of information may be relevant.

1.3.1 Condensed Local Descriptors

Now, let's analyze and interpret local descriptors, which are quantities assigned to a point in three-dimensional space. There are two common ways of representing local descriptors for molecules, which are implemented in PRIMoRDiA. The first is the condensate for atoms, where values are assigned to each atom and written as a list. All entries in the input file produce a file with the extension '.lrd', specifically for FOA methods (option 1 in the input file), the file suffix 'FOA.lrd'. So we should have four of these files in our directory, as we can verify using the 'ls' command in the Linux terminal. In the image below (Figure 4), the information from these files was transferred to a libreoffice calc table.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	acrolein																	
2	n	atom	charge	nucleophilicity	Electrophilicity	RAS	Nephtilicity	Softness	Hardness_A	Hardness_B	Hardness_C	Hardness_D	Multiphilic	Electrophilic	Fukushima	electron_dens	softness_dual	
3	1	C	0.33294001	0.05813121	0.24245778	0.1502945	0.18610597	0.03740503	0.03042091	4.94981225	0.53949589	0.8300977	0.18432657	0.53936451	0.29764723	4.03881718	0.03730929	
4	2	C	-0.3058	0.14870988	0.18610597	0.16740503	0.03740508	0.03388423	0.34933683	0.47364046	0.42232891	0.10699025	0.5323669	0.31778527	4.41101912	0.007507407		
5	3	C	-0.17526001	0.01019888	0.40576901	0.20834444	0.39449413	0.04217072	0.03884466	0.44304479	0.404056396	0.12672573	0.1257413	0	0.41659135	3.8540127	0.07992089	
6	4	H	0.1586	0.00439569	0	0.0219784	-0.00439569	0.00044446	1.03643919	0.48783043	0.44440703	0.04508659	-0.01257413	0	0.00439569	0.8415022	-0.00088973	
7	5	H	0.14487	0.00872356	0	0.00436178	-0.00872356	0.00088286	1.05040768	0.49567418	0.31656707	0.08947755	-0.02495425	0	0.00872356	0.85154018	-0.00176572	
8	6	H	0.164	0.00251001	0	0.00125501	-0.00251001	0.00025402	1.03093172	0.48824934	0.43045971	0.02574517	-0.00718003	0	0.00251001	0.83602856	-0.00050805	
9	7	H	0.1086	0.10010896	0	0.05005448	-0.10010896	0.01013146	1.03447281	0.49562144	0.5067473	0.1268176	-0.28636747	0	0.10010896	0.89140894	-0.02026292	
10	8	O	-0.42789	0.7663518	0.1656403	0.46599605	0.60071151	0.09432164	0.35164999	0.16773862	0.7981397	0.171837003	0.47382365	0.85218647	6.80067944	-0.12158921		
11	acrolein_orca																	
12	n	atom	charge	nucleophilicity	Electrophilicity	RAS	Nephtilicity	Softness	Hardness_A	Hardness_B	Hardness_C	Hardness_D	Multiphilic	Electrophilic	Fukushima	electron_dens	softness_dual	
13	1	C	-0.07785	0.00856342	0.37404656	0.19130499	0.36548314	0.03129708	8.24604405	0.63619646	0.53489816	0.65273068	0.65110281	0.17759495	5.39937145	0.05979225		
14	2	C	-0.008105	0.24890483	0.54868606	0.39879645	0.29978323	0.06524224	8.37360566	0.44065424	0.20259541	3.49241144	0.52183263	0.95510125	0.64918335	5.5735377	0.04904389	
15	3	C	-0.56422698	0.2065308	1.07524597	0.64094952	0.68692888	0.10485905	0.7308274	0.22478756	0.38280903	1.51195954	0.18168054	0.96361895	5.54246155	0.14209992		
16	4	H	0.247532	0	1E-08	0	1E-08	0	0.7783176	0.4617565	0.36054316	2E-08	0	0.50932434	0			
17	5	H	0.26056701	0	0	0	0	0	0.76843417	0.41462748	0.33547358	0	0	0	0.50285669	0		
18	6	H	0.28144801	0	1E-08	0	1E-08	0	0.71388428	0.45478462	0.3507261	1E-08	2E-08	0	0.46716262	0		
19	7	H	0.212835	0	1E-08	0	1E-08	0	0.48190146	0.23154057	0.23160303	1E-08	2E-08	0	0.65174309	0		
20	8	O	-0.35926399	0.10140029	0.22133924	0.16136977	0.11993895	0.02639975	15.36692881	0.31593567	0.1829073	1.41947716	0.20877772	0.38528519	0.20468475	10.09424832	0.01962175	
21	acrolein_gam																	
22	n	atom	charge	nucleophilicity	Electrophilicity	RAS	Nephtilicity	Softness	Hardness_A	Hardness_B	Hardness_C	Hardness_D	Multiphilic	Electrophilic	Fukushima	electron_dens	softness_dual	
23	1	C	-0.073806	0.00699686	0.28112824	0.14361255	0.27503137	0.02347192	8.6915237	0.51738682	0.3358784	0.4854781	0.49067615	0.49133172	0.17446733	5.68444676	0.04495091	
24	2	C	-0.107531	0.26555519	0.61913133	0.44234326	0.35357615	0.07229623	8.77926891	0.19666325	3.77779943	0.61794994	1.08206443	0.64130931	5.83986492	0.0577882		
25	3	C	-0.54746097	0.20837279	1.07380778	0.64109208	0.68543499	0.10477928	8.33696295	0.3811123	0.23160617	3.84272923	1.51253275	1.87670876	0.95337421	5.52910717	0.141446	
26	4	H	0.27123001	0	1E-08	0	1E-08	0	0	0.7179924	0.47259819	0.37152194	2E-08	0	0	0.46399688	0	
27	5	H	0.26720201	0	0	0	0	0	0	0.74361405	0.41969553	0.34082147	0	0	0	0.48614274	0	
28	6	H	0.28829199	0	1E-08	0	1E-08	0	0	0.68949957	0.48125159	0.3517602	2E-08	2E-08	0	0.45075849	0	
29	7	H	0.22368699	0	1E-08	0	1E-08	0	0	0.99595306	0.49271314	0.23081399	2E-08	2E-08	0	0.62716783	0	
30	8	O	-0.32143301	0.08810155	0.1764545	0.13227802	0.08835294	0.02161941	5.31295784	0.32614372	1.21008673	0.15441567	0.30839197	0.20206548	10.04424539	0.01444033		
31	crolein_gauss																	
32	n	atom	charge	nucleophilicity	Electrophilicity	RAS	Nephtilicity	Softness	Hardness_A	Hardness_B	Hardness_C	Hardness_D	Multiphilic	Electrophilic	Fukushima	electron_dens	softness_dual	
33	1	C	0.22238933	0.01880785	0.28125051	0.15206643	0.2665171	0.05913402	3.1386547	0.49194098	0.78983154	0.59873875	1.01724633	1.08903224	0.25463977	4.8983123	0.10364043	
34	2	C	-0.12598789	0.2457265	0.14716403	0.18586834	0.07774062	0.07227855	3.45258034	0.41825103	0.35722551	1.29890901	0.51619766	0.30211787	5.56387603	-0.03010186		
35	3	C	-0.29290217	0.1925081	0.40238878	0.2198198	0.40113797	0.08548124	3.3060900	0.36735684	0.43886779	-0.64672785	1.51306887	1.60454563	0.347595	5.15120404	0.15599038	
36	4	H	0.14092536	0.01139882	1E-08	0	0.00569941	0.00221633	0.4417659	0.44310391	0.46600681	0.07918318	-0.04350713	0E-08	0.00889375	0.69697705	-0.00443265	
37	5	H	0.16184655	0.00131596	0	0	0.00056978	-0.00131596	0.00025587	0.41460000	0.39551872	0.32166033	0.00921457	0	0.00107743	0.64603505	-0.00045174	
38	6	H	0.18443182	0.00635514	0	0	0.00317757	-0.00635514	0.00123594	0.38956864	0.44996962	0.40554100	0.04449973	-0.02425638	0	0.00607536	0.60988491	-0.00247132
39	7	H	0.1056417	0.14217657	0	0	0.07108829	-0.14217657	0.02764408	0.46551061	0.45037121	0.42904196	0.99554385	-0.5266147	1E-08	0.10597094	0.84657186	-0.05528815
40	8	O	-0.39634469	0.60178937	0.2421949	0.42199214	-0.35954464	0.16409999	5.47261397	0.20563931	0.19254757	3.76357534	-1.37250505	0.9244120	0.725056543	0.30395383	-0.13895357	
41																		

Figure 4: Condensed descriptors.

These values are mapped to the coordinates of each atomic nucleus, as shown in the figure above (Figure 4), presenting acrolein oxygen as the most reactive to receive an electrophilic attack.

The carbon atoms of the double bond also show significant values of nucleophilicity. The highest electrophilicity value is for carbon 3. This type of representation provides a means of quantitative, intermolecular and intramolecular comparison.

In Figure 5, we show the values mapped to the atomic centers, for

the electrophilicity and nucleophilicity descriptors, calculated with the different packages of computational chemistry and electronic structure methods.

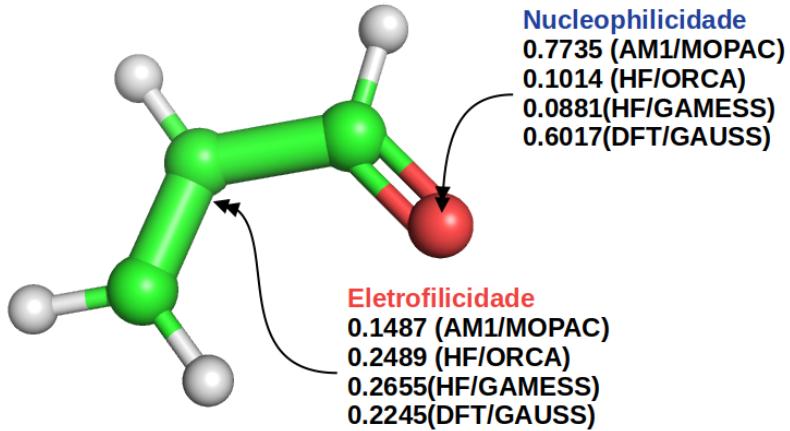


Figure 5: Most reactive atoms in the acrolein molecule.

1.3.2 Volumetric Local Descriptors

For the volumetric representation, the generated Cubes files are used to render the descriptors in three-dimensional space. In the next images, we will show how to generate this graphic representation for the most used reactivity descriptors using the Pymol software.

You can download Pymol from the repository (for Linux). We strongly recommend that you do the basic Pymol tutorials, especially if you want to use it to generate your scientific figures, as we have done several times in our publications.

The next steps consist of generating the volume rendering in Pymol, an object that graphically represents the scalar field in three-dimensional space, assigning each voxel of the same value the same color and opacity. Opacity is an important feature, as it is necessary to define at least two levels of scalar field values with different opacities in order to render a volume. If these two or more levels have the same opacity, only the one closest to the edges of the grid will appear.

PRIMoRDiA writes scripts to automate the visualization of these files in Pymol, let's run one of them for acrolein calculated with semi-empirical. Open Pymol and type the command that is highlighted in the figure below (Figure 6).

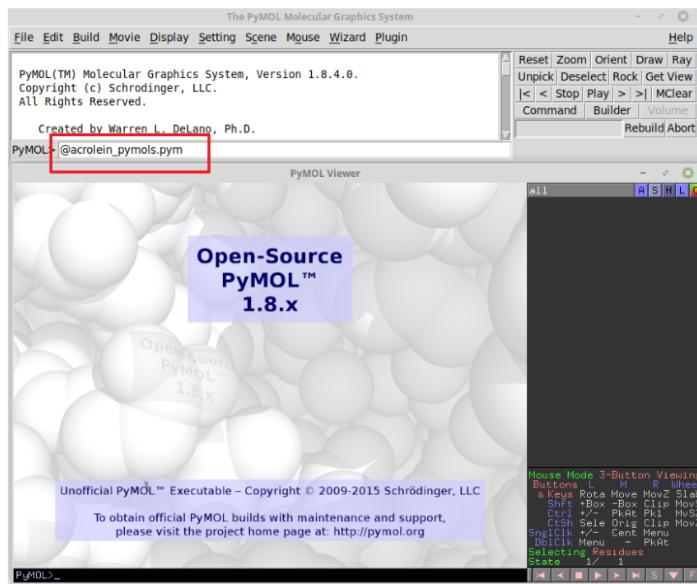


Figure 6: Pymol window showing the command to execute the script and visualize the descriptors in volumetric representation.

The command highlighted in the red box in Figure 6 is repeated below

```
@acrolein_pymols.pym
```

After executing this command, all cube files will be loaded, as shown in the figure below (Figure 7).

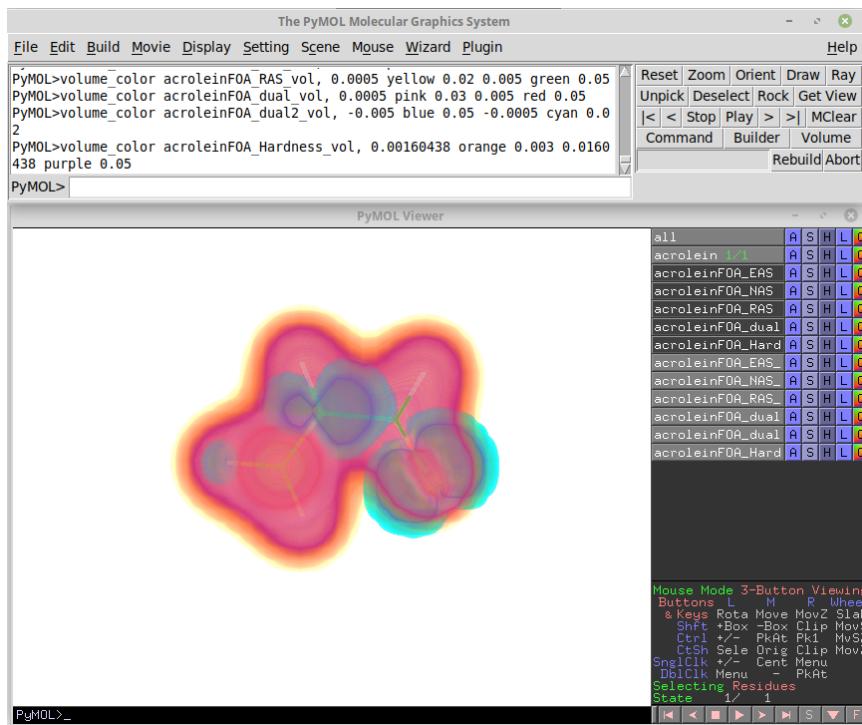


Figure 7: Visualization of volumetric descriptors generated by the script produced by PRIMoR-DiA.

By selecting the acroleinFOA_EAS_vol object in the side window of Pymol, we will only show the nucleophilicity descriptor. Using the command in Pymol

```
draw 1000,800,antialias=2
```

An image is rendered with a resolution of 1000X800 pixels, and with a parameter to increase the image quality. followed by the command of

```
png acrolein_eas.png
```

saves the image to a file in *png* format in the folder, as shown below in (Figure 8)

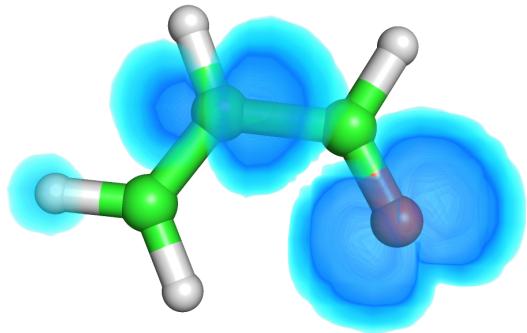


Figure 8: Saved image example for the nucleophilicity descriptor for the acrolein molecule.

The same process can be repeated for the other volume objects. You can also use the Pymol interface to change surface values, colors, and opacities. Below, the local hardness is the subject of this example from Figure 9.

And in Figure 10, we show the descriptors for acrolein considering all the calculation methods tested, showing the susceptible places of the molecule to receive electrophilic (blue) and nucleophilic (pink) attacks.

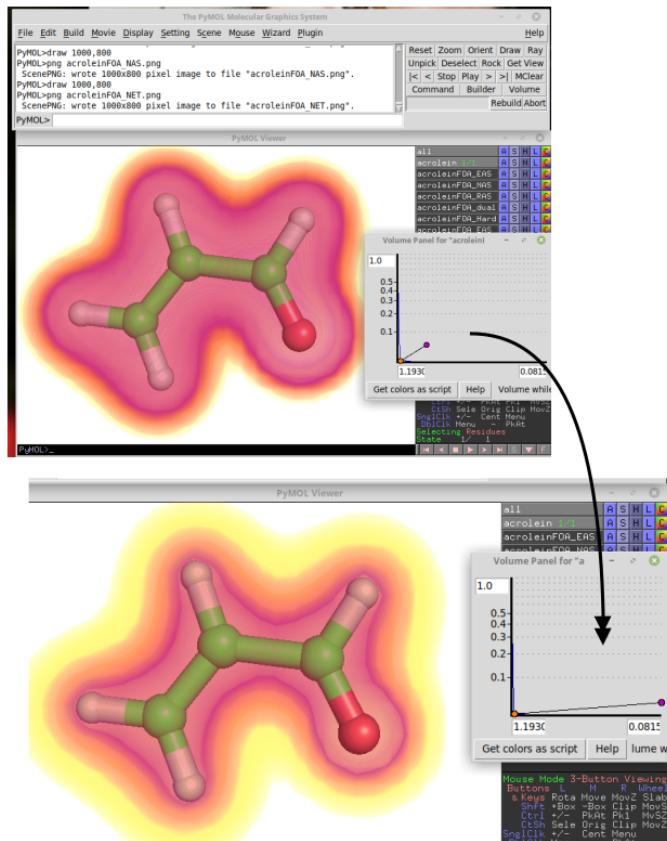


Figure 9: Local hardness rendering.

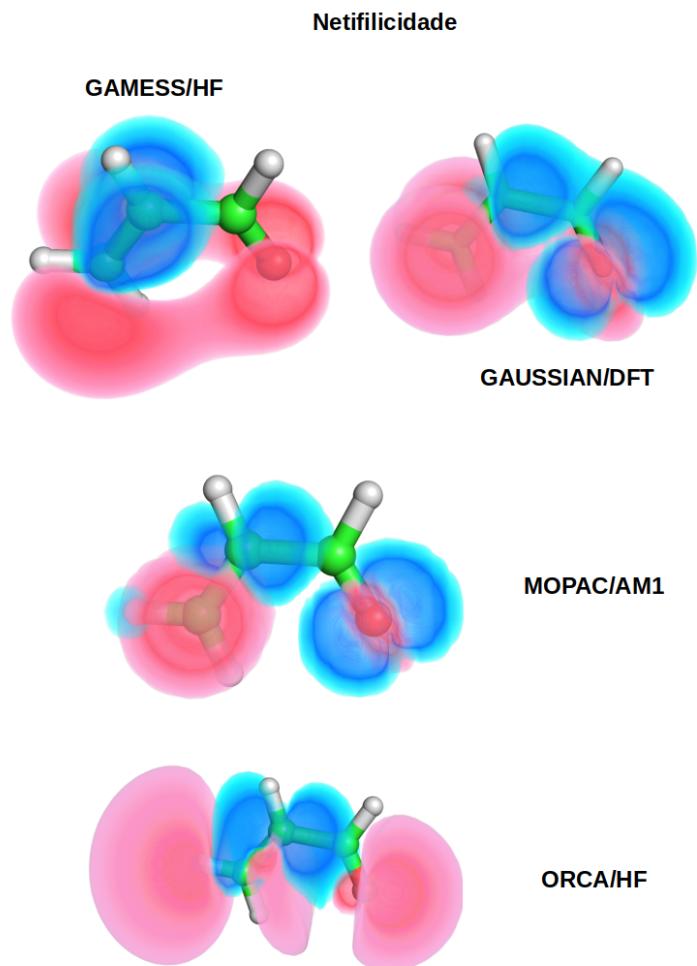


Figure 10: Netphilicity descriptor for acrolein using different electronic structure methods worked on in this tutorial.

Therefore, in this tutorial we can also see the influence of electronic structure methods in obtaining the descriptors, which is significant but tends to be much smaller than the effect on the energy results, causing most of the time the descriptors can result in the same patterns even using less expensive quantum methods.

2 Tutorial 2: Finite Differences Approximations

This second tutorial demonstrates the calculation of descriptors in PRIMoRDiA with the approximation of finite differences, using the benzene molecule as a model system.

There is little difference between this tutorial and the first one in the part of executing commands and analyzing results tools. About what is needed for the calculation, the finite difference method requires two extra files of electronic structure, for the positively and negatively charged states in relation to the electronic state that one wants to calculate the properties.

2.1 Context

In general, the finite difference method is a widely used tool for calculating derivatives numerically, that is, replacing an analytical method. In this more specific case, the calculation of these electronic properties defined as derivatives with respect to the number of electrons, the finite difference method is an almost intuitive way, since it is not possible to derive a non-continuous function.

Therefore, the finite difference method is a traditional way of calculating descriptors and is implemented in PRIMoRDiA. Also, due to the fact that the properties are calculated using information from the difference of complete electron densities, the local descriptors calculated by this method tend to be more accurate with respect to molecular orbital effects other than HOMO and LUMO. However, the need for two more single points (energy calculation) and obtaining the total electronic density makes this method computationally more expensive.

In this tutorial we will use the benzene molecule as an example for the descriptors with this approximation method. The benzene molecule does not have specific reactivity points due to its symmetry, but it is possible to visualize properties related to aromaticity and ring geometry.

2.2 File Preparation and Execution

All the files necessary to carry out the tutorial are in the repository with the code, in the compressed file named `Tutorials_Files`. In this example we will use the electronic structure for the benzene molecule in three states of charge, calculated with the semi-empirical method AM1 in the MOPAC computational chemistry package. Note that in the input for running this tutorial, shown in the Listing 3 text box, the names of the MOPAC output files have the extension `".out"`.

Listing 3: Edited input to execute tutorial 2

```
#RT normal
#PR pymols extrard
2 benzene.out benzene_cat.out benzene_an.out true 30 1 mopac
```

These files do not contain the complete electronic structure for the systems, however PRIMoRDiA understands that in the same folder these files are also corresponding with the extension ".mgf". These files have the matrices needed for linear algebra operations to correctly obtain the total electron density with the data coming out of the MOPAC.

For large molecules, MOPAC cannot generate this file, as it is too expensive to write for structures with many atoms. In this case, the calculation of finite differences has to be done using files of the ".aux" type, from which PRIMoRDiA generates a total electron density based on molecular orbitals, but does not guarantee its exact integration for the number of electrons in the system.

The calculation of descriptors using finite differences for large molecules is indeed quite unusual, due to the computational cost of obtaining three electronic structures and the total density calculations for the entire system extension. However, with the appropriate modifications of the descriptors using the Frozen Orbital approach, FOA, it is not necessary to resort to finite differences to study proteins, DNA fragments and/or other molecules relevant to biological processes.

Going to run PRIMoRDiA itself, it's very simple and straightforward, once you have the input and the necessary files in the folder, just run the following command in your terminal

```
computer@user:$ /path/to/PRIMoRDiA/PRIMoRDiA_1.25v -f primordia.input
```

Exactly the same as shown in the previous tutorial. After executing this command, the program should finish without any error and generate several files with the results in the folder where the program ran.

2.3 Results Analysis

As in the previous tutorial, we are going to open the files containing the summary of the global and local descriptors, and finish the analysis by visualizing the volumetric local descriptors using the Pymol software.

In Figure 11, we show the file benzene_FD.GRD opened in a text editor, showing the values obtained for the global descriptors and other electronic properties that are extracted by PRIMoRDiA. The big difference to the first tutorial is the presence of electronic energy

values for the cationic and anionic states. Furthermore, the value of *gap* here remains the energy difference of the HOMO and LUMO for the electronic structure of the neutral state of charge, that is, the information regarding ionization potential and electron affinity is not redundant. in this case.

```

1 Calculus method: Finite differences approximation
2 name: benzene
3 HOMO E LUMO E T_Energy Energy_CAT Energy_AN IP EA ECP Hardness Softness Electrophilicity GAP N_MAX HOF
4 -9.80500 0.20390 0.00000 -3237.85948 -3247.47964 -9.46727 -0.15289 -4.81008 4.65719 0.21472 2.48399 10.00890 2.06566 23.20334 |

```

Figure 11: PRIMoRDiA output file for the global descriptors calculated for the Benzene molecule with the finite difference method.

Moving on to the analysis of condensed local descriptors, the file "benzeneFD.lrd" is shown in Figure 12 opened in a text editor. As we've seen before, this file is formatted so that it can be easily copied and pasted into spreadsheet software, as shown in Figure 13.

```

1 benzene
2 n atom nucleophilicity electrophilicity radicality netphiliicity hardness Vee hardness_lcp Fukui_pot_left Fukui_pot_right Fukui_pot_zero soft
3 1 C -0.051500 -0.060300 -0.055900 -0.008800 0.291247 2.152071 0.566609 0.573317 0.569963 -0.001890 -0.002577 -0.021859 0.000000 -0.147000 2.
4 2 C 0.168800 0.161400 0.165100 -0.007400 0.283744 2.461019 0.393545 0.398093 0.395819 -0.001589 0.007612 -0.018382 0.000000 -0.147000 3.2577
5 3 C 0.210500 0.222500 0.216500 0.012000 0.286553 2.315726 0.360641 0.349851 0.355246 0.002577 0.009982 0.029808 0.000000 -0.147000 3.092126
6 4 C -0.051600 -0.060200 -0.055900 -0.008600 0.291465 2.058173 0.566717 0.573314 0.570016 -0.001847 -0.002577 -0.021362 0.000000 -0.147000 2.
7 5 C 0.168800 0.161200 0.165000 -0.007600 0.284857 1.999467 0.393433 0.398172 0.395802 -0.001632 0.007607 -0.018878 0.000000 -0.147000 2.6631
8 6 C 0.210200 0.222400 0.216300 0.012200 0.287434 1.948558 0.360826 0.349759 0.355293 0.002620 0.009973 0.030305 0.000000 -0.147000 2.618937
9 7 H 0.063700 0.067100 0.065400 0.003400 0.277020 0.327542 0.308489 0.302755 0.305622 0.000730 0.003015 0.008446 0.000000 0.147000 0.454877 -
10 8 H 0.055300 0.056200 0.055750 0.000900 0.287474 0.330961 0.405863 0.401814 0.403839 0.000193 0.002570 0.002236 0.000000 0.147000 0.454944 -
11 9 H 0.053400 0.053200 0.053300 -0.000200 0.284322 0.331713 0.424469 0.429192 0.426830 -0.000043 0.002457 -0.000497 0.000000 0.147000 0.45493
12 10 H 0.063700 0.067200 0.065450 0.003500 0.273633 0.327542 0.308459 0.302787 0.305623 0.000752 0.003018 0.008694 0.000000 0.147000 0.454879
13 11 H 0.055300 0.056200 0.055750 0.000900 0.270863 0.330962 0.405809 0.401694 0.403751 0.000193 0.002570 0.002236 0.000000 0.147000 0.454946
14 12 H 0.053500 0.053200 0.053350 -0.000300 0.271063 0.331704 0.424203 0.428991 0.426597 -0.000064 0.002460 -0.000745 0.000000 0.147000 0.4549
15 |

```

Figure 12: PRIMoRDiA output file for the condensed local descriptors for the atoms of the benzene molecule, obtained with the finite difference method.

	A	B	C	D	E	F	G	H	I	J	K	L	
1	benzene	n	atom	nucleophilicity	electrophilicity	radicality	netphiliicity	hardness_Vee	hardness_lcp	Fukui_pot_left	Fukui_pot_right	Fukui_pot_zero	softness_du
2			1	C	-0.0515	-0.0603	-0.0559	-0.0088	0.291247	2.152071	0.566609	0.573317	0.569963
3			2	C	0.1688	0.1614	0.1651	-0.0074	0.283744	2.461019	0.393545	0.398093	0.395819
4			3	C	0.2105	0.2225	0.2165	0.012	0.286553	2.315726	0.360641	0.349851	0.355246
5			4	C	-0.0516	-0.0602	-0.0559	-0.0086	0.291465	2.058173	0.566717	0.573314	0.570016
6			5	C	0.1688	0.1612	0.165	-0.0076	0.284857	1.999467	0.393433	0.398172	0.395802
7			6	C	0.2102	0.2224	0.2163	0.0122	0.287434	1.948558	0.360826	0.349759	0.355293
8			7	H	0.0637	0.0671	0.0654	0.0034	0.27702	0.327542	0.308489	0.302755	0.305622
9			8	H	0.0553	0.0562	0.05575	0.0009	0.287474	0.330961	0.405863	0.401814	0.403839
10			9	H	0.0534	0.0532	0.0533	-0.0002	0.284322	0.331713	0.424469	0.429192	0.42683
11			10	H	0.0637	0.0672	0.06545	0.0035	0.273633	0.327542	0.308459	0.302787	0.305623
12			11	H	0.0553	0.0562	0.05575	0.0009	0.270863	0.330962	0.405809	0.401694	0.403751
13			12	H	0.0535	0.0532	0.05335	-0.0003	0.271063	0.331704	0.424203	0.428991	0.426597
14													
15													

Figure 13: Electronic chart with condensed local descriptors data for the atoms of the benzene molecule obtained with the finite difference method.

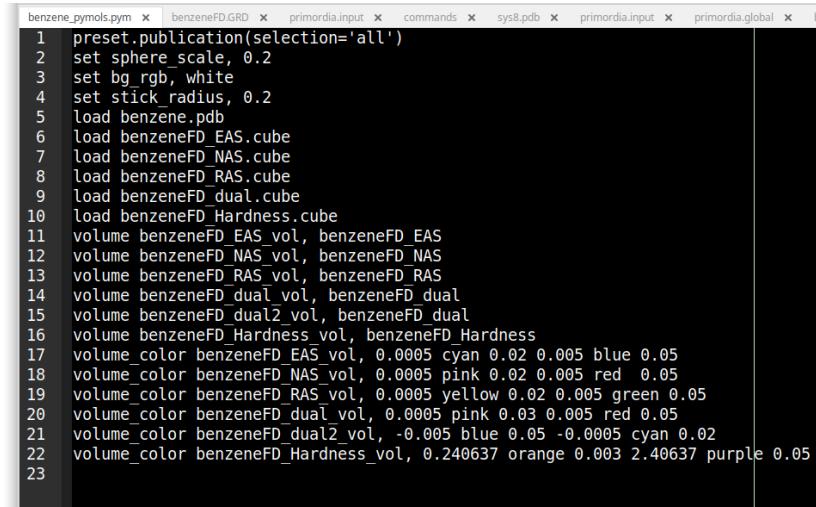
From the data displayed in Figure 13, we can observe the various local descriptors obtained with the PRIMoRDiA calculation, as well as properties extracted from the output file of computational chemistry programs, as in this case the *charge column*, showing the

partial charges for the atoms in the reference system. The descriptor called *Fukushima* is the location of the boundary orbitals in each atom, and therefore is not changed by finite difference methods.

Looking at the nucleophilicity and electrophilicity descriptors, we can see that there are negative values associated with some atoms. When these values are small compared to the most reactive atoms, there is nothing to worry about the quality of the calculations, since it is a characteristic of the finite difference method to indicate regions where the electron density decreases in the face of a charge gain process, or increase in the face of a process of loss of load. These variations can occur, as long as they are small, and must be interpreted with care within the context of the modeled process.

Still on these two descriptors, we can see that atoms that have high nucleophilicity also have high electrophilicity. Because it is symmetrical, benzene tends to show reactivity patterns well distributed in the carbon atoms and generally guided by the ring geometry. Therefore, the analysis of these numerical descriptors per atom can end up becoming confusing for this type of systems, and that is why it is more interesting to evaluate the local descriptors in the volumetric representation.

As in all the tutorials in this document, we are going to use the Pymol graphics software, taking advantage of the script written by PRIMoRDiA, which in the folder must have the name of *benzene_pymols.pym*, which can be viewed open in editor of text in Figure 14.



```
benzene_pymols.pym x | benzeneFD.GRD x | primordia.input x | commands x | sys8.pdb x | primordia.input x | primordia.global x | be
1  preset.publication(selection='all')
2  set sphere_scale, 0.2
3  set bg_rgb, white
4  set stick_radius, 0.2
5  load benzene.pdb
6  load benzeneFD_EAS.cube
7  load benzeneFD_NAS.cube
8  load benzeneFD_RAS.cube
9  load benzeneFD_dual.cube
10 load benzeneFD_Hardness.cube
11 volume benzeneFD_EAS.vol, benzeneFD_EAS
12 volume benzeneFD_NAS.vol, benzeneFD_NAS
13 volume benzeneFD_RAS.vol, benzeneFD_RAS
14 volume benzeneFD_dual.vol, benzeneFD_dual
15 volume benzeneFD_dual2.vol, benzeneFD_dual
16 volume benzeneFD_Hardness.vol, benzeneFD_Hardness
17 volume_color benzeneFD_EAS.vol, 0.0005 cyan 0.02 0.005 blue 0.05
18 volume_color benzeneFD_NAS.vol, 0.0005 pink 0.02 0.005 red 0.05
19 volume_color benzeneFD_RAS.vol, 0.0005 yellow 0.02 0.005 green 0.05
20 volume_color benzeneFD_dual.vol, 0.0005 pink 0.03 0.005 red 0.05
21 volume_color benzeneFD_dual2.vol, -0.005 blue 0.05 -0.0005 cyan 0.02
22 volume_color benzeneFD_Hardness.vol, 0.240637 orange 0.003 2.40637 purple 0.05
```

Figure 14: Text editor with the script file written by PRIMoRDiA to automate the visualization of volumetric descriptors in Pymol.

Through your terminal, with the path corresponding to the directory/folder of the tutorial files, open the Pymol window, and in your terminal run the following command

```
@benzene_pymols.pym
```

From this command, Pymol will load the .cube files and a PDB file also generated by PRIMoRDiA, all at once. The window will look like the one shown in ??.

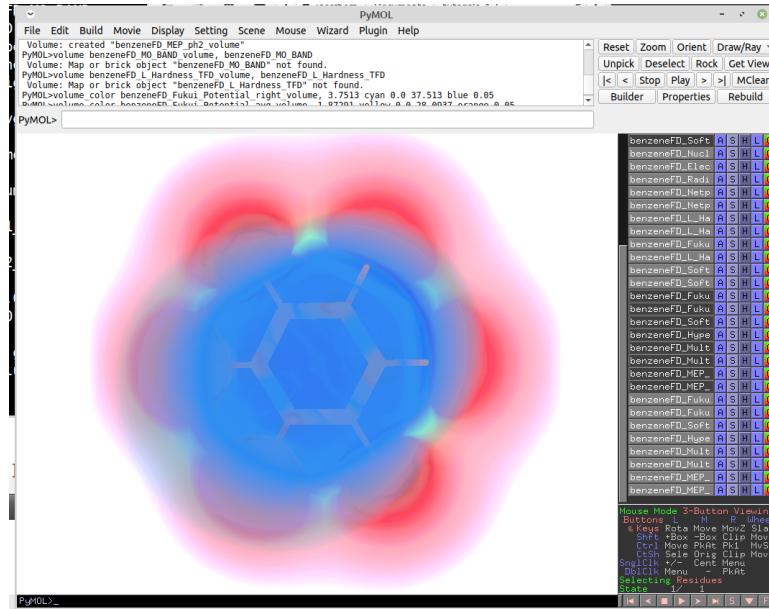


Figure 15: Pymol software with tutorial data loaded from the script generated in PRIMoRDiA.

The loaded objects are listed on the side of the window, and the ones that are brightest are those that are active and appearing on the screen. Select only one and to save the image of this representation in high resolution execute the following command

```
draw 3000,2800,antialias=2
```

After a few seconds, or minutes depending on the computing power and graphics cards available, this image appears highlighted on a black background, as in Figure 16. The image has been rendered, to save it permanently to disk, use the following command

```
png benzene_nuc.png
```

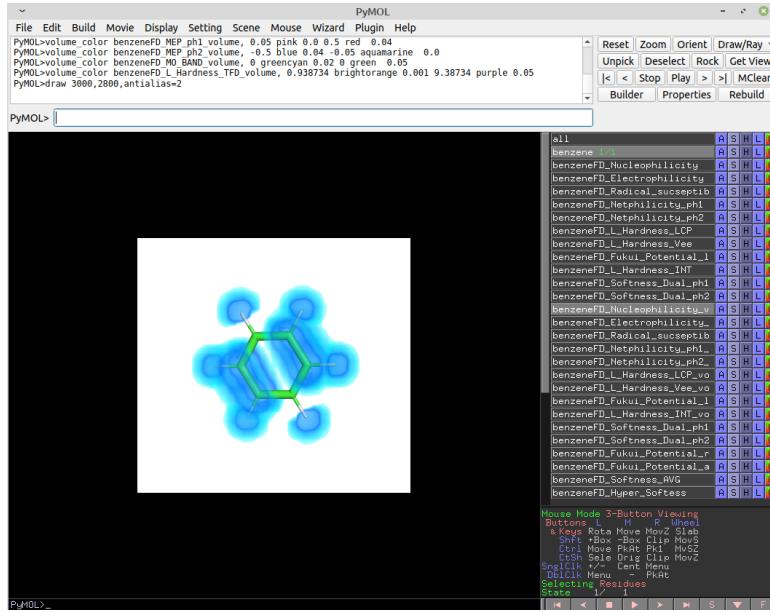


Figure 16: Image of the volumetric nucleophilicity descriptor rendered in Pymol software.

We can notice that this descriptor presents a very symmetrical distribution for the whole molecule, but not following the circular pattern, giving preference to four ring carbon atoms to the detriment of the other two.

An interesting descriptor to observe is the local chemical hardness, which in PRIMoRDiA can be obtained with up to seven different working equations, which are fully explained in the user guide. PRIMoRDiA does not put all possible local descriptors in the script, even so as not to overload the graphics software, because depending on the available computing resources, it can crash.

Also, for all descriptors, PRIMoRDiA suggests a color palette, so there is usually no need to manually adjust the values and opacities of iso-surfaces, as shown in previous tutorials. In any case, it is recommended that Pymol tutorials be carried out for the production of more customized images.

Returning to the local hardness descriptors, in Figure 17 we show the rendering of this descriptor calculated with the working equation that is based on the approximation of the local chemical potential, abbreviated LCP. In the figure Figure 18, the estimated local hardness with the approximation of the inter-electronic interaction part of the molecular electrostatic potential is shown. Another local property that represents hard-hard interactions, those controlled-by-charge as explained in the user guide, is the molecular electrostatic potential itself shown in Figure 19.

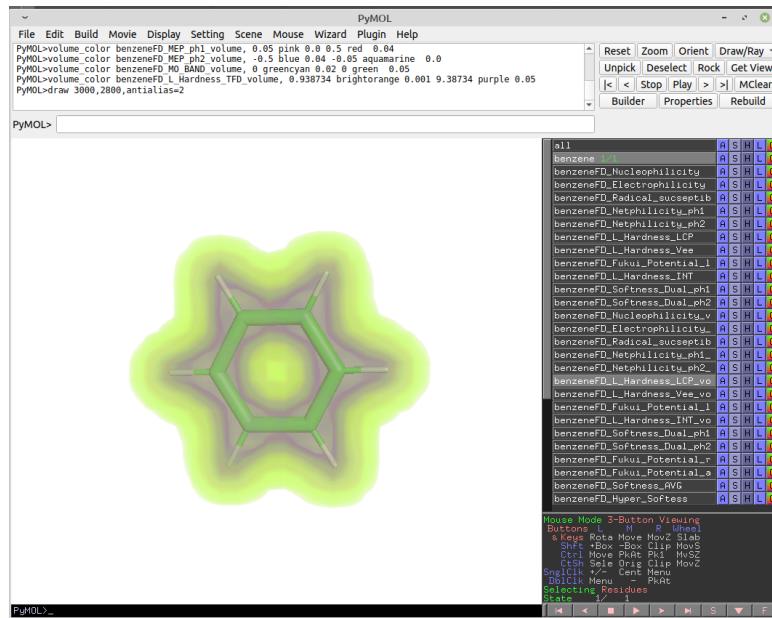


Figure 17: Local hardness using, local chemical potential approximation, rendered in Pymol.

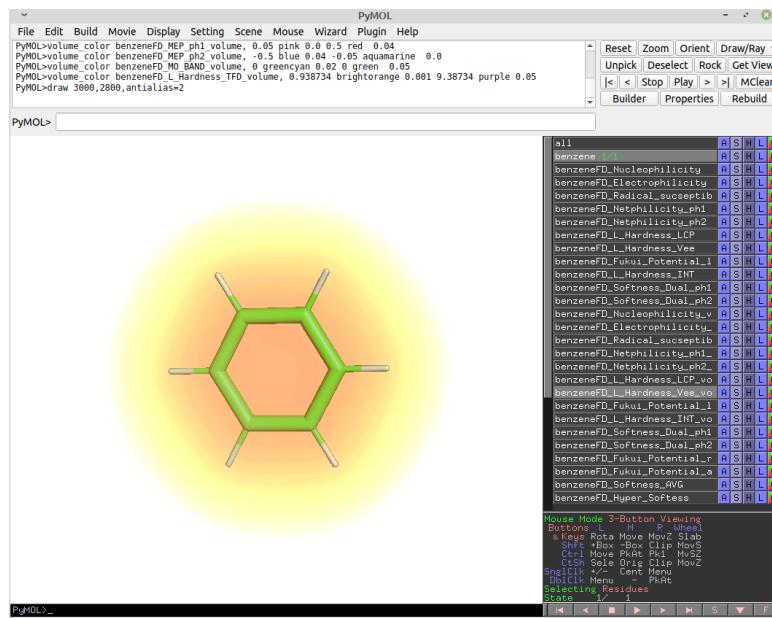


Figure 18: Local hardness using, inter-electronic interaction approximation of molecular electrostatic potential, rendered in Pymol.

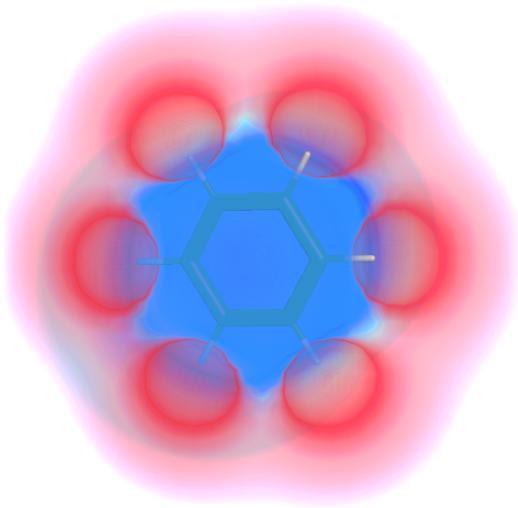


Figure 19: Molecular electrostatic potential estimated from partial charges. Image rendered and saved in png.

From these descriptors, we can see how the main interactions with electron density arise from the center of the aromatic ring.

3 Tutorial 3: Band Descriptors

As with the frozen orbital approximation, in this tutorial we will explore two methods of combining boundary orbitals for fitting descriptors for macromolecules. For the calculations, we will use a very common polypeptide in tests, the TRP-cage, whose electronic structure was obtained using semi-empirical methods implemented in MOPAC, including the MOZYME linear scaling method.

Topics covered in this tutorial:

- Calculation of descriptors using our boundary orbital banding methods.
- Use of the Pymol graphics package to visualize and generate high quality images for the condensed descriptors.
- Demonstration of descriptors condensed by residues
- Volumetric representation for these descriptors, especially for user-defined bounded regions
- Energy State Density Analysis

3.1 Context

O PRIMoRDiA é um programa que foi desenvolvido desde o início para o cálculo de descriptores de reatividade para grandes moléculas com relevância para sistemas biológicos, mais especificamente proteínas e fragmentos de DNA/RNA. Essas moléculas são biopolímeros e possuem uma estrutura eletrônica peculiar, com vários orbitais moleculares com energias similares as do HOMO e LUMO e com distribuição espacial relevante para processos químicos que envolvam esses sistemas.

3.2 File Preparation and Execution

The files needed to perform the tutorial are in the Tutorials_Files compressed folder of this repository. The files correspond to outputs from the mopac program of calculations made with the semi-empirical method PM7, one with the linear scaling method mozyme 2jof_PM7_lmo.aux, and without this method 2jof_PM7.aux. For each of these files there is a PDB file, which contains information relevant to the biopolymer that will be used by PRIMoRDiA to write descriptors and special representations. These PDBs are based on what can be found by the 2JOF code. The cartoon representation of this polypeptide is found below in Figure 20.

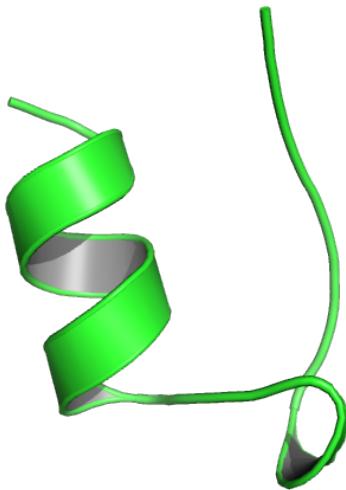


Figure 20: Cartoon representation of TRP-cage structure.

To facilitate the production of input from these files we will use the following command

```
computer@user:$ path/to/PRIMoRDiA_1.25v -input -op 3 -p mopac
```

A primordia.input call file is generated in the folder, just like the one shown in **??**. The first keyword found in the file is "#RT normal" to indicate that the Run Type is of the common type expected for PRIMoRDiA execution. The keyword "#PR" indicates that the next entries in the line are special parameters, which in this case, for now, are "eband" and "1", indicating that the limit to consider molecular orbitals in relation to HOMO-LUMO is 1eV. The two lines starting with the integer "3", indicate the special PRIMoRDiA calculation option for macromolecules.

Listing 4: Edited input for the tutorial execution.

```
#RT normal
#PR eband 1 extrard pymols
3 2jof_PM7.aux true 40 0 2jof_PM7.pdb mopac 0 0 0 0 10 EW
3 2jof_PM7_lmo.aux true 40 0 2jof_PM7_lmo.pdb mopac 0 0 0 0 EW
```

These arguments are explained in detail in the user guide, but they are also described one by one in the list below.

- "3" : Indicates the descriptor calculation option for PRIMoRDiA
- "2jof.aux": Name of the output file that contains electronic structure information
- "true" : Keyword for local hardness option to be calculated in

volumetric version

- "40" : Granularity of the cube files to be generated for the volumetric descriptors, zero indicates that these grid calculations will not be generated.
- "0" : Maximum number of molecular orbitals to be used by the BD method
- "2jof_PM7.pdb": PDB name with reference information
- "0" : Coordinate of the x axis for the center of the box to be generated for the volumetric descriptors
- "0" : Coordinate of the y-axis to the center of the box to be generated for the volumetric descriptors
- "0" : Coordinate of the z axis for the center of the box to be generated for the volumetric descriptors
- "0" : Size of the side of the box to be generated from the center given by the coordinates above
- "EW" : Keyword to signal the energy-weighted method of calculating descriptors for macromolecules

For the production of cube files, with volumetric data, it is possible to limit a calculation area to obtain a larger definition with a lower computational cost. For this it is necessary to define a box where the grid will be calculated, where the center and the size are given in the input file, as indicated above.

PRIMoRDiA has implemented two methods to combine these molecular orbitals:

1. Band Density (BD: acronym and Keyword used within the program)
2. Energy weighted (EW: acronym and Keyword used within the program)

The first one is inspired by the electronic density calculation, with the particularity of considering only a selection of molecular orbitals from HOMO, for the occupied ones, and LUMO for the virtual ones. The second is a combination that is weighted by the difference in energy of the molecular orbitals of the HOMO-LUMO boundary orbitals. There is a threshold for considering these orbitals which is defined by a user defined power cutoff value.

Let's edit the input file so that the first input runs with the BD method and the second runs with a box defined for the volumetric descriptors. In this example we will define the center of the box in the nitrogen coordinates of the side chain of residue TRP_6, information taken from the PDB file shown in Figure 21.

primordia.input	x	2jof.pdb	x
72	ATOM	71	HE21 GLN A 5 21.493 -0.087 -3.479 1.00 20.00 A H
73	ATOM	72	HE22 GLN A 5 21.487 -0.274 -5.238 1.00 20.00 A H
74	ATOM	73	N TRP A 6 26.657 1.026 -0.296 1.00 20.00 A N
75	ATOM	74	CA TRP A 6 27.939 0.765 0.370 1.00 20.00 A C
76	ATOM	75	C TRP A 6 27.748 0.150 1.763 1.00 20.00 A C
77	ATOM	76	O TRP A 6 28.319 -0.901 2.056 1.00 20.00 A O
78	ATOM	77	CB TRP A 6 28.769 2.060 0.434 1.00 20.00 A C
79	ATOM	78	CG TRP A 6 30.176 1.872 0.902 1.00 20.00 A C
80	ATOM	79	CD1 TRP A 6 31.203 1.507 0.110 1.00 20.00 A C
81	ATOM	80	CD2 TRP A 6 30.744 2.042 2.239 1.00 20.00 A C
82	ATOM	81	CE2 TRP A 6 32.119 1.664 2.192 1.00 20.00 A C
83	ATOM	82	CE3 TRP A 6 30.248 2.491 3.482 1.00 20.00 A C
84	ATOM	83	NE1 TRP A 6 32.335 1.325 0.876 1.00 20.00 A N
85	ATOM	84	CZ2 TRP A 6 32.957 1.710 3.316 1.00 20.00 A C
86	ATOM	85	CZ3 TRP A 6 31.084 2.567 4.614 1.00 20.00 A C
87	ATOM	86	CH2 TRP A 6 32.434 2.181 4.532 1.00 20.00 A C
88	ATOM	87	H TRP A 6 26.531 1.932 -0.742 1.00 20.00 A H
89	ATOM	88	HA TRP A 6 28.488 0.036 -0.224 1.00 20.00 A H
90	ATOM	89	HB2 TRP A 6 28.799 2.512 -0.558 1.00 20.00 A H
91	ATOM	90	HB3 TRP A 6 28.292 2.767 1.107 1.00 20.00 A H
92	ATOM	91	HD1 TRP A 6 31.132 1.354 -0.957 1.00 20.00 A H
93	ATOM	92	HE1 TRP A 6 33.220 0.985 0.488 1.00 20.00 A H
94	ATOM	93	HE3 TRP A 6 29.219 2.809 3.550 1.00 20.00 A H
95	ATOM	94	HZ2 TRP A 6 33.993 1.416 3.234 1.00 20.00 A H

Figure 21: TRP-cage PDB file opened in text editor.

Let's manually modify the input with this coordinate data and box size 10 angstroms, with the input looking as shown in the Listing 5 text box.

Listing 5: Edited input for the tutorial execution

```
#RT normal
#PR eband 1 extrard pymols
3 2jof_PM7.aux true 40 0 2jof_PM7.pdb mopac 32.335 1.325 0.876 10 EW
3 2jof_PM7_1mo.aux true 40 10 2jof_PM7_1mo.pdb mopac 0 0 0 0 BD
```

Finally, with this input we can start the program with the following calculation

```
computer@user:$ path/to/PRIMoRDiA_1.25v -f primordia.input
```

The program should finish its execution without errors and produce several files in the directory, with the extension ".cube", ".pym", ".lrd" and etc. We will explain the files with the results of descriptors and how to analyze them in next section of this tutorial.

3.3 Analysis Results

From the results produced we have the global descriptors, local in condensed representation (".lrd" files), local in volumetric representation (".cube" files) and condensed locations written in PDB (in "RD_PDB" folders). For macromolecules, it is interesting to observe the global descriptors more for the sake of monitoring the normality of the quantum calculation performed, since the processes in these systems tend to be only local, with little relevance to the propensity of an entire protein to transfer or receive electrons. That's why we're not going to analyze the "primordia.global" file.

For this stable release of the program, PRIMoRDiA 1.25v, the main

modification is the consolidation of the analysis automation scripts and production of publication quality results, using the Pymol graphics package and the R statistical package. Therefore, in the next steps of the tutorial, we will focus on producing images from these scripts.

3.3.1 Condensed descriptors

Local descriptors are spatial coordinate-dependent mathematical functions, describing trends from a certain point in molecular space. In the case of condensed descriptors, these values are partitioned for each atomic center, simplifying the interpretation of the results, and in the case of large systems, making the visualization clearer, since the volumetric representation for protein systems and the like can be quite polluted. .

However, this type of representation can be quite complicated to work with, as it is a list of values by atoms, that is, a table that can be significantly large for macromolecules. In order to facilitate the analysis of these results, PRIMoRDiA writes a PDB file for each descriptor calculated with the values in the b-factor field, and so we can use the Pymol graphics package to visualize the descriptors by coloring the atoms with a custom palette. To operate Pymol as automatically as possible, PRIMoRDiA writes a script file with the suffix "`_pymols_pdb.pym`", with an example of its structure shown in Figure 22.

```

1 preset.publication(selection='all')
2 set sphere_scale, 0.2
3 set bg_rgb, white
4 set stick_radius, 0.18
5 load 2jof_PM7_PDB_RD/2jof_PM7_nucleophilicity.pdb
6 load 2jof_PM7_PDB_RD/2jof_PM7_electrophilicity.pdb
7 load 2jof_PM7_PDB_RD/2jof_PM7_radicality.pdb
8 load 2jof_PM7_PDB_RD/2jof_PM7_netphilicity.pdb
9 load 2jof_PM7_PDB_RD/2jof_PM7_hardness_Vee.pdb
10 load 2jof_PM7_PDB_RD/2jof_PM7_hardness_lcp.pdb
11 load 2jof_PM7_PDB_RD/2jof_PM7_fukui_pot_left.pdb
12 load 2jof_PM7_PDB_RD/2jof_PM7_fukui_pot_right.pdb
13 load 2jof_PM7_PDB_RD/2jof_PM7_fukui_pot_zero.pdb
14 load 2jof_PM7_PDB_RD/2jof_PM7_softness_dual.pdb
15 load 2jof_PM7_PDB_RD/2jof_PM7_hyper_softness.pdb
16 load 2jof_PM7_PDB_RD/2jof_PM7_fukushima.pdb
17 load 2jof_PM7_PDB_RD/2jof_PM7_mep.pdb
18 load 2jof_PM7_PDB_RD/2jof_PM7_hardness_TFD.pdb
19 load 2jof_PM7_PDB_RD/2jof_PM7_softness_avg.pdb
20 load 2jof_PM7_PDB_RD/2jof_PM7_hardness_int.pdb
21 load 2jof_PM7_PDB_RD/2jof_PM7_multiphilicity.pdb
22 load 2jof_PM7_PDB_RD/2jof_PM7_charge.pdb
23 load 2jof_PM7_PDB_RD/2jof_PM7_electron_density.pdb
24 spectrum b, blue_white_red, minimum=-0.3, maximum=0.3
25 spectrum b, white_yellow_orange_red_black, minimum=0.1, maximum=0.5
26 spectrum b, white_cyan_blue, minimum=0, maximum=0.1
27 spectrum b, white_pink_red, minimum=0, maximum=0.1
28

```

Figure 22: Automation script for the visualization of local descriptors condensed in Pymol written in PRIMoRDiA and opened in a text editor.

This file has commands to run in Pymol modifying view parameters and to load PDBs with descriptors. In addition, it already runs some color palettes, one of which overlaps the other, but they are

used for the user to pick and choose one or modify its parameters and colors. Open your Pymol, run the command shown below

```
@2jof_PM7_pymols_pdb.pym
```

After executing this command, the pymol window should open and load all PDB files and execute all commands that configure the color palette based on the descriptor values. The last "spectrum b..." command executed should be the one that appears, as in Figure 23.

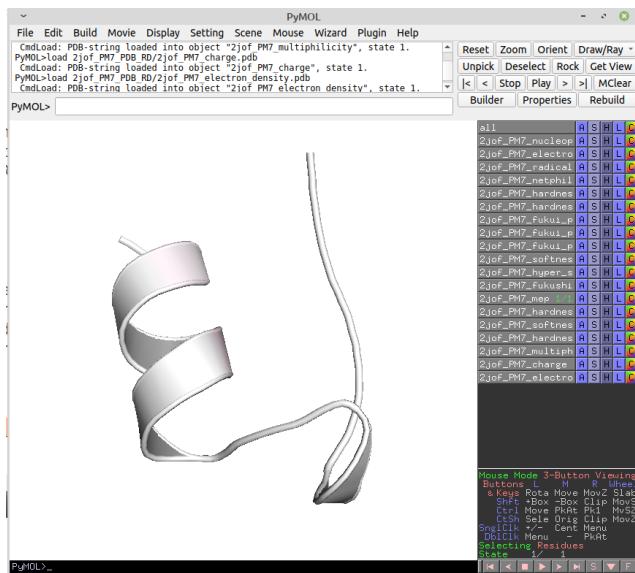


Figure 23: Condensed local descriptors for the TRP-cage structure loaded into Pymol.

In the next image we show Pymol with only one of the objects active, 2jof_PM7_nucleophilicity, and showing the command to be executed in the terminal, which was selected/copied from one of those that was executed when the script was called.

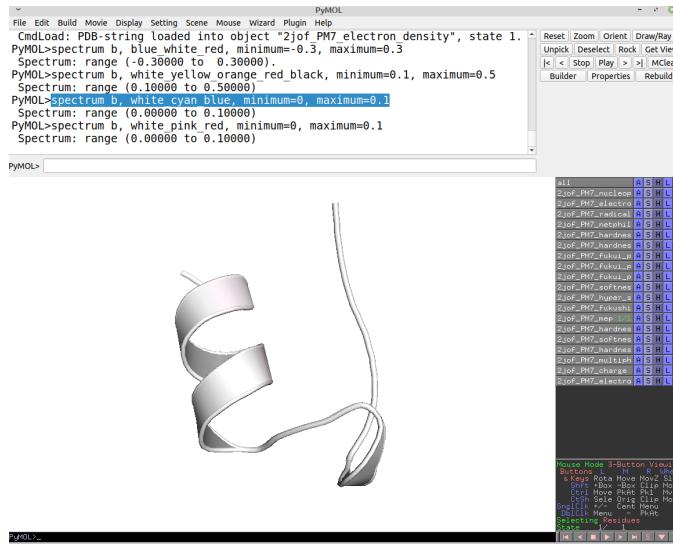


Figure 24: Pymol window showing the command that has already been executed.

After executing this command in Pymol the result is as shown Figure 25, where the atoms of some side chains are selected, since only the secondary structure is in evidence, with the next command in Pymol the selected atoms appear in representation of canes.

show sticks, sele

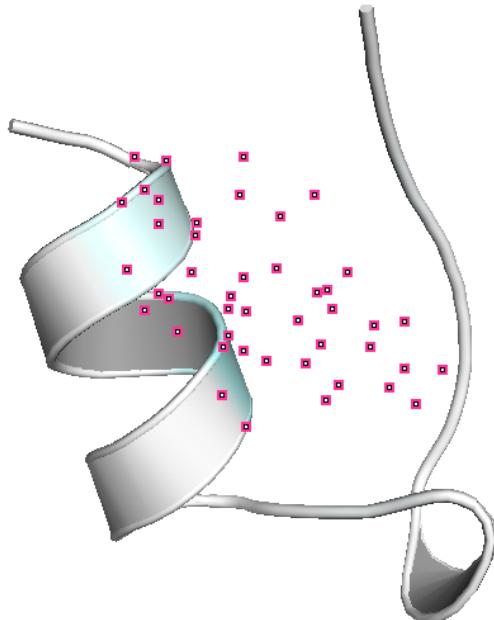


Figure 25: Structure in Pymol with two sidechains selected.

The atoms and bonds between them will be shown in the representa-

tion of sticks, resulting in the figure Figure 26, showing the painted structure based on the values of susceptibility to an electrophilic attack, the same as nucleophilicity. Where the blue color is darker is where the values are higher, it is possible to change these maximum and minimum values as well as the colors as well. In the figure below you can see that the command "ray" was executed in Pymol. Type the same and execute, after a few seconds a transparent background appears, meaning that the image has already been rendered and can be saved with the "png" command, as already used for volumetric descriptors in previous tutorials.

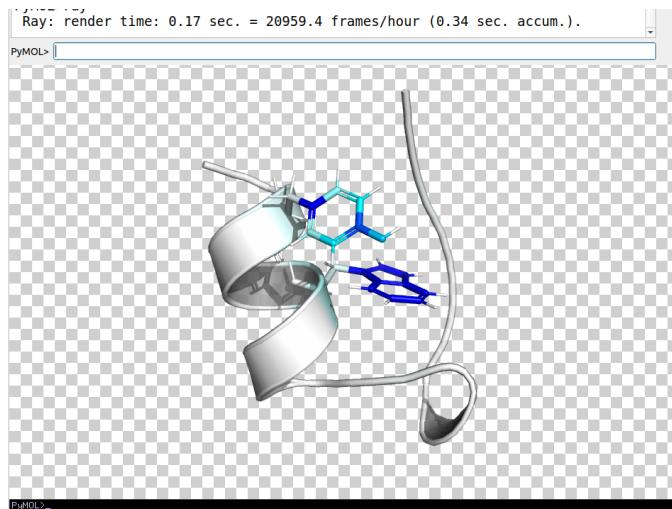


Figure 26: Condensed nucleophilicity local descriptor representation for the TRP-cage structure, with transparency background, rendered and ready to be saved to file.

This process of coloring atoms, rendering and saving these images can be repeated for all other descriptors. For local hardness descriptors it is interesting to use other *ray tracing* modes. For the next image shown, the 2jof_PM7_hardness_B object was activated in Pymol, and the following commands were executed in Pymol

```
set ray_trace_mode=1
show sticks
set spectrum b,white_yellow_orange_red_black,minimum=0.1,maximum=0.2
ray
png 2jof_hardneess.png
```

The image saved on disk should be similar to the one shown in Figure 27, showing the local hardness for the TRP-cage structure.

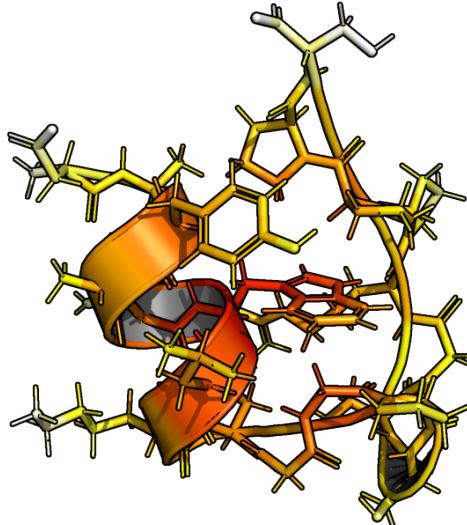


Figure 27: Image saved for local hardness in condensed representation generated in Pymol.

3.3.2 Volumetric Representation

Descriptors with volumetric representation follow the same visualization logic as the other last two tutorials. The difference, in the case of specific calculations for macromolecules, through the BD and EW methods, is in the selection and combination of molecular orbitals. To automate the visualization of the basic descriptors in volumetric representation, PRIMoRDiA writes a script similar to the one shown in Figure 28, which is for the 2jof_PM7_lmo.aux entry. For the other entry, 2jof_PM7.aux, a script like this was also written, which after being executed opens the structures and renders the cubes in Pymol as in Figure 29.

```

preset.publication(selection='all')
set sphere_scale, 0.2
set bg_rgb, white
set stick_radius, 0.18
load 2jof.PM7_lmo.pdb
load 2jof.PM7_lmoFOA_HOMO_ph1.cube
load 2jof.PM7_lmoFOA_HOMO_ph2.cube
load 2jof.PM7_lmoFOA_LUMO_ph1.cube
load 2jof.PM7_lmoFOA_LUMO_ph2.cube
load 2jof.PM7_lmoFOA_Nucleophilicity.cube
load 2jof.PM7_lmoFOA_Electrophilicity.cube
load 2jof.PM7_lmoFOA_Radical_susceptibility.cube
load 2jof.PM7_lmoFOA_Netphilicity_ph1.cube
load 2jof.PM7_lmoFOA_Netphilicity_ph2.cube
load 2jof.PM7_lmoFOA_L_Hardness_LCP.cube
load 2jof.PM7_lmoFOA_L_Hardness_Vee.cube
load 2jof.PM7_lmoFOA_Fukui_Potential_left.cube
load 2jof.PM7_lmoFOA_L_Hardness_INT.cube
load 2jof.PM7_lmoFOA_Softness_Dual_ph1.cube
load 2jof.PM7_lmoFOA_Softness_Dual_ph2.cube
volume 2jof.PM7_lmoFOA_HOMO_ph1.volume, 2jof.PM7_lmoFOA_HOMO_ph1
volume 2jof.PM7_lmoFOA_HOMO_ph2.volume, 2jof.PM7_lmoFOA_HOMO_ph2
volume 2jof.PM7_lmoFOA_LUMO_ph1.volume, 2jof.PM7_lmoFOA_LUMO_ph1
volume 2jof.PM7_lmoFOA_LUMO_ph2.volume, 2jof.PM7_lmoFOA_LUMO_ph2
volume 2jof.PM7_lmoFOA_Nucleophilicity.volume, 2jof.PM7_lmoFOA_Nucleophilicity
volume 2jof.PM7_lmoFOA_Electrophilicity.volume, 2jof.PM7_lmoFOA_Electrophilicity
volume 2jof.PM7_lmoFOA_Radical_susceptibility.volume, 2jof.PM7_lmoFOA_Radical_susceptibility
volume 2jof.PM7_lmoFOA_Netphilicity_ph1.volume, 2jof.PM7_lmoFOA_Netphilicity_ph1
volume 2jof.PM7_lmoFOA_Netphilicity_ph2.volume, 2jof.PM7_lmoFOA_Netphilicity_ph2
volume 2jof.PM7_lmoFOA_L_Hardness_LCP.volume, 2jof.PM7_lmoFOA_L_Hardness_LCP
volume 2jof.PM7_lmoFOA_L_Hardness_Vee.volume, 2jof.PM7_lmoFOA_L_Hardness_Vee
volume 2jof.PM7_lmoFOA_Fukui_Potential_left.volume, 2jof.PM7_lmoFOA_Fukui_Potential_left
volume 2jof.PM7_lmoFOA_L_Hardness_INT.volume, 2jof.PM7_lmoFOA_L_Hardness_INT
volume 2jof.PM7_lmoFOA_Softness_Dual_ph1.volume, 2jof.PM7_lmoFOA_Softness_Dual_ph1
volume 2jof.PM7_lmoFOA_Softness_Dual_ph2.volume, 2jof.PM7_lmoFOA_Softness_Dual_ph2

```

Figure 28: Part of the automation script for the visualization of volumetric local descriptors in Pymol written in PRIMoRDiA and opened in a text editor.

In Figure 29 we went ahead and selected only the phases of the netphilicity descriptor, remembering that all descriptors that have negative and positive values are written in two identical files, but with different names, so that Pymol can render them separately. Also, in the same image you can see that it already has some side chains already selected, which are the ones that you can see that the volumes are rendered over their atoms.

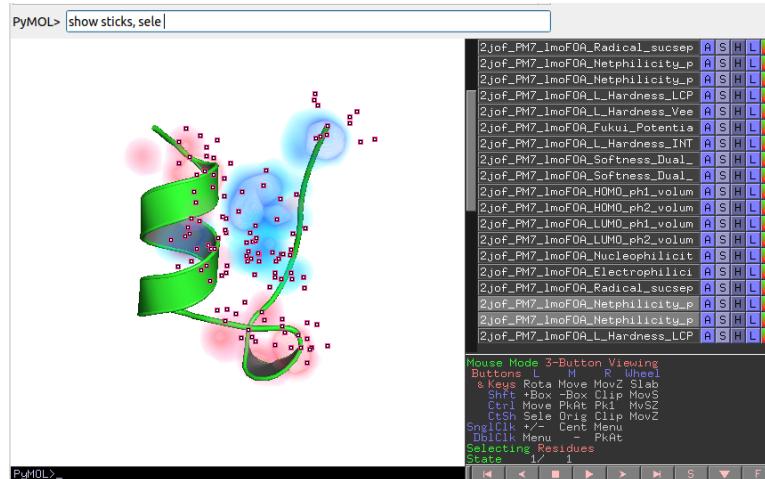


Figure 29: Pymol window showing the volumetric local descriptors loaded by the script generated in PRIMoRDiA for the TRP-cage structure with some selected residues.

In the Pymol command area shown in Figure 29 appears the command so that the representation of sticks is activated for these selected strings. After this command is executed the sticks appear

as shown in Figure 30. The *draw* command can be used to render and then save the image to disk with the *png* command, as already exemplified in tutorials 1 and 2.

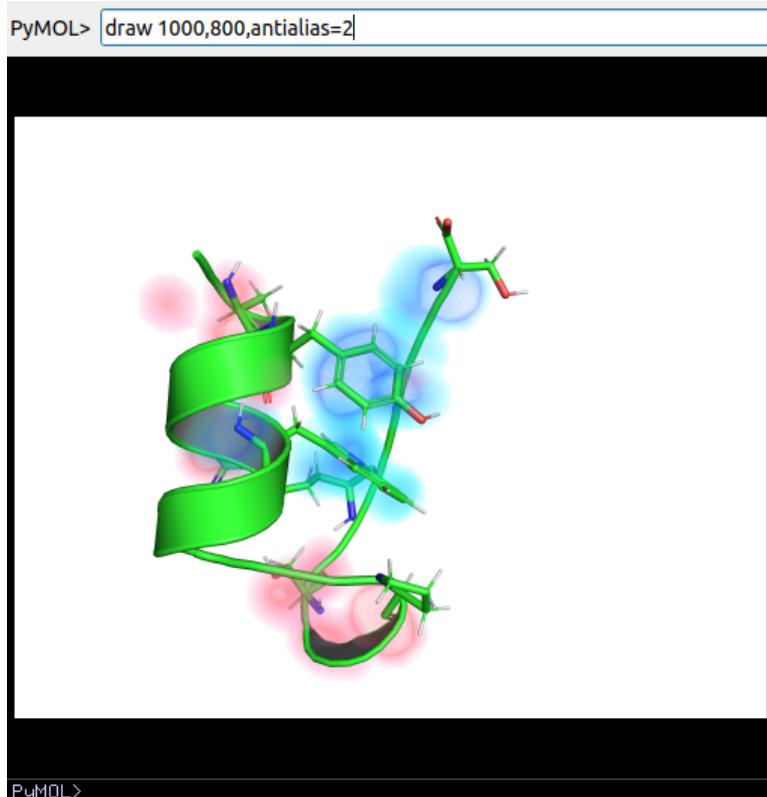


Figure 30: Pymol window showing the volumetric local descriptors loaded by the script generated in PRIMoRDiA for the TRP-cage structure with some residues shown in stick representation.

Now let's view the results of the calculated descriptors for the box defined in the input file, running the Pymol script corresponding to that input. The result should be as shown in Figure 31, loading and rendering various descriptors for the bounded region.

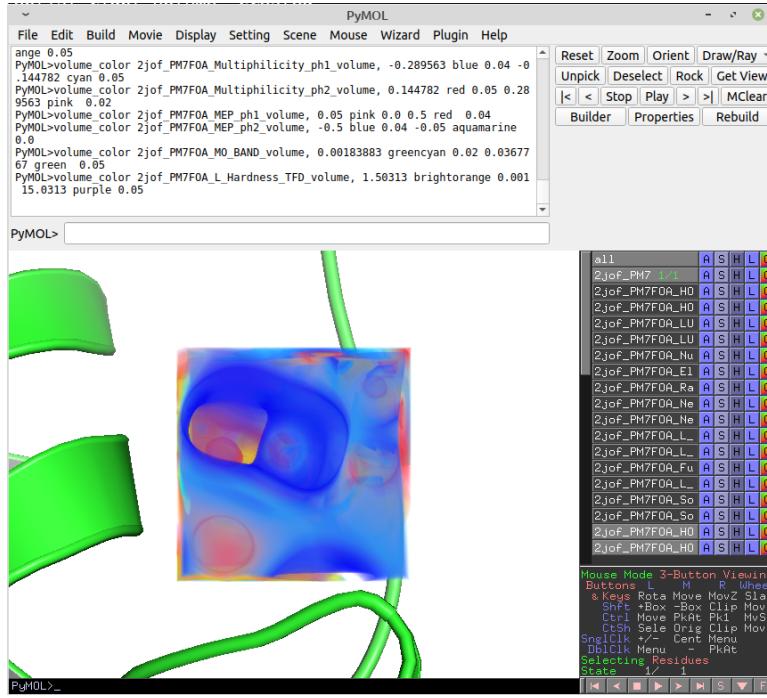


Figure 31: Pymol window showing the volumetric local descriptors loaded by the script generated in PRIMoRDiA for a specific cubic region of the TRP-cage.

If we turn off all volume objects, we are left with only the cube objects loaded, which in the viewer appear marking the edges of the cube's boundaries. In Figure 32, we show this threshold in red, with a different color to draw attention.

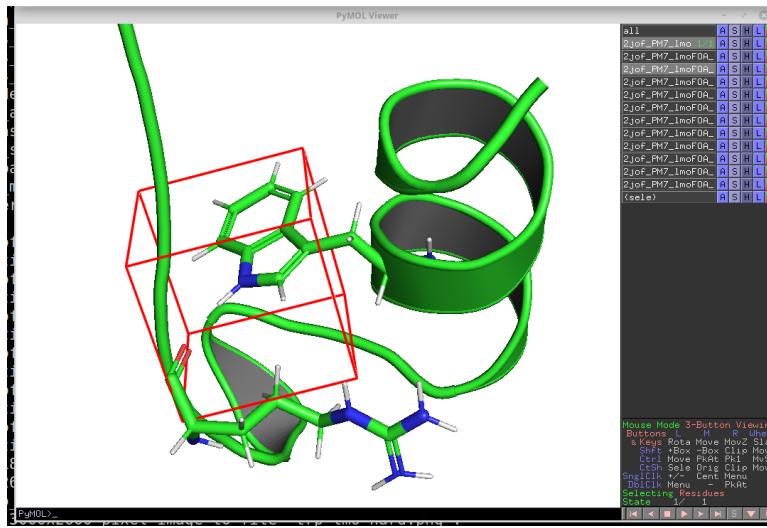


Figure 32: .

In relation to other volumetric descriptors, those defined within these spaces limited by the user, do not present any practical difference. The issue here is that with systems with a large number

of atoms it is possible to visualize the descriptors for specific regions without losing resolution and that can be calculated in a practicable time. In the next figures are presented some descriptors rendered for this delimited region and already saved in publication quality within Pymol. In Figure 33, the local hardness descriptor using the approximation based on the local chemical potential. In Figure 33, we show this volume rendered with a different palette, better highlighting regions that have higher values than others.

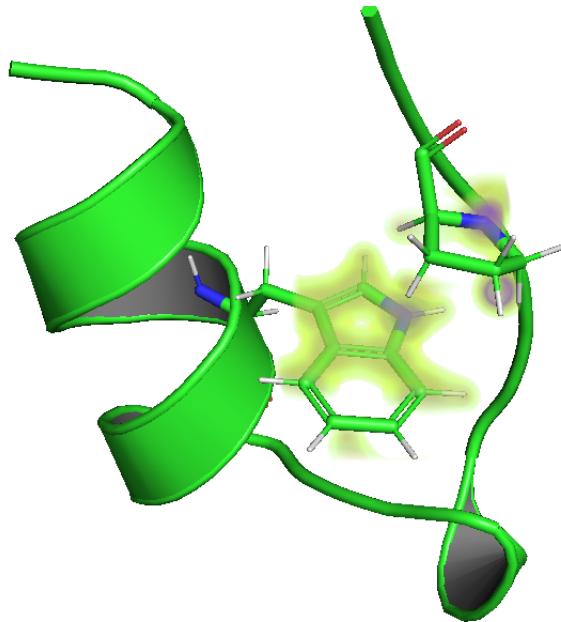


Figure 33: Example descriptor image rendered for user-bounded region. Local hardness based on approximation of local chemical potential.

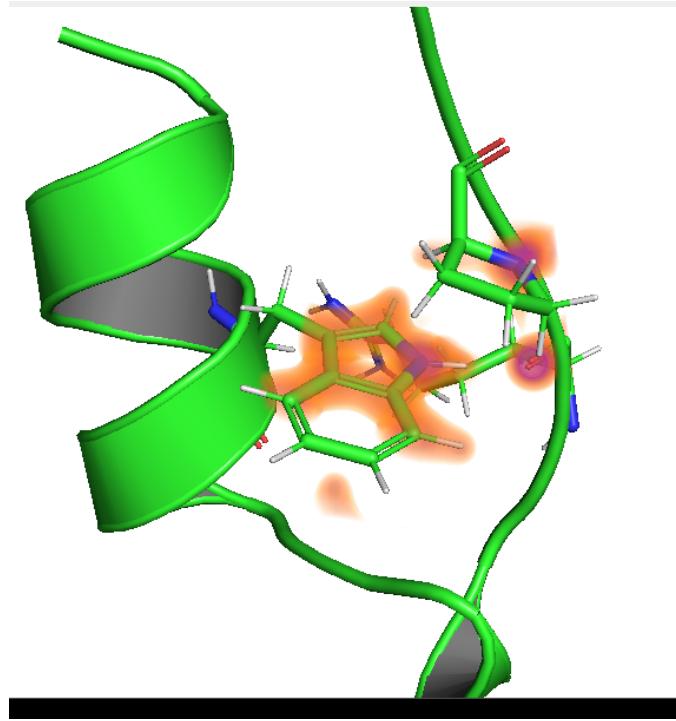


Figure 34: Example descriptor image rendered for user-bounded region. Local hardness based on approximation of local chemical potential.

In Figure 35, the last image is enlarged and the distance between the hydrogen of the side chain and the oxygen of the residue of the main chain is evidenced, showing that in this region where there is clearly a hydrogen bond also presents the highest values of the descriptor of local hardness in question.

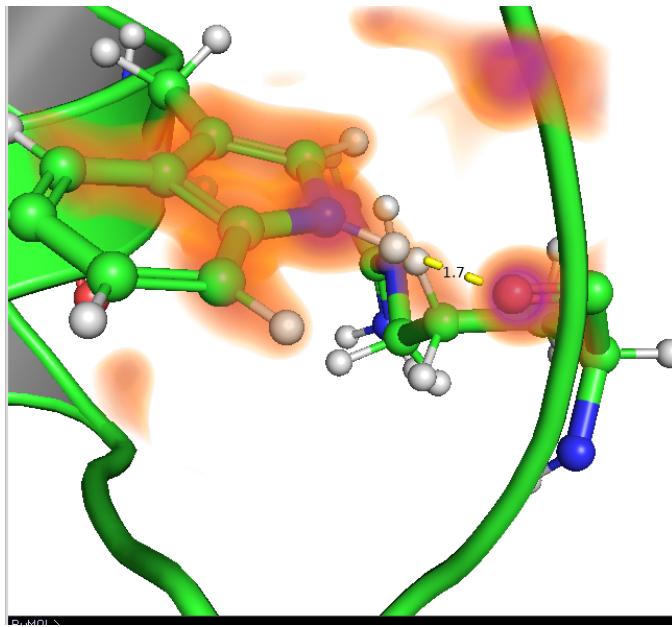


Figure 35: Example descriptor image rendered for user-bounded region. Enlarged image for the region of hydrogen bonding interaction, showing the distance between atoms. Local hardness based on approximation of local chemical potential.

3.4 Spacial representation for residues and the Density of State

For biological macromolecules there are several analyzes focused on properties and statistics depending on the monomers that compose them. For the most part, these systems are proteins that are composed of fragments of amino acids, called amino acid residues. In the same way that we have already presented the condensed descriptors for atoms, it is interesting to analyze these values for each of these fragments of the system. PRIMoRDiA uses the information from the PDB to add the descriptors of atoms per residue and write it to a formatted file, as in Figure 36.

```

1 #res Nucleophilicity Electrophilicity Radicals Hardness_Vee Hardness_LCP Fukui_pot_left Fukui_pot_right Fukui_pot_zero softn
2 1ASP 0.000047 0.196202 0.098124 0.196155 1.572429 1.246886 3.831282 4.933386 -0.048172 0.005918 -0.443072 0.693494 -0.073080 43.48
3 2ALA 0.001107 0.195596 0.098352 0.194490 1.332675 0.833347 3.695541 4.433898 4.064720 -0.047763 0.005932 -0.439310 0.692051 0.064300 29.061
4 3TYR 0.660673 0.587201 0.623937 -0.073472 2.941187 1.770429 12.965459 10.502487 11.733973 0.018043 0.037629 0.165957 2.509603 0.022630 62.0
5 4ALA 0.000527 0.086474 0.043501 0.085948 1.364701 0.826317 3.610692 4.180204 3.895448 -0.021107 0.002623 -0.194137 0.305985 -0.013820 28.81
6 5GLN 0.001350 0.022433 0.011891 0.021083 2.307037 1.433682 5.614325 6.609253 6.111789 -0.005178 0.000717 -0.047622 0.080175 -0.006120 49.99
7 6TRP 2.325697 0.384173 1.354935 -1.941524 3.811355 1.953870 21.420384 12.47379 16.928882 0.476799 0.081715 4.385480 2.886220 -0.001400 69.
8 7LEU 0.001024 0.074981 0.038003 0.073957 2.697134 1.398316 9.588930 8.048403 8.186666 -0.018162 0.002292 -0.167052 0.265689 -0.028690 48.76
9 8LYS 0.000008 0.084905 0.042456 0.084896 2.724491 1.424280 6.479512 7.440169 6.959840 -0.020849 0.002561 -0.191762 0.300096 0.978680 49.667
10 9ASP 0.000004 0.109826 0.054915 0.109822 1.719865 1.363234 4.209618 5.265296 4.737457 -0.026970 0.003312 -0.248064 0.388178 -0.927560 45.44
11 10GLY 0.000168 0.099295 0.049732 0.099127 0.994898 0.630461 2.698183 3.119944 2.909064 -0.024344 0.002999 -0.223907 0.351063 -0.041440 21.9
12 11GLY 0.001816 0.074693 0.038254 0.072878 1.103087 0.601506 4.380177 3.545539 3.962856 -0.017897 0.002307 -0.164615 0.265193 0.089660 20.97
13 12PRO 0.0001395 0.111834 0.056614 0.110439 1.788980 1.114722 6.485771 5.592681 6.039226 -0.027122 0.003414 -0.249459 0.396187 -0.019270 38.8
14 13SER 0.000006 0.147275 0.073641 0.147261 1.371789 0.993846 3.558035 4.428792 3.993414 -0.036166 0.004441 -0.332648 0.520539 -0.037140 34.6
15 14SER 0.000001 0.181425 0.090713 0.181424 1.546528 1.007916 4.110005 5.188317 4.649161 -0.044554 0.005471 -0.409798 0.641238 -0.039740 35.1
16 15GLY 0.000000 0.094519 0.047258 0.094518 0.889732 0.641020 2.528992 2.992615 2.760803 -0.023212 0.002850 -0.213496 0.334070 0.027620 22.35
17 16ARG 0.0000451 0.481992 0.241222 0.481541 3.371486 1.748639 10.318387 11.524630 10.921508 -0.118257 0.014548 -1.087696 1.703870 0.945780 60
18 17PRO 0.000529 0.027119 0.053180 1.714908 1.065170 5.489696 4.814029 5.151863 -0.013060 0.001636 -0.120122 0.190180 0.017260 37.14
19 18PRO 0.004618 0.010071 0.007345 0.005453 1.931782 1.104373 8.498590 5.456805 6.977697 -0.001339 0.000443 -0.012317 0.038631 0.018800 38.51
20 19PRO 0.009572 0.0063030 0.001801 0.002457 1.935335 1.107264 6.870249 5.435734 6.152992 -0.006604 0.000109 -0.005551 0.011083 -0.033250 38.6
21 20SER 0.000006 0.000364 0.000185 0.000358 1.249890 1.147167 3.736653 3.080400 3.408526 -0.000088 0.000011 -0.000808 0.001290 -0.942470 40.0
22 Residues_Average 0.150000 0.150000 0.150000 -0.000000 1.918465 1.167622 6.504524 5.951498 6.228011 -0.000000 0.000946 -0.000000 0.628742 0.0
23

```

Figure 36: File with local descriptors condensed by residue (amino acid fragment) for the TRP-cage structure opened in a text editor.

This data can be easily selected/copied and transferred to an excel/libreoffice table, as exemplified in Figure 37. For a given trajectory, from molecular dynamics or reaction coordinates, values for specific residues can be gathered and analyzed during these processes. More on this is exemplified in tutorial 5.

#res	Nucleophilicity	Electrophilicity	Radicality	Netphilicity	Hardness_Vo	Hardness_LO	Fukui_pot_lep	Fukui_pot_rif	Fukui_pot_zep	softness	dual hyp
1ASP	4.7E-05	0.196202	0.098124	0.196155	1.572429	1.246886	3.831282	4.933386	4.382334	-0.048172	
2ALA	0.001107	0.195596	0.098352	0.19449	1.332675	0.833347	3.695541	4.433898	4.06472	-0.047763	
3TYR	0.660673	0.587201	0.623937	-0.073472	2.941187	1.770429	12.965459	10.502487	11.733973	0.018043	
4ALA	0.000527	0.086474	0.043501	0.085948	1.364701	0.826317	3.610692	4.180204	3.895448	-0.021107	
5GLN	0.00135	0.022433	0.011891	0.021083	2.307037	1.433682	5.614325	6.609253	6.111789	-0.005178	
6TRP	2.325697	0.384173	1.354935	-1.941524	3.811355	1.95387	21.420384	12.437379	16.928882	0.476799	
7LEU	0.001024	0.074981	0.038003	0.073957	2.697134	1.398316	9.58893	8.048403	8.818666	-0.018162	
8LYS	8E-06	0.084905	0.042456	0.084896	2.724491	1.42428	6.479512	7.440169	6.95984	-0.020849	
9ASP	4E-06	0.109826	0.054915	0.109822	1.719865	1.303234	4.209618	5.265296	4.737457	-0.02697	
10GLY	0.000168	0.099295	0.049732	0.099127	0.994898	0.630461	2.698183	3.119944	2.909064	-0.024344	
11GLY	0.0001816	0.074693	0.038254	0.072878	1.103087	0.601506	4.380177	3.545539	3.962858	-0.017897	
12PRO	0.001395	0.111834	0.056614	0.110439	1.78898	1.114722	6.485771	5.592681	6.039226	-0.027122	
13SER	6E-06	0.147275	0.073641	0.147269	1.371789	0.993846	3.558035	4.428792	3.993414	-0.036166	
14SER	1E-06	0.181425	0.090713	0.181424	1.546528	1.007916	4.110005	5.188317	4.649161	-0.044554	
15GLY	0	0.094519	0.047259	0.094518	0.889732	0.64102	2.528992	2.992615	2.760803	-0.023212	
16ARG	0.000451	0.481992	0.241222	0.481541	3.371486	1.748639	10.318387	11.52463	10.921508	-0.118257	
17PRO	0.000529	0.053709	0.027119	0.05318	1.714908	1.06517	5.489696	4.814029	5.151863	-0.01306	
18PRO	0.004618	0.010071	0.007345	0.005453	1.931782	1.104373	8.49859	5.456805	6.977697	-0.001339	
19PRO	0.000572	0.00303	0.001801	0.002457	1.935335	1.107264	6.870249	5.435734	6.152992	-0.000604	
20SER	6E-06	0.000364	0.000185	0.000358	1.24989	1.147167	3.736653	3.0804	3.408526	-8.E-05	
Residues Av		0.15	0.15	0.15	0	1.918465	1.167622	6.504524	5.951498	6.228011	0

Figure 37: Electronic spreadsheet with local descriptors condensed by residue (amino acid fragment) for the structure of the TRP-cage.

Through the analysis of these numerical data, it is possible to indicate the reactivity of biological systems by their residues, which is very useful when defining important domains for enzymes, such as active site, binding site, allosteric site, etc. Furthermore, these data tables are rich in information for machine learning algorithms, with electronic properties calculated by quantum methods for the systems' monomers.

One of PRIMoRDiA's features is to write an energy table and a script to be run in R to produce a density graph of energy states. The analysis of these data is interesting for biopolymers, due to the high concentration of energy levels with energy similar to that of HOMO and LUMO. This script is produced when the keyword "dos" is used in the input parameters field and can be useful to evaluate the number of maximum orbitals used or the power cutoff value for counting molecular orbitals.

In the folder of this tutorial, after the input has been executed, there should be files ending in "DOS.R" and "DOS". To generate an image automatically with the script, just run the command

```
computer@user:$ Rscript 2jof_PM7_1mo_DOS.R
```

And in the folder should appear an image like the picture below

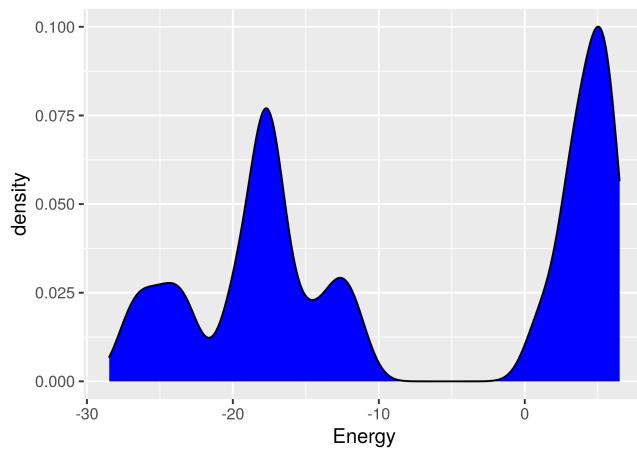


Figure 38: Energy state density graph for TRP-Cage molecular orbitals calculated with PM7 together with MOZYME algorithm.

And the command below creates the figure of the density of states for the TRP calculated normally with PM7

```
computer@user:$ Rscript 2jof_PM7_DOS.R
```

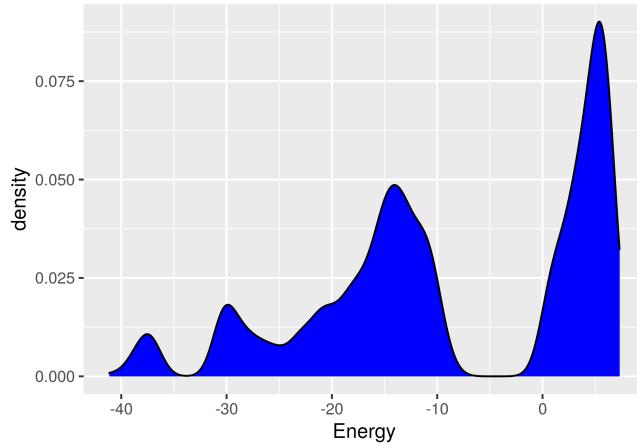


Figure 39: Energy state density graph for TRP-Cage molecular orbitals calculated with PM7.

We can observe that the energy values of the orbitals calculated with the MOZYME algorithm are more concentrated than those that did not use this calculation acceleration strategy. This makes the HOMO-LUMO gap larger in this first case, which ends up changing a lot when choosing the molecular orbitals to match when calculating the descriptors. Therefore, the ideal for new systems is whenever possible to analyze this graph, which can easily be produced with the help of PRIMoRDiA.

4 Tutorial 4: Acid strength analysis

This tutorial uses the same approach and logic for interpreting the results of Tutorial 1, but focused on something of greater interest in day-to-day chemistry. Like that first one. We will demonstrate how to use the program to obtain the reactivity descriptors using the information from the boundary molecular orbitals, option 1 of the PRIMoRDiA calculation, starting from semi-empirical calculations in MOPAC, for four amino acids, generating reactivity descriptors that can be used to explain the preferred sites of protonation.

4.1 Context

Amino acids are amphoteric organic molecules, that is, they can act as acids and bases, due to the amine and carboxylic acid groups. In addition, some have side chains that are considered to be acids or bases, and when they are in the composition of a protein, this acidity/basicity will be relevant for various processes and properties.

Acid strength is measured by pKa, which is the negative of the log of the acid strength constant. In short, when the pH is lower than the pKa of a group in the molecule, this group tends to be protonated, otherwise without this proton. That is, the lower the pKa, the more acidic and the higher the more basic. However, pKa measurements change with the chemical environment and obtaining them theoretically presents several challenges when the systems involved are proteins.

Descriptors can be useful to identify sites on molecules for preferred sites for electron transfer. More specifically, local hardness is the ideal descriptor for proton transfer, which is a small charged particle and therefore dominated by interactions of an electrostatic nature. In this tutorial, we will calculate the descriptors for a cysteine, aspartic acid, glutamic acid, and a lysine, and cross-reference with pKa and ionization potential data.

4.2 Required Files and Execution

All the files necessary for the execution can be found in the compressed folder provided with the program, and they are the files that contain the output information of the MOPAC program for the four amino acids studied. As well as an input file with the same lines of text as shown below.

Listing 6: Input for tutorial 4.

```
#RT normal
#PR pymols
1 asp.aux true 40 mopac
1 cys.aux true 40 mopac
1 glu.aux true 40 mopac
1 lys.aux true 40 mopac
```

As already shown in previous tutorials, after having the input in the folder where the files are, just run the program with the "-f" flag. After that the files with the results must be created in the folder.

4.3 Results Analysis

The results for this tutorial are very similar to those obtained in tutorial 1, and we used the same calculation option to generate the descriptors with approximation of frozen orbitals. In Figure 40 the text file with the results of the global descriptors is shown, in Figure 41 the same data transferred to a Libre Office spreadsheet.

```
primordia.global × aspFOA_EAS.cube × primordia.input × commands ×
1 Molecule Energy Ecat Ean HOF IP EA ECP Hardness Softness Electrophilicity nMax gap
2 asp -7998.93000000 0.00000000 0.00000000 -174.71400000 4.86000000 -3.10000000 -0.88000000 3.98000000
3 cys -5694.33000000 0.00000000 0.00000000 -38.59950000 8.83400000 0.49400000 -4.66400000 4.17000000 0
4 glu -9264.45000000 0.00000000 0.00000000 -157.49800000 4.03300000 -2.06700000 -0.98300000 3.05000000
5 lys -9454.76000000 0.00000000 0.00000000 -61.88870000 8.41100000 0.62000000 -4.51550000 3.89550000 0
6
```

Figure 40: Text file with the results of the global descriptors for the calculated amino acids.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Molecule	Energy	Ecat	Ean	HOF	IP	EA	ECP	Hardness	Softness	Electrophilicity	nMax	gap
2	asp	-7998.93	0	0	-174.714	4.86	-3.1	-0.88	3.98	0.25125628	0.09728643	0.44221106	7.96
3	cys	-5694.33	0	0	-38.5995	8.834	0.494	-4.664	4.17	0.23980815	2.60826091	2.23693046	8.34
4	glu	-9264.45	0	0	-157.498	4.033	-2.067	-0.983	3.05	0.32786885	0.15840803	0.64459016	6.1
5	lys	-9454.76	0	0	-61.8887	8.411	0.62	-4.5155	3.8955	0.25670646	2.61708898	2.31831601	7.791

Figure 41: Spreadsheet with the results of the global descriptors for the calculated amino acids.

What can be extracted from these results is that for the same electronic structure method, lysine and aspartic acid molecules have very similar chemical hardness, cysteine with a much higher hardness than all of them and glutamic acid with the lowest. In fact, glutamic acid has two oxygens that generally act as proton acceptors and/or participate in nucleophilic attacks in enzymatic environments. However, as much as cysteine appears to be harder in these results, it is the amino acid with the greatest ability to receive electrons along with lysine.

As for the local descriptors, for each calculated molecule a ".lrd" file was generated, as explained in previous tutorials, how to explore

and analyze them in spreadsheets. Data from the most appropriate local descriptors to assess acid strength will be gathered later for proton acceptor/acceptor atoms.

To help qualitatively in the analysis, we can generate the volumetric descriptors in the Pymol graphics package. This can be easily accomplished using the scripts generated by PRIMORDiA, as we have seen in other tutorials. Open Pymol and run the ".pymols" script in the program's terminal. The result should be similar to the one shown in Figure 42, with the cube objects loaded, the volume objects generated and selected.

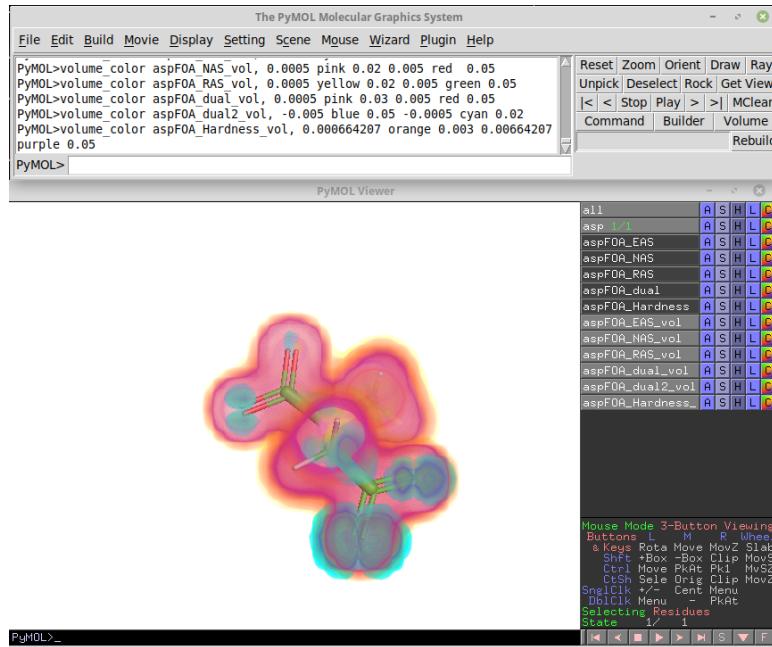


Figure 42: Basic local descriptors in volumetric representation for aspartic acid, rendered in the Pymol program.

In Figure 43 the Netphilicity descriptor, or the two phases of the dual descriptor, is shown for aspartic acid. With the command "Draw", in Pymol it is possible to render the image to save it in the desired resolution and in high quality, resulting in Figure 44. In this image it is possible to see the surfaces rendered in blue are located in oxygen atoms, and represents the tendency of these places to receive an electrolytic attack, and therefore they are nucleophilic, and in pink and red places where they tend to receive a nucleophilic attack and therefore are electrophiles. For aspartic acid, almost all of the carboxylic acid oxygens appear to be nucleophilic, but with a clear preference for one of the side chain oxygens. Meanwhile, in the amine group it appears to be an electrophile, with a spherical disposition of the descriptor around the group, signaling that any proton can be removed so that the electron pair of this positively charged nitrogen is restored.

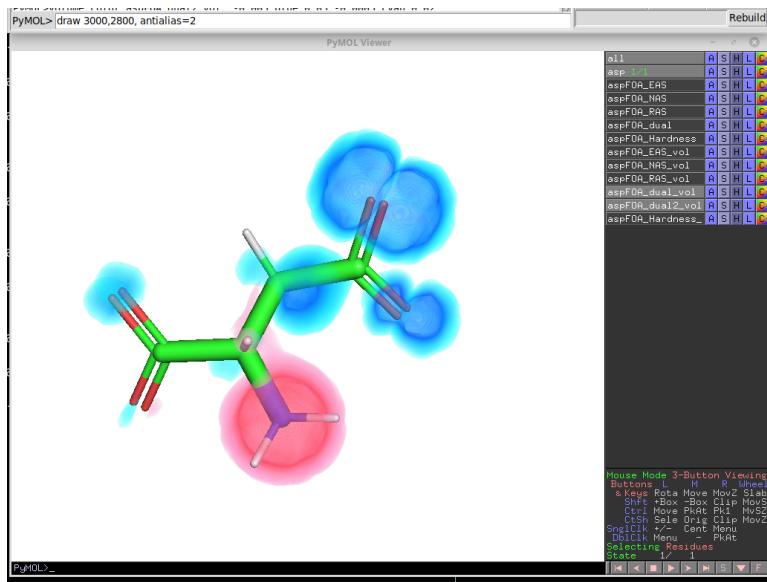


Figure 43: Netphilicity(dual) descriptor shown in Pymol for aspartic acid molecule, with terminal command to render image in high quality.

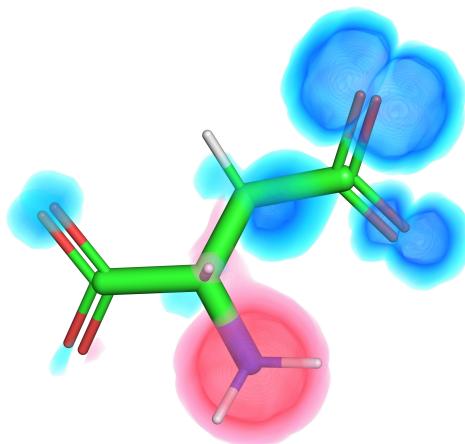


Figure 44: Image save to disk for the Netphilicity(dual) descriptor for the aspartic acid molecule.

Moving on to the analysis of the hardness descriptor, just deactivate the previous volume objects in Pymol and activate the last one, "aspFOA_Hardness_vol", a volumetric representation of the descriptor should appear as shown in Figure 45

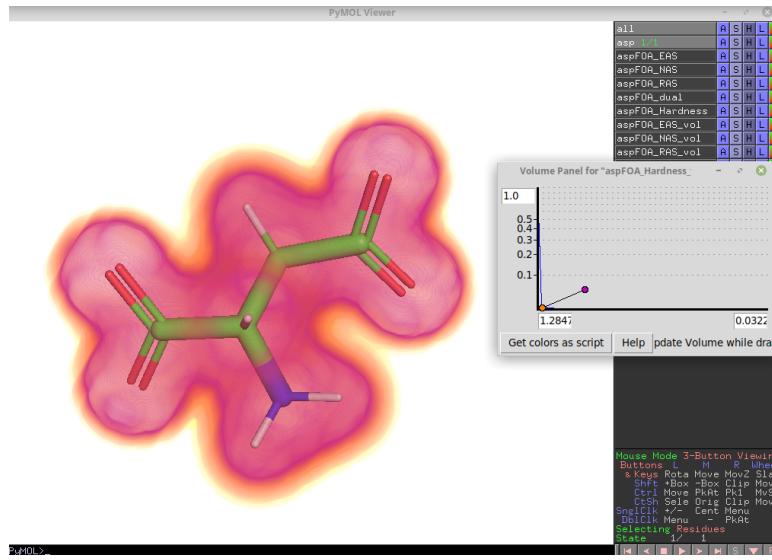


Figure 45: Local hardness descriptor for aspartic acid, showing the Pymol volume panel where it is possible to create and fit iso-surfaces.

The window that appears next to Figure 46 is opened by placing "volume" in the panel next to the Pymol command execution area. By clicking and moving the points in this window it is possible to change the colors, the values of the iso-surfaces and their opacities. It is also possible to add more iso-surfaces. The figure below shows a modification made to the local hardness descriptors.

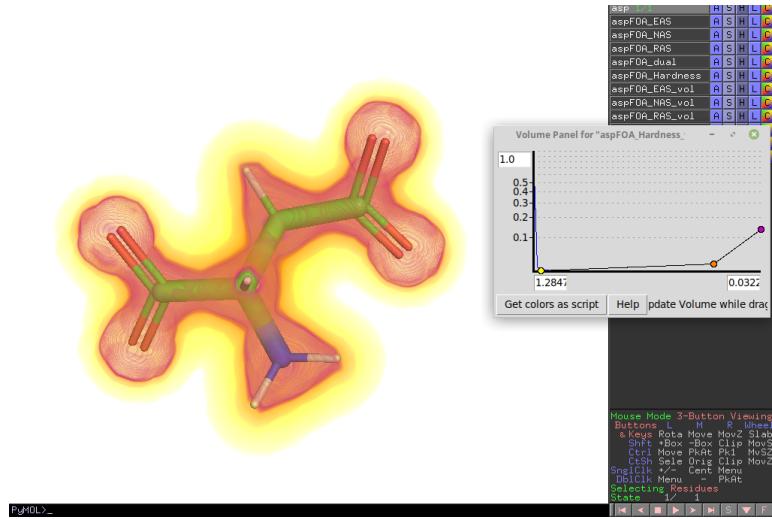


Figure 46: Local hardness descriptor for aspartic acid, showing Pymol volume panel with new suggested values for iso-surfaces for better visualization.

Repeating the process of rendering and saving the images for the other systems, we obtain the descriptors for four systems, as shown in the next two figures: Netphilicity in Figure 47 and hardness in Figure 48.

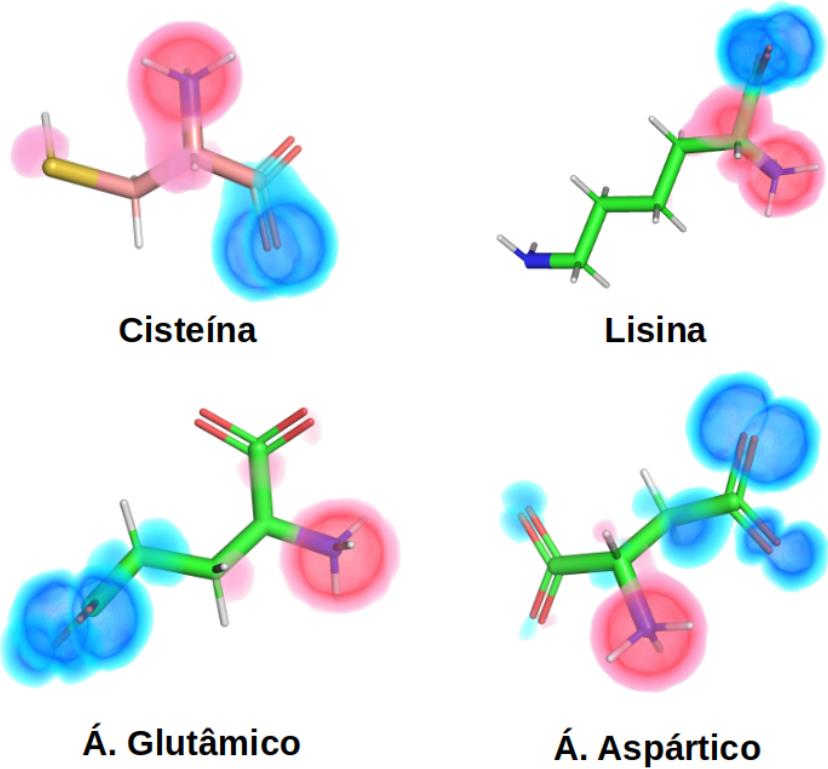


Figure 47: Netphilicity local descriptor for the four amino acids analyzed in the tutorial.

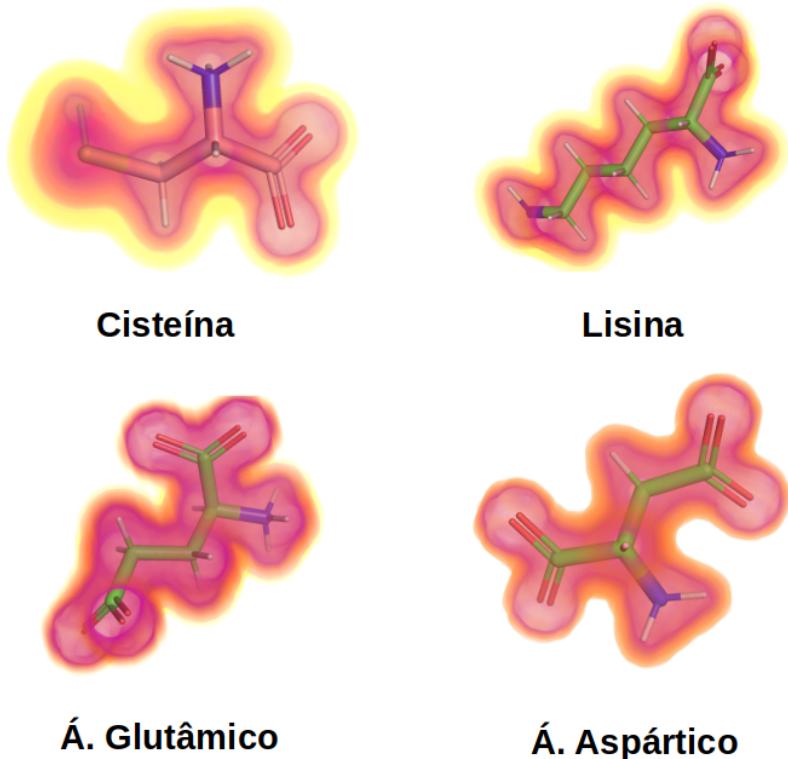


Figure 48: Local hardness descriptor for the four amino acids analyzed in the tutorial, using the local chemical potential method.

	-CO ₂ H	-NH ₂	sidechain	pI
Aspartic Acid	1.99	9.90	3.90	2.98
Cysteine	1.92	10.70	8.37	5.15
Glutamic Acid	2.10	9.47	4.07	3.08
Lysine	2.16	9.06	10.54	9.47

Figure 49: Table with the results of local descriptors for the main chain proton acceptor/donor groups of each of the four amino acids.

	Nucleofilicidade		Eletrofilicidade	Dureza			
	O1	O2	N1	N1	O1	O2	
Aspartic Acid	0.0524	0.8118	0.3003	2.6830	3.6419	3.1563	
Cysteine	0.3138	0.6704	0.2332	2.9926	3.7783	3.5915	
Glutamic Acid	0.0633	0.0011	0.3071	1.7035	2.2454	1.5802	
Lysine	0.3193	0.6697	0.2687	1.6476	1.9888	2.1211	
	carga parcial			global			
	N1	O1	O2	ECP	Hardness	IP	EA
Aspartic Acid	-0.0728	-0.5939	-0.6224	-0.88	3.98	4.86	-3.1
Cysteine	-0.0516	-0.4869	-0.5251	-4.664	4.17	8.834	0.494
Glutamic Acid	-0.0727	-0.5556	-0.5533	-0.983	3.05	4.033	-2.067
Lysine	-0.0738	-0.4896	-0.5603	-4.5155	3.8955	8.411	0.62

Figure 50: Table with the results of local descriptors for the side chain proton acceptor/donator atoms of each of the four amino acids. Also a complementary table with the global descriptors, partial charges and the known value of Ionization Potential.

Putting together the information that we can observe from the condensed and volumetric descriptors these theoretical quantities are able to locate the portions of the molecules where proton transfers are most likely to occur. The chemical hardness descriptor is the most appropriate for this type of process, transfer of small charged groups, and showed a relationship with the pKa of the amine groups, with cysteine showing the highest hardness in nitrogen and also the highest pKa. Obviously, this small study is fraught with limitations and is only intended to show a practical relationship between the results obtained with PRIMoRDiA and a common molecular modeling problem.

5 Tutorial 5: Enzymatic Reaction Analysis

This tutorial will show a practical example of how PRIMoRDiA can be used to study biological systems and their main processes. The great strength of PRIMoRDiA is in the modified descriptors for macromolecules that it makes available and in the automation of the processing of these electronic properties. We will enter the electronic structure of a fragment of an enzymatic system for a small reaction trajectory and we will leave with a statistical summary of the evolution of the descriptors in this process.

5.1 Context

Enzymatic reactions are chemical reactions catalyzed by proteins, which cause chemical transformations to occur in a timely manner for the functioning of the metabolism of living organisms. Therefore, enzymes are very important components for the maintenance of life, participating from DNA replication to the decomposition of sugar.

The study of the mechanism of these reactions is very important for drug development, understanding of metabolism, disease mechanisms and replication of microorganisms and industrial applications of these enzymes. In one of the studies of our group, it was with the mechanism of the reaction catalyzed by triose-phosphate-isomerase (TIM), more specifically the step shown in the scheme below in Figure 51.

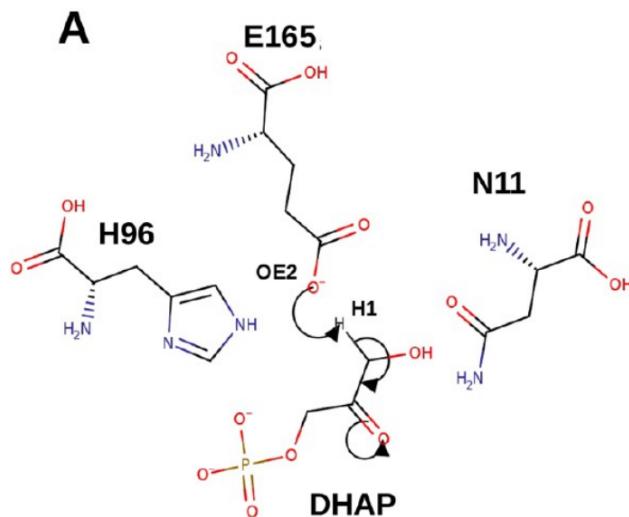


Figure 51: Representation of the enzymatic catalysis mechanism studied in this tutorial.

We used this enzyme to test the effectiveness of the descriptors in theoretically characterizing this reaction, describing the main forces, interactions and effects of the environment. TIM is one of the most characterized enzymes in the literature, it is a central enzyme

in the glycolytic pathway of interconversion of D-glyceraldehyde 3-phosphate (GAP) to dihydroxyacetone phosphate, with great efficiency.

In this tutorial we are going to calculate the descriptors for this system on a reaction trajectory obtained by QM/MM simulation.

5.2 Required Files and Execution

The files needed for the calculations of this tutorial are in the . These are ".aux" and ".pdb" files, for the reaction path steps. This reaction path only encompasses 110 atoms, done this way to get lighter files for the tutorial. The scan trajectory was obtained as explained in the easyhybrid tutorial.

To do the special analysis for the enzymatic reaction we go to another execution method, which is signaled in PRIMoRDiA by the keyword "reaction" after "#RT" in the first line, that is, the Run Type switches to reaction mode. In the folder you already have the input file, which now has several important elements to quote, and is shown in the figure below

It is possible to notice that there are several other lines with previously unused keywords and only one ".aux" file entry to run with the calculation option 3. As this is a calculation for a chemical reaction, PRIMoRDiA understands that entry as valid for all auxs in the folder that have the suffix flagged, in this case in our input it was the suffix "sys", running from "sys0.aux" to "sys14 .aux", obeying the "dimX" parameter.

Let's analyze the other parameters of the input

Listing 7: Input for the tutorial execution.

```
#RT reaction
#PR eband 2 Rscript pymols
#Reaction RC1 98 107 51
#Reaction dimX 15
#TRJ residues 2 4 7
3 sys.aux none 30 10 sys.pdb mopac
```

| 0 0 0 0 EW |

Second line

- Second line
 - #PR: flag to signal that parameters will be given to PRIMoR-DiA
 - eband: keyword signaling that the next parameter is a cut-off energy value to consider molecular orbitals
 - Rscript: Important keyword to generate the descriptor analysis scripts during the reaction trajectory

- pymols: keyword for output of scripts for Pymol
- Third line
 - #Reaction: flag to signal that parameters will be passed for reaction coordinate analysis
 - RC1: keyword signaling that the number of atoms that make up the reaction coordinate will be passed
 - 98: GAP carbon (C1) atom number
 - 107: number of hydrogen atom (H1) attached to DHAP
 - 51: oxygen atom number (OE2) of GLU
- Fourth Line
 - #Reaction: flag to signal that parameters will be passed for reaction coordinate analysis
 - dimX : keyword to signal that the next value in the line gives the number of steps in the trajectory
 - 15 : number of steps in the trajectory
- Fifth Line
 - #TRJ: flag to signal that parameters will be passed to deal with the set of frames that make up the reaction trajectory
 - residues : keyword indicating that the next integer values passed in this line are the number of residues in the PBD that you want to analyze for the given trajectory.
 - 2: catalytic histidine PDB number
 - 4: number in the catalytic glutamate PDB
 - 7: number in the PDB of the residue representing the substrate.
- Sixth Line
 - 3 : PRIMoRDiA calculation option for macromolecules
 - sys.aux: base name for the files, before the dot the prefix without the number indicating the frame, and after the dot the file extension
 - none : keyword signaling that no local hardness methods were requested to calculate the volumetric representation, which would no longer be done since the grid value is zero
 - 0 : grid fineness for volumetric representation, always leave it at zero until a new software update
 - 10 : maximum number of boundary molecular orbitals to be used if the orbital matching method is BD

- sys.pdb: base name for pdb files, same logic for files containing the electronic structure
- mopac : name of the program from which the electronic structure was calculated
- 0 : parameter not relevant with grid 0
- 0 : parameter not relevant with grid 0
- 0 : parameter not relevant with grid 0
- 0 : parameter not relevant with grid 0
- EW : Keyword for the molecular orbital matching method

5.3 Results Analysis

To analyze all the descriptors and other energy properties, PRIMoRDiA creates a script to run in R which in turn produces graphs of these quantities varying as a function of the frames. After the execution, files with ".R" suffix should appear, one of them named "sys0_reaction_analysis.R" and another "sys0_residuos_analysis.R". Let's run the first one with Rscript, as in the command below

```
computer@user:$ Rscript sys0\reaction\_analysis.R
```

After this command, R should produce several image files with global and local properties for each atom of the reaction coordinate. The figure below shows one of these files with global properties.

```
primordia.input x cys.aux x sys6.aux x
1 #RT reaction
2 #PR eband 2 Rscript pymols
3 #Reaction RC1 98 107 51
4 #Reaction dimX 15
5 #TRJ residues 2 4 7
6 3 sys.aux none 0 10 sys.pdb mopac 0 0 0 0 EW
7
```

Figure 52: File containing the global descriptors generated by PRIMoRDiA.

In the first graph of Figure 53 HOF is the heat of formation that Mopac calculates, and it is one of the best ways of changing energy in a chemical reaction simulated with a semi-empirical method. Remembering that all these properties are in reference to the first value, that is, each point is a variation to the initial state. In these simulations, a structure that is normally desired to be determined is the transition structure, which is determined by the maximum point of energy variation, in this case in the penultimate step. However, in our calculation with PM6, the quantum area was much smaller than the one used in our publication and the HOF was not a good indicator of the transition state. Electronic energy, on

the other hand, shows a first maximum at the zero point of the reaction coordinate, which is the moment when the proton is the same distance from carbon and oxygen. In general, the other global descriptors indicate an increase in chemical softness (softness) and consequent decrease in hardness (hardness), increase in chemical potential and total electrophilicity indicate that the system tends to receive electron transfer.

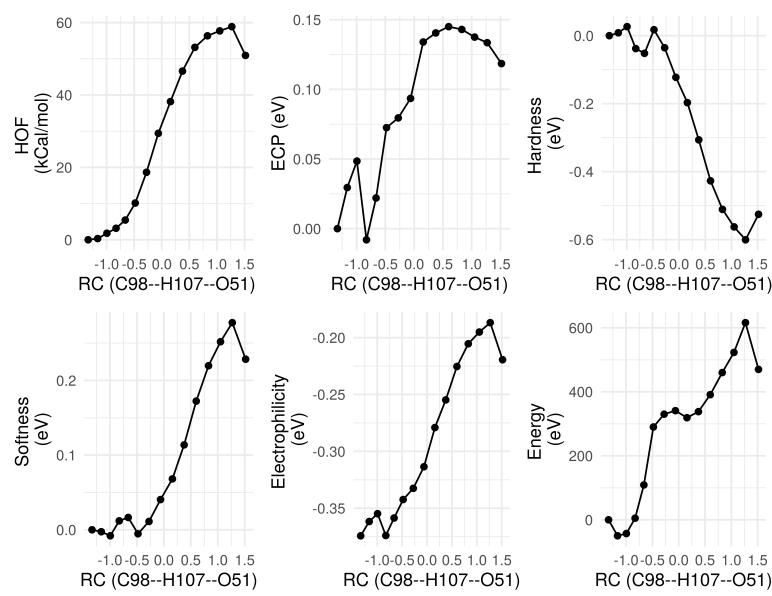


Figure 53: Global descriptors calculated for each step of the reaction trajectory.

In Figure 54 below we can check the energy variation for the same reaction calculated with different methods, published in an article in the Journal of Chemical Information Modeling.

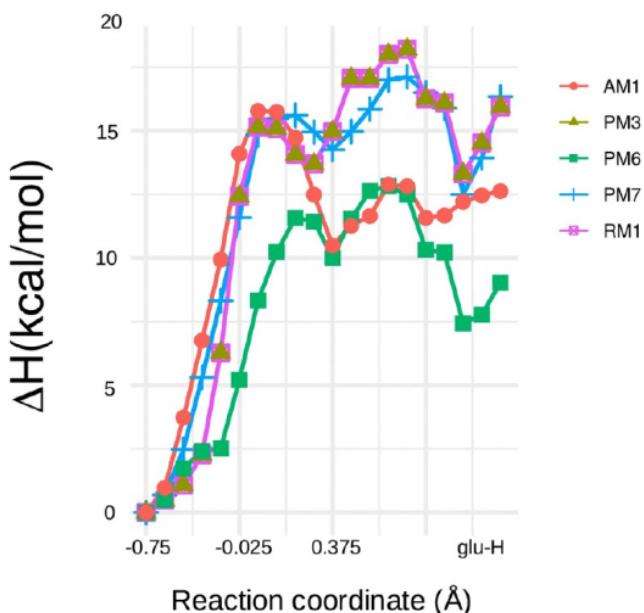


Figure 3. Variation of the enthalpy of formation for the simulated reaction coordinate geometries of the TIM catalysis first elementary step, the GLU165 residue protonation (glu-H) through the H1 extraction from DHAP.

Figure 54: Energy variation calculated for this reaction in a theoretical study published by our group. DOI:10.1021/acs.jcim.9b00860

In the analysis that was published, most of the methods show a transition state close to -0.025 of the reaction coordinate, almost at the point of equal distance from the proton transfer, then there is a trough, denoting an intermediate state and then another increase of energy to another valley in which the product must settle. As seen in the image above, PM6 is the most discordant method in terms of heat of formation.

In Figure 55, we show the descriptors for the oxygen atom of the GLU residue. In the script generated by PRIMoRDiA we analyzed 9 properties per atom of the reaction coordinate. For oxygen, it is interesting to highlight some, such as the decrease in electron density, reaching a minimum in the transition state and then returning to the value in the products. The partial charge also increases, indicating that there is a transfer of electrons from oxygen to carbon, and that the surrounding system then "returns" charge to oxygen, but in the end it still has less, since now the carboxylic group is protonated and therefore neutral. As for the local hardness, this atom presents different behaviors for each type of calculation method, to interpret what each one means it is important to understand each model. Briefly, the LCP-based hardness has more correction with the global hardness descriptor, indicating that it had a propensity to interact by hard-hard process and then this behavior decreased as load was

transferred.

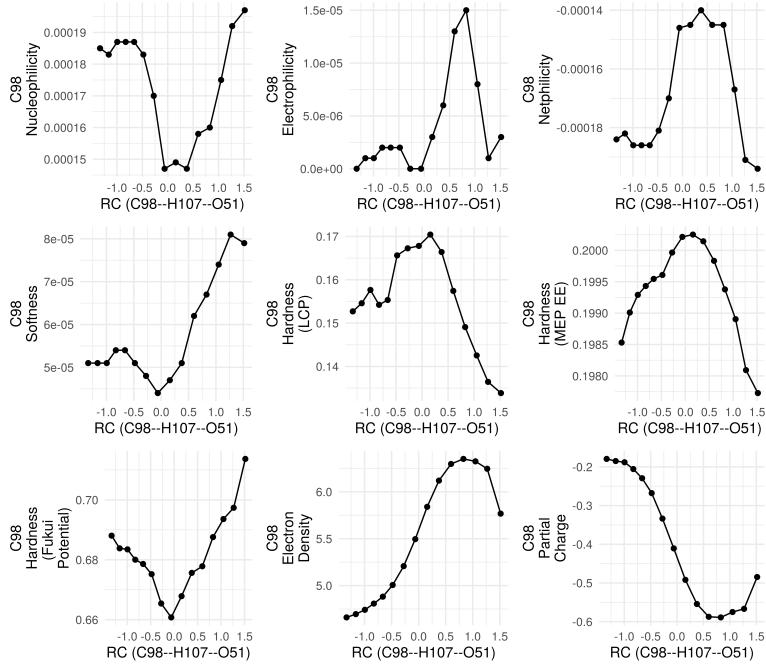


Figure 55: Local descriptors calculated for the O2E atom of the carboxylic group of the catalytic Glutamate for the reaction trajectory.

The descriptors for C1 carbon are in Figure 56. There's a lot of interesting stuff to analyze in this case. First, as for two hardness methods, the maximum value matches the transition state and closely resembles the reaction energy profile. Since the dominant interaction for proton transfer is hard-hard, carbon assumes the maximum value for this descriptor, in two methods mepEE and LCP. The hardness method based on the Fukui potential gives the force exerted on the nuclei with respect to the variation in the number of electrons, which has a more different interpretation from the general one about common hardness.

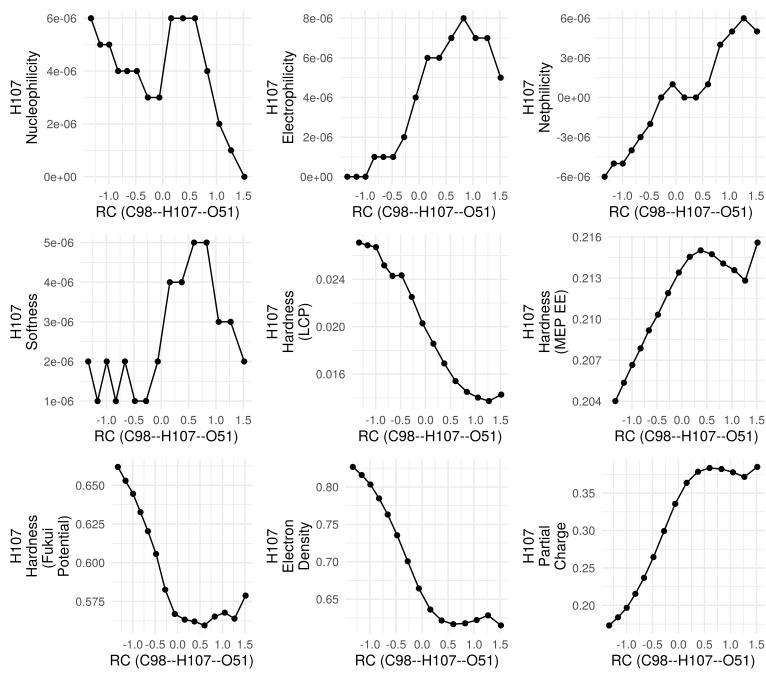


Figure 56: Local descriptors calculated for the substrate C1 atom for the reaction trajectory.

The descriptors for the transferred atom, the hydrogen of DHAP (H1), are shown in the figure below. Descriptors based on molecular orbitals, such as electrophilicity, nucleophilicity and netphilicity showed variation but with very low values. Now it's interesting how the electron density is decreasing, becoming more and more a proton, a positively charged hydrogen. The local hardness with the greatest connection with electrostatic interactions (MEP EE) was the one that presented the greatest coherence with the process, with an increase in electrostatic interactions close to the transition state, decreasing and reaching the maximum in the products, where the partial charge reaches the highest value, and this atom tends to be transferred by the same type of interaction.

As in Tutorial 3, we will generate descriptor images for some points of the trajectory. PRiMoRDiA writes a Pymol script for each file to automate the visualization of descriptors. Open Pymol and run the file prefixed with "sys0" and suffixed with ".pym". As shown in ?? and in the command box below.

```
@sys0_pymols.pym
```

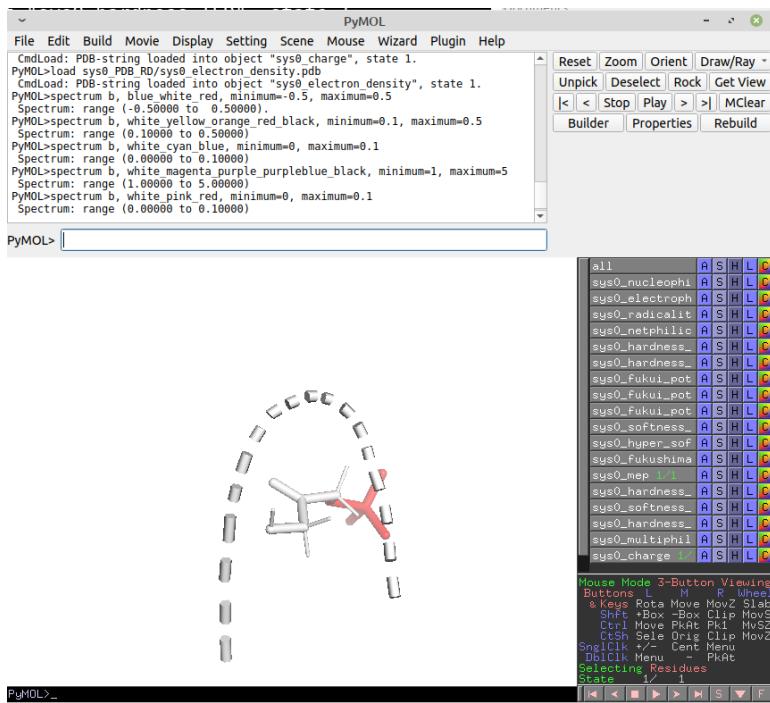


Figure 57: Pymol window with loaded descriptor results and the descriptor object

The next figures were the result of the rendering process in pymol for the local hardness descriptor with the second method available in PRIMoRDiA. The object with the entire protein was colored with the carbons in green and all other non-cartoon representations were hidden so as not to disturb the visualization of the object with the descriptor. For the chosen palette, the command used was

```
spectrum b, white_yellow_orange_red_black, minimum=0.1,maximum=0.2
```

These values can be adjusted slightly to improve the highlighting of atoms with the highest hardness value. Then, with the command "ray" followed by "png" the images were saved. Figure 58 shows the local hardness for the first coordinate step, in Figure 59 for the seventh structure, which corresponds to the transition state, and in Figure 60 for the step corresponding to the products.

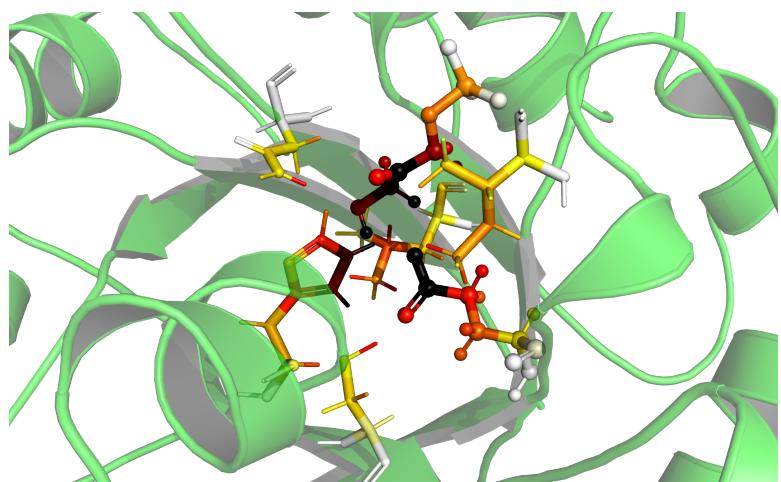


Figure 58: Local hardness for the quantum-mechanically treated enzyme atoms for the structure corresponds to the first step of the reaction trajectory.

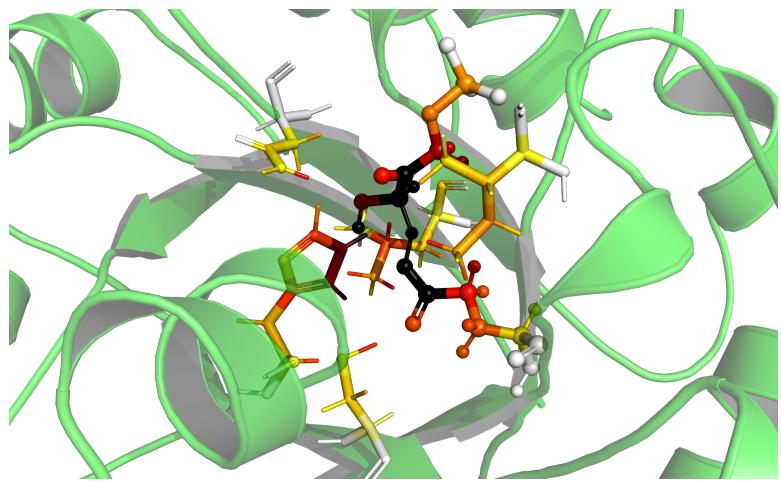


Figure 59: Local hardness for the quantum-mechanically treated enzyme atoms for the structure corresponds to the transition state of the reaction trajectory.

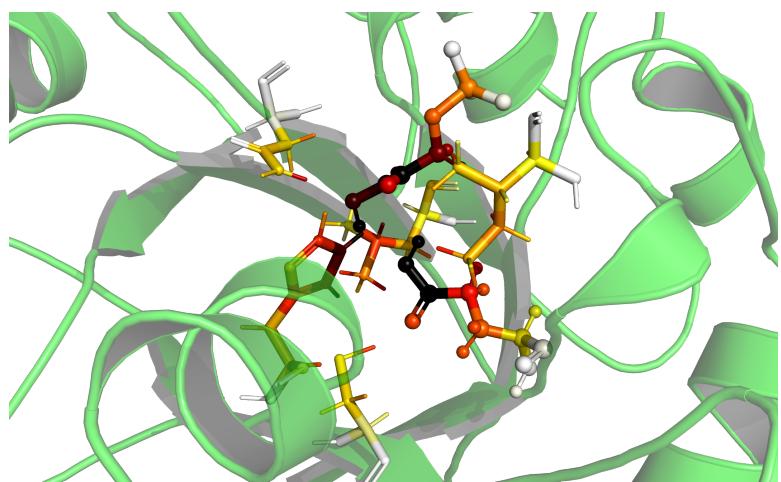


Figure 60: Local hardness for the quantum-mechanically treated enzyme atoms for the structure corresponds to the transition state of the reaction trajectory.

PRIMoRDiA also writes special R scripts for the analysis of descriptors for specific residues, indicated in the input, producing bar graphs with the mean and standard deviation, as in the figure below for the electrophilicity descriptor in Figure 61, nucleophilicity in Figure 62.

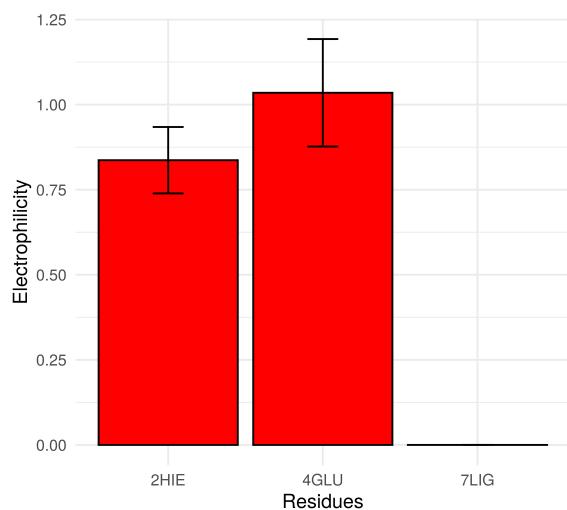


Figure 61: Bar graph for the electrophilicity in the indicated residues during the trajectory.

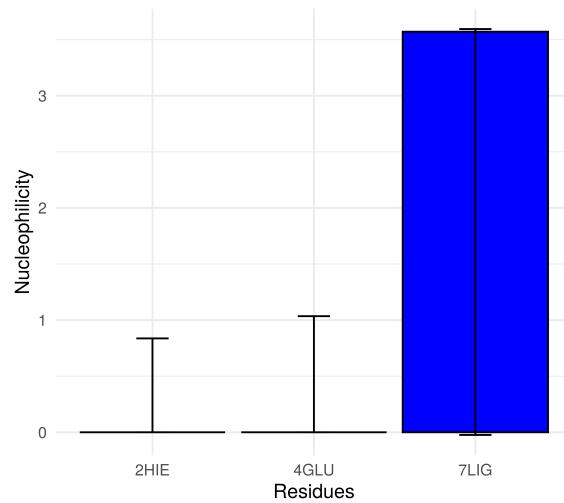


Figure 62: .

Also, the variation of these values during the trajectory is followed with the calculation of moving averages, as in Figure 63 it is shown for electron density in each residue indicated in the input, for local hardness in Figure 64 and local softness Figure 65.

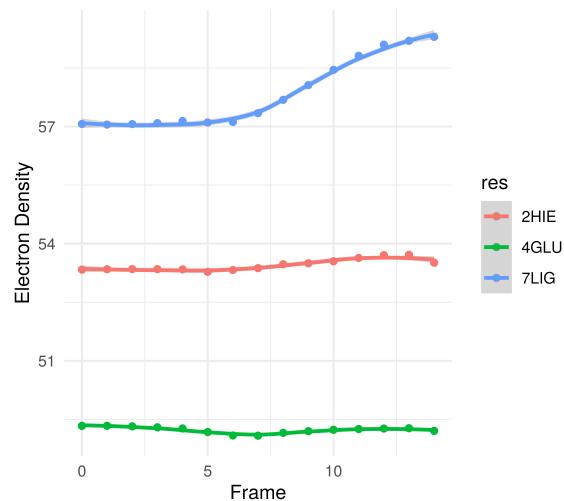


Figure 63: Moving averages for the electronic density for the residues indicated in the input for trajectory analysis.

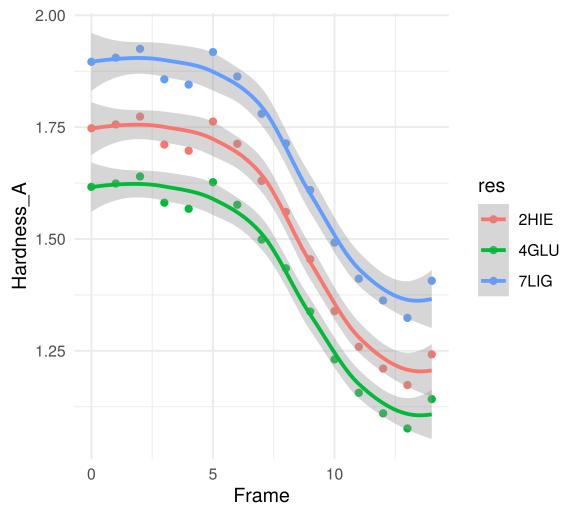


Figure 64: Moving averages for the local hardness for the residues indicated in the input for trajectory analysis.

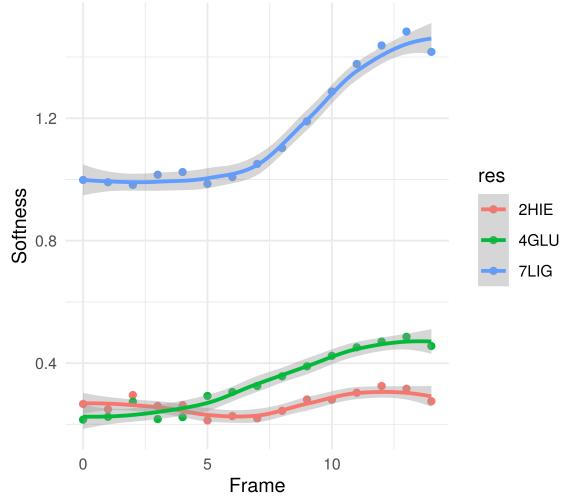


Figure 65: Moving averages for the local softness for the residuals indicated in the input for trajectory analysis.

We can see how the electron density and local softness in LIG, the substrate, increases. In this transition structure formed in the product of our simulation, the substrate forms a double bond when the proton is donated, and as the descriptors indicate carbon and molecules received charge donation. The increase in softness may mean that now, with the pair formed, this molecule has a greater propensity to form covalent bonds through overlapping molecular orbitals and also to transfer charge.

There are several analyzes that we can do based on reactivity descriptors, which in turn are also numerous, for each atom involved and for each residue. PRIMORDiA provides an automatic way to

arrange all these results as ready for analysis as possible.

6 Tutorial Extra: Google Colabs

PRIMoRDiA is software written to compile and run simply and efficiently in Linux environments. However, the user will not always have access to a local machine with a Unix-based operating system, and often for educational and software testing purposes, the user is unwilling to switch operating systems. In fact, PRIMoRDiA is not intended to be cross-platform software in essence, but with the cloud computing capabilities that are available these days, it is possible that the program can be run from any device with internet access.

In this tutorial we will explore the use of the Google Colabs platform. All you need to do is have a gmail account, and features like 12GB of ram and 100GB of storage will be available for free. In addition to the platform issue, the lack of local resources can also be resolved through this approach.

Let's exemplify the installation and execution of the first tutorial presented in this document. The first step is with your Google account to enter the colabs platform: <https://colab.research.google.com>. There are basic tutorials there teaching you how to use the platform for simple tasks and data analysis in Python. In fact, Google colabs provides you with a Python environment, with the same logic as a Jupyter notebook, which are ways to run interactive Python scripts in the Browser.

In our main repository directory, there is a file named "Tutorial_PRIMoRDiA_Colab.ipynb". This file can be opened in Google Colab where you can perform all the steps of this tutorial. However, it is also possible to create your own file and copy the commands as shown in the next figures.

In Figure 66 we see the commands needed to install two libraries needed to compile PRIMoRDiA, it's actually the same compilation process performed on a local Linux machine. The big difference is that in this environment the main language is Python3, and commands that will need to run in "bash" (the interface language between the user and Unix-based systems), need to start with an exclamation point "!" or percentage "%". Basically, for executing binaries, the former is used and for commands necessary for file browsing, the latter is used.

▼ Installation of third party libraries

```
[ ] !apt install g++-8  
!apt install libeigen3-dev
```

Figure 66: Installing external libraries in the Google Colab environment.

Therefore, in a code cell of your file opened in Colab, type the same commands shown in Figure 66, and execute it by clicking on the button commonly known as "play". After this execution, let's clone the PRIMoRDiA repository, navigate inside the folder, configure the compilation and run it. This walkthrough is shown in Figure 67.

Clone the Repo and Install

```
[ ] !git clone https://github.com/igorChem/PRIMoRDiA1.0v  
%cd /content/PRIMoRDiA1.0v  
!cmake .  
!make
```

Testing the compiled executable without arguments:

```
▶ ./PRIMoRDiA_1.25v
```

Figure 67: Configuring and compiling the executable.

Remembering that in Colab, the main directory is /content. This data stays in the cloud, taking up space on your Google Drive. To access the data in the folder, just access the menu on the left of the screen, where you can also select the files and do the download.

PRIMoRDiA's execution logic follows the same as shown in the other tutorials. To run them don't forget to first decompress the directory using the command "!tar -xf Tutorials_Files.tar.bz". After that it is possible to enter the directory where the data for the execution of the first tutorial are, as shown in Figure 68.

▼ Tutorial 1

Go to the folder where are the Tutorial Files

```
[ ] %cd /content/PRIMoRDia1.0v/Tutorials_Files/Tutorial_1  
[ ] !/content/PRIMoRDia1.0v/PRIMoRDia_1.25v -input -op 1 -p mopac -grid 40
```

Let us see the produced input:

```
[ ] %cat < primordia.input
```

Let us use the edited input provided along the tutorial files

```
[ ] %cat < primordia_edit.input  
[ ] !/content/PRIMoRDia1.0v/PRIMoRDia_1.25v -f primordia_edit.input
```

List the results of the calculation

```
▶ %ls
```

Figure 68: Commands used in the execution of the first tutorial within the Google Colabs platform.

After executing the commands, it may take a few minutes, the results will appear in the same folder. To continue doing the analysis part of tutorial 1, as already described in this document, you need to download the files to use the graphical tools on your local machine.

In conclusion, there is no big difference in the execution of the tutorials when performed on the Google Colabs platform, and it can be a very helpful tool to learn how to use the software and for classes or academic courses, or even for those with very limited computing resources.