NCI MILANO manual

v 1.0

GABRIELE SALEH
Dept. of Chemistry, University of Milan, Italy
Center for Material Crystallography, Aarhus, Denmark
gabriele.saleh@umimi.it

LEONARDO LO PRESTI
CNR-ISTM, c/o Dept. of Chemistry, University of Milan, Italy
Center for Material Crystallography, Aarhus, Denmark

CARLO GATTI
CNR-ISTM, c/o Dept. of Chemistry, University of Milan, Italy
Center for Material Crystallography, Aarhus, Denmark

INDEX

Sommario

1. HOW TO RUN

To run the code, the name of the input file (the template to build such file is given along with the source code) has to be written in the command line after the name of the executable:

```
./NCImilano.exe  name-of-input-file
```

If the calculation is time-consuming, you can send it in background by appending "&" to the command line.

2. GENERALITIES ABOUT THE CODE

NCImilano takes wavefunctions and/or electron densities (either experimentally or theoretically derived) and calculates several quantities related to non-covalent interactions(NCI) description. *NCImilano* is composed of three different sections:

(1) calculation of properties from wavefunction
(2) conversion of TOPOND grid files to cube format
(3) calculation of quantities related to electron density-based non-covalent interactions description (RDG, $\rho*\text{sign}(\lambda_2)$ and approximate Energy Densities calculated using Abramov's functional, hereinafter collectively called "Abramov's Energies" [1])

PROPERTIES FROM WAVEFUNCTION section. This part of the program can be used to calculate grid files (cube format) of Electron Density, its derivatives and Energy Density starting from a wavefunction file (".wfn") produced by GAUSSIAN [2]. The only input required is the wavefunction file, whereas the quantities which can be calculated are: electron density, its derivatives and energy densities (see appendix A for a more detailed description of the latter).

TOPOND TO CUBE CONVERSION section. This section must be used when dealing with grid files obtained from periodical calculation performed by CRYSTAL 06/09 [3]. In order to produce 3D grid files of electron density and its derivatives, the TOPOND08 [4] code has to be run* before *NCImilano*. The purpose of this section is to convert grid files from the output format of TOPOND to the cube format (in order to be read from the third part of the program, see *infra*). The user can also choose which atoms/molecules will be written in the cube file (that is which atoms/molecules will be visualized together with isosurfaces)

RDG, $\rho*\text{sign}(\lambda_2)$ AND ABRAMOV'S ENERGY section. This section reads grid files (cube format) of electron density and its gradient and Laplacian in order to calculate Reduced Density Gradient (RDG), $\rho*\text{sign}(\lambda_2)$ (*i.e.* the electron density multiplied by the sign second greatest eigenvalue of the electron density Hessian matrix) and Abramov's Energies [1] (see appendix A). The user can decide to apply spatial as well as electron density cutoff in the process of calculating these quantities. This section can also produce files which can be used to build plot of RDG versus $\rho*\text{sign}(\lambda_2)$ and RDG versus ρ.

Details about these quantities and how they can be used to investigate non-covalent interactions are given in Appendix B.

3. THE INPUT FILE

We provide an input file template listing all accepted commands. Please do not modify the command keywords and do not change the order they appear in the template. No further commands besides the

* CRYSTAL can only calculate grid files of electron density (not its derivatives) and electrostatic potential

ones already present in the input template are needed, the only exception being the SYMM lines in the TOPOND TO CUBE CONVERSION section (see *infra*). Comment lines are obviously allowed[†].

The input file is composed by two different type of command:

- lines containing "**:**" symbol: in these lines, the user has to insert characters (file names) or number (cutoff, etc.; important: *all the input values are considered as given in atomic unit*) after the ":" symbol
- lines containing "**=**" symbol. This lines are characterized by various keywords written after the symbol "=". The user has to choose which ones he/she wants to use by putting "*" symbol right before the corresponding keyword

The first two lines of the input file do not belong to any specific program session. In the first one:

OUTPUT_FILE_NAME:

Append the name of the output file [character; here, as in the rest of the input file, the maximum allowed length of keyword+"file_name" is 45 characters]. Such file is used by the program to write what has been done (read and written files and options chosen by the user) and possible problems during execution. Moreover, when RDG is calculated, the program prints also a table with the values of the calculated functions (electron density, ρ*sign(λ_2) and Abramov's Energies, see *infra*) contained in the regions of space having RDG lower than a certain value, *i.e.* the integral of such quantities over the space contained into isosurfaces.
Empty line in the place where the output file name is supposed to be written will result in error and the program will stop.

The second line:

SECTIONS= (*)WFN (*)TOPOND (*)RDG&other_prop

is used to choose which sections will be considered (more than one option is allowed).

3.1 PROPETRIES FROM WAVEFUNCTION SECTION

In this section the only file needed as input is the wavefunction file calculated by GAUSSIAN (".wfn"). The properties which can be calculated are:

- Electron density, ρ(*r*)
- Electron density gradient, $\nabla\rho(r)$
- Electron density laplacian, $\nabla^2\rho(r)$
- Kinetic Energy Density , G(*r*) [see appendix A]
- Potential Energy Density, V(*r*) [see appendix A]
- Total Energy Density, H(*r*) [see appendix A]

When a property is chosen, an output cube file of such property is printed.

[†] no special symbols are specifically required in order to identify a comment line. In case a comment line contains words identical to the ones in the template input file, the use of non-alphanumeric symbol in front of the comment line will avoid any problem.

Available commands:

INPUT:
Name of the wavefunction file, including the extension [character]

OUTPUT:
Name of the output cube file(s). This *file_name* will be suffixed to identify the specific property contained in the cube file. In particular

- $\rho(r)$ cube file: *file_name*_rho.cube
- $\nabla\rho(r)$ cube file: *file_name_gradrho.cube*
- $\nabla^2\rho(r)$ cube file: *file_name_laplrho.cube*
- G(r) cube file: *file_name*_G.cube
- V(r) cube file: *file_name*_V.cube
- H(r) cube file *file_name*_H.cube

CENTER:
X,Y,Z coordinate of the center of the grid file [3 floating point entries]

N_POINTS_X:
Number of points in the X direction [integer]

STEPSIZE_X:
Distance between the points along X direction [floating point]

N_POINTS_Y, STEPSIZE_Y, N_POINTS_Z, STEPSIZE_Z
The same as before for Y and Z direction

The length of the grid along one direction is given by (N_POINTS-1)*STEPSIZE. In the grid, the orientation of the axis is the same as in the "wfn" file, that is the "Standard Orientation" in the GAUSSIAN calculation.

3.2 TOPOND TO CUBE CONVERSION SECTION

The purpose of this section is to convert TOPOND grid files into cube format (in order to be read in the third section).
Since TOPOND does not write any atoms into grid file, also the output of the CRYSTAL calculation is required as input, in order to read the atoms coordinates of the asymmetric unit. The user can choose which and how many asymmetric units will be written in the output cube file (that is which atoms/molecules will be visualized along with isosurfaces).
Up to three grid files can be converted at the same time, but the atoms written into the cube file will be the same in each of them.

Available commands:

N_OF_GRIDS:
number of grid files to be converted from TOPOND to cube format. The maximum number of grid files which can be converted in one run is 3. [integer]

The next two lines are repeated 3 times into the template since 3 is the maximum allowed number of grid files. If N GRIDS is lower than 3, the latter lines are ignored
GRID_FILE_NAME:
name of the TOPOND grid file to be converted into cube format [character]

CUBE_NAME(OUTPUT):
name of the corresponding output cube file [character]. The suffix ".cube" will be added automatically.

CRYSTAL_OUTPUT_NAME:
The name of the CRYSTAL output (with extension) [character]

N_OF_SYMM_OPERATIONS: Here the user has to insert the number of symmetry operations to be read in the following part, *i.e.* the number of asymmetric units which will be written into cube files (X,Y,Z symmetry operation is already present into the template but is neither assumed nor it needs to be used).

'SYMM entries'
The user has to insert as many SYMM entries ("SYMM" keyword+ the symmetry operation itself) as the number given in «number of symm. op.». If a symmetry operation contains translation, the latter can be written both as fraction or floating point[‡]. Examples of allowed format are: "SYMM y-x,-x,z+1/6","SYMM-X,-Y,0.5+Z","SYMM 1/2-x,Y,Z+0.5". Empty spaces within the symmetry operation should be avoided.

3.3 RDG, ρ*sign(λ_2) AND ABRAMOV'S ENERGY SECTION

In this section the user can calculate Reduced Density Gradient (RDG), ρ*sign(λ_2) and Abramov's energy [3]. The input required are the cube files of electron density and its gradient and Laplacian, which can also be the ones obtained in the two previous sections. Electron density derivatives can also be calculated numerically from electron density. An example of RDG isosurfaces with Abramov's energy mapped on them is reported in Fig.1 (Appendix B).
When RDG, ρ*sign(λ_2) and/or Abramov's energy option is chosen, the corresponding cube file will be written. The user can also choose to print a file containing, for each point of the cube file, the value of RDG, electron density and ρ*sign(λ_2), which can be used to build 2-dimensional Cartesian plots as the ones present in refs. 5 and 6.

Available commands:

INPUT= (*)XD2006 (*)GAUSSIAN_CUBE (*)CUBE
The user has to select the format of input cube files (only one option is allowed). In case grid files are produced by XD2006 [7], also the grid files having ".grd" extension (normally printed by XD along with cubes) are required as input. If the cube files have been calculated with GAUSSIAN, the user has to select «GAUSSIAN_CUBE» option by starring it[§]. For all other cube files, hence also for cube files calculated in the

[‡] in the output file symmetry operations will be reported with only 2 decimal places; however, during the calculation they will be used as double precision floating points
[§] This is because GAUSSIAN writes into the cube files of gradient and laplacian additional information (e.g. the gradient components), therefore the reading format has to be different

two previous sections of this program, the «CUBE» option has to be chosen. All the output grid files are written in cube format; moreover, when «XD2006» option is chosen, additional output grid files in ".grd" format are printed. The latter files can be given as input to XDGRAPH (the graphical routine of XD) to create two and three dimensional plots of RDG, ρ*sign(λ_2) and Abramov's energies[**].

CALCULATING= (*)RDG (*)rhosign (*)Abr (*)plot
By starring the various options, the user decides which quantities will be calculated. «RDG» is the Reduced Density Gradient, «rhosign» is ρ*sign(λ_2) and «Abr» is the Abramov's energies; «plot» is the file containing the information to build Cartesian plot of RDG *VS* ρ*sign(λ_2) or RDG *VS* ρ.

RHO_INPUT_CUBE:
The input cube file of electron density [character].

RHO_INPUT_GRD(XD only):
To be used only when grid files have been produced by XD2006. The user has to insert the name of the ".grd" grid file, including extension [character]

GRAD= (*)none (*)read (*)calc
With this option, the user choose whether he/she wants to read the cube file, calculate it with a numerical method [8] (for each cartesian direction, the derivatives for two most external points cannot be calculated with such method, [††] therefore they will be set to zero) or simply don't use the electron density gradient. In the latter case only ρ*sign(λ_2) can be calculated. If «calc» option is chosen, the calculated grid of electron density gradient will be printed in a file called "gradient.cube".

GRAD_INPUT_CUBE:
Name of the input cube file of electron density gradient (ignored if the option «read» in the former line is not starred)

GRAD_INPUT_RDG(XD only):
Line to insert the name of the input ".grd" grid file of electron density gradient (considered only if «read» is chosen in «GRAD=» option and if the input format chosen is «XD2006»)

LAPL= / LAPL_INPUT_CUBE: / LAPL_INPUT_GRD(XD only):
These lines have the same meaning as three previous ones but in this case they refer to electron density laplacian. The file in which the numerically calculated electron density laplacian is written will be called "laplacian.cube"

RDG_CALCULATION: (*)all_points (*)rho_cutoff (*)rhosign_cutoff
The user chooses here in which points RDG will be calculated (only one option allowed):

- «all_points»: the RDG will be calculated for all the points contained in the input grid file
- «rho_cutoff»: the RDG will be calculated in a certain range of ρ
- «rhosign_cutoff»: the RDG will be calculated in a certain range of ρ*sign(λ_2)

In the points where RDG will not be calculated, the value will be set to 10^{10}: in such a way it will not be considered when isosurfaces are built. Note that the calculation of RDG requires the knowledge of electron density gradient: if such quantity has not been read from grid file nor calculated, RDG calculation will not be carried out. Moreover if the grid points positions in the two input grid files are not identical, the

[**] To do so, the user has to give two dimensional grid files of $\rho(r)$ and/or $\nabla\rho(r)$ and/or $\nabla^2\rho(r)$ as input. Regarding ρ*sign(λ_2), this quantity cannot be calculated for the two most external points of the grid file: therefore 2D grid files of this function cannot be printed, since to calculate λ_2 the value of ED in the two neighboring points in each cartesian direction is needed (in general when input grid files have less than 5 points in one directions, ρ*sign(λ_2), $\nabla\rho(r)$ and$\nabla^2\rho(r)$ cannot be calculated)

[††] The numerical method used here requires the knowledge of the electron density in the two neighboring points

program will stop (it does not make any sense to calculate RDG using electron density and gradient taken from two different points of the space!). In other words, the input cube files have to have the same center, grid step and number of points in each direction.

LOWER_VALUE:
Here the lower value for the cutoff has to be inserted (ignored if «all_points» is starred in the previous command) [floating point]

UPPER_VALUE:
Here the upper value for the cutoff has to be inserted (ignored if «all_points» is starred in the «RDG CALCULATION» command)[floating point]

RDG_OUTPUT:
The name of output grid file of RDG [character]. The suffix ".cube" will be added automatically (when XD2006 files are given as input, an additional RDG grid file with extension ".grd" is printed)

SPACE_CUTOFF= (*)none (*)atomdist (*)external
This option allows the user to apply a spatial cutoff on the calculation of RDG (also in this case, the points which do not fulfill the cutoff conditions will be set to 10^{10}):

- «none» means that no spatial cutoff will be applied
- «atomdist» means that only the points within a certain distance (called "TOLERANCE" in the following) from each atoms will be considered[‡‡].
- «external» means that, for every Cartesian direction, the RDG will be calculated only inside the cartesian bounding box (i.e. the highest and the lowest values of the atomic coordinates in each direction). The user can also add a distance tolerance (called "TOLERANCE" in the following) to increase the dimension of such bounding box.

TOLERANCE:
This is the spatial tolerance to which the previous input line is referred to. If in the latter «none» is starred, this line will be ignored

RHO_SIGN_CALCULATION= (*)all_points (*)RDG_points
With this option it is possible to decide whether $\rho*\text{sign}(\lambda_2)$ will be calculated for all the points of the input grid files («all_points») or only in those points where RDG was calculated («RDG_points»). If the latter option is chosen, the points out from RDG cutoffs defined previously will be set to zero. Moreover, since the derivatives are calculated with a numerical method (the same used to calculate electron density gradient and laplacian), for each Cartesian direction the correct sign of λ_2 cannot be calculated for the two most external points of the grid (as for numerically calculated gradient and laplacian); therefore in such points $\rho*\text{sign}(\lambda_2)$ will be set to zero as well.

RHO_SIGN_OUTPUT:
The name of the output cube file of $\rho*\text{sign}(\lambda_2)$ has to be inserted here. The suffix ".cube" will be added automatically (when XD2006 files are given as input, an additional $\rho*\text{sign}(\lambda_2)$ grid file with extension ".grd" is printed)

ENERGY_CALCULATION= (*)all_points (*)RDG_points

[‡‡] In other words, the space in which RDG will be calculated is built by superimposing spheres centered at each atoms position and having a certain ray defined by the user

The same as for RHO_SIGN_CALCULATION: the user chooses whether Energy Density will be calculated in all points or only in those points where RDG was calculated. Note that the calculation of Abramov's Energies requires the knowledge of electron density gradient and Lapalcian: if such quantities have not been read from grid file nor calculated, Abramov's energy calculation will not be done. As for RDG, if electron density and its derivatives cube files are read, the position of grid points has to be the same in all cubes.

ENERGY_OUTPUT:
The name of the output cube files of energy density (approximated with Abramov's functional) has to be inserted here. Three output cube files will be created (kinetic, potential and total energy density). The name of such files will be (*file_name* is the character string inserted here):

- Kinetic energy density: *file_name*_Gabr.cube
- Potential energy density: *file_name*_Vabr.cube
- Total energy density: *file_name*_Habr.cube

As for RDG and $\rho*\text{sign}(\lambda_2)$, if XD2006 input grid files were given as input, additional grid files of Abramov's energy density in ".grd" format will be printed.

2DPLOT= (*)rho (*)rhosign
This option is used to build a output file containing, for each point of the input cube file, the value of RDG and $\rho*\text{sign}(\lambda_2)$ (if «rhosign» is starred) or the value of RDG and ρ (if «rho» is starred). Both options can be chosen at the same time.

PLOT_NAME:
The name of the output plot has to be inserted here. Depending on the quantity calculated, the output file will be called (*plot_name* is the character string inserted here)):

- rho*_plot_name* (file for RDG VS ρ)
- rhosign*_plot_name* (file for RDG VS $\rho*\text{sign}(\lambda_2)$)

APPENDIX A: ENERGY DENSITIES

As it is well known in literature, the kinetic energy densityis not uniquely defined [9]. Here we choose the following positive definite expression (for an orthonormal wavefunction):

$$G(\bar{r}) = \frac{1}{2}\sum_{i=1}^{N} \alpha_i \, |\nabla\varphi_i(\bar{r})|^2$$

φ_i being the i-th Natural Orbitals and α_i its occupation number.

When only electron density is available the corresponding kinetic energy density is estimated using Abramov's functional [1]:

$$G(\bar{r}) = \frac{3}{10}(3\pi^2)^{\frac{2}{3}}\rho(\bar{r})^{5/3} + \frac{(\nabla\rho(\bar{r}))^2}{72\rho(\bar{r})} + \frac{1}{6}\nabla^2\rho(\bar{r})$$

Potential energy density is calculated exploiting the local form of the virial theorem, that is (in atomic unit):

$V(\bar{r})$= ¼ $\nabla^2\rho(\bar{r})$ - 2G($\bar{r}$)

Where *V* is the potential energy density. Such expression is used regardless of the way in which kinetic energy has been calculated.

The total Energy density is simply the sum of potential and the above defined kinetic energy contributions:

H($\bar{r}$)= G($\bar{r}$) + *V*($\bar{r}$)

APPENDIX B: RDG-BASED NON-COVALENT INTERACTIONS DESCRIPTION

In 2010, Johnson *et al*. proposed a Non-Covalent Interaction (NCI) descriptor [5], entirely based on ED and its derivatives. The two quantities exploited are:

- The Reduced Density Gradient $s(\bar{r}) = \frac{|\nabla\rho(\bar{r})|}{2(3\pi^2)^{1/3}\rho(\bar{r})^{4/3}}$
- $\rho(\bar{r})sign(\lambda_2)$, where λ_2 is the second greatest eigenvalue of electron density Hessian matrix

In particular, the RDG isosurfaces are used to detect the presence of NCI, while ρ*sign(λ_2) is mapped onto isosurfaces to investigate both the nature and the strength of NCI. For details about how these quantities are used for the investigation of non-covalent interactions the reader is referred to refs. 5 and 6.

Although initially applied only to electron density distribution obtained either from theoretical calculation or Independent Atom Model (IAM), the reliability of this descriptor when applied to experimentally derived electron densities (single crystal low temperature X-ray diffraction) has been demonstrated through a comparison with the NCI picture obtained from fully periodical calculation [6]. Regarding the application of RDG-based NCI descriptor to IAM, a detailed discussion about the differences and similarities between the NCI picture obtained from experimentally derived and IAM electron density distribution is reported in ref. 10.

IAM-ED distribution grid files can be obtained either with XD2006 using the relative keywords in the XDPROP section, or with CRYSTAL. Regarding the latter, the keyword PATO can be added in the process of producing files for TOPOND: in this way, the grid files produced from TOPOND will correspond to IAM density. In both cases, the resulting grid files can be given as input to NCImilano to calculate RDG, $\rho*sign(\lambda_2)$ and Abramov's energies from IAM ED distribution.

Quite recently, our group proposed a new approach in which the energy density is mapped onto isosurfaces (instead of $\rho*sign(\lambda_2)$) [11]. One of the main advantages in using the energy density is that it is a physically rooted quantity, as the integration of H(r) over the whole space gives the electronic energy of the system. When mapped onto RDG isosurfaces, the energy density is able to rank the strength of the various interactions; moreover, the possibility of partitioning the total energy density into potential and kinetic contributions allows one to gain more insight into the different kinds of NCI and to distinguish better their specific nature. In addition, using such quantity, the arbitrary distinction between attractive and repulsive interactions, which is present in the original formulation of this NCI descriptor, is avoided.

An example of energy density mapped onto RDG isosurfaces is reported in Fig. 1.

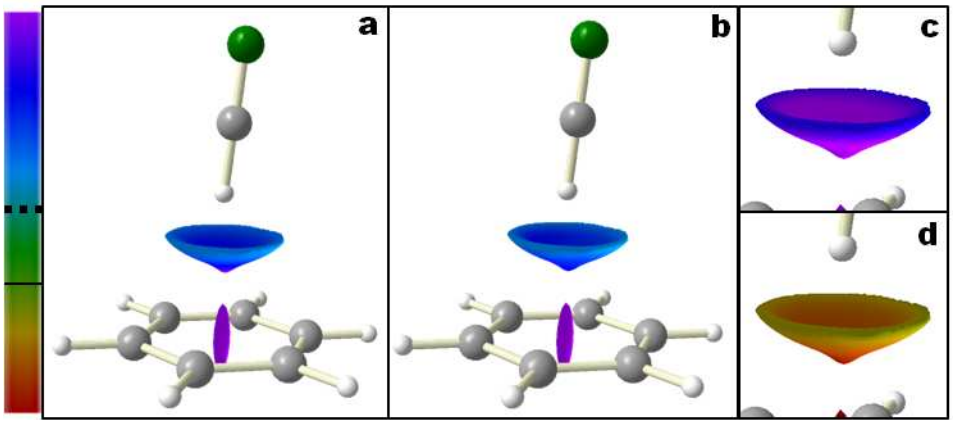

Figure 1. 0.5 RDG isosurfaces for a dimer of benzene and HCN. The function mapped onto isosurface is: **a)** approximate total energy density obtained using Abramov's functional **b)** exact total energy density from wavefunction **c)** kinetic energy density contribution to b) (enlargement of isosurface corresponding to C-H···π) **d)** potential energy density contribution to b) (enlargement of isosurface corresponding to C-H···π). The colour scale is reported on the left. For a) and b), the extreme values are -2.9 (red) and +5.6 (violet) kJ/mol (solid line indicates zero value); for c) and d), the extreme values are -13.4 (red) and +13.4 (violet) kJ/mol (dotted line indicates zero value)

REFERENCES

[1] Y.A. Abramov., *Acta Cryst.***1997**, A53, 264-272
[2] Gaussian 09, Revision **A.1**, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, Gaussian, Inc., Wallingford CT, **2009**
[3] R. Dovesi, V. R. Saunders, C. Roetti, R. Orlando, C. M. Zicovich-Wilson, F. Pascale, B. Civalleri, K. Doll, N. M. Harrison, I. J. Bush, P. D'Arco, M. Llunell, *CRYSTAL06 User's Manual*, University of Torino, Torino, Italy, **2006**
[4] C. Gatti, TOPOND98: *an electron density topological program for system periodic in N (N=0-3)dimensions, User's Manual*, CNR-ISTM, Milano, Italy, **1999;** C. Gatti, V. R. Saunders, C. Roetti, *J. Chem. Phys.* **1994**, *101*, 10686–10696
[5] E. R. Johnson, S. Keinan, P. Mori-Sanchez, J. Contreras-Garcia, A. J.Cohen, W. Yang, *J. Am. Chem. Soc*. **2010**, 132, 6498 – 6506
[6] G. Saleh, C. Gatti, L. Lo Presti, J. Contreras-Garcia, *Chem. Eur. J.* **2012**, 18, 15523-15536
[7] A. Volkov, P. Macchi, L. J. Farrugia, C. Gatti, P. Mallinson, T. Richter, T. Koritsanszky, *XD2006 - A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors*, **2006**. See also http://xd.chem.buffalo.edu/
[8] A. Volkov, Y. Abramov, P. Coppens, C. Gatti, *Acta Crystallogr. Sect.A* **2000**, 56, 332 – 339
[9] R.F.W. Bader, H.J.T. Preston, *Int. J. Quant. Chem*. **2004**, 3, 327-347
[10]G. Saleh, C. Gatti, L. LoPresti, *Comput. Theor. Chem*. **2012**,998,148–163
[11] See abstract book of Electron Charge Density Meeting 6