Practical computational toolkits for dendrimers and dendrons structure design

Nuno Martinho[1,2,3], Liana C. Silva[1], Helena F. Florindo[1], Steve Brocchini[2], Teresa Barata[2,*] and Mire Zloh[3,*]

[1] Research Institute for Medicines (iMed.ULisboa), Faculty of Pharmacy, Universidade de Lisboa, Av. Professor Gama Pinto, Lisbon 1649-003, Portugal

[2] Department of Pharmaceutics, UCL School of Pharmacy, University College London, 29/39 Brunswick Square, London WC1N 1AX, UK

[3] School of Life and Medical Sciences, University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK

* Authors to whom correspondence should be addressed. e-mail: t.barata@ucl.ac.uk & m.zloh@herts.ac.uk

ABSTRACT

Dendrimers and dendrons offer an excellent platform for developing novel drug delivery systems and medicines. The rational design and further development of these repetitively branched systems are restricted by difficulties in scalable synthesis and structural determination, which can be overcome by judicious use of molecular modelling and molecular simulations. A major difficulty to utilise *in silico* studies to design dendrimers lies in the laborious generation of their structures. Current modelling tools utilise automated assembly of simpler dendrimers or the inefficient manual assembly of monomer precursors to generate more complicated dendrimer structures. Herein we describe two novel graphical user interface (GUI) toolkits written in Python that provide an improved degree of automation for rapid assembly of dendrimers and generation of their 2D and 3D structures. Our first toolkit uses the RDkit library, SMILES nomenclature of monomers and SMARTS reaction nomenclature to generate SMILES and mol files of dendrimers without 3D coordinates. These files are used for simple graphical representations and storing their structures in databases. The second toolkit assembles complex topology dendrimers from monomers to construct 3D dendrimer structures to be used as starting points for simulation using existing and widely available software and force fields. Both tools were validated for ease-of-use to prototype dendrimer structure and the second toolkit was especially relevant for dendrimers of high complexity and size.

INTRODUCTION

Dendrimers and dendrons are a type of hyper-branched macromolecules characterised by a well-defined three-dimensional branching architecture with a high degree of mono-disperse molecular weight characteristics emanating from a multifunctional core [1]. They are often defined by the composition and chemistry of their core, branching (internal monomers) and by their multivalent terminal groups (surface groups). This results in a wide diversity of different topologies described in the literature since different core precursor and branching monomers can be used in their preparation. In particular, the use of different types of monomers can have dramatic changes in the size, volume, shape, flexibility, physicochemical properties and available space (interior density) of dendrimers [2, 3]. These features provide an excellent platform for potential applications of dendrimers for various purposes, for example: enhancing solubility and delivery of drugs [4–8], optimization of therapeutic agents toxicity [9, 10], developing sensors [11], diagnostic agents [4] and transfection reagents [12]. The multivalent surface of dendrimers can also be utilized for passive or active targeting by modification of terminal groups using small molecules or macromolecules to employ relevant molecular recognition mechanisms [12–17].

Dendrimers are prepared following a wide range of synthetic routes. Divergent routes essentially proceed from the core outwards, and convergent routes tend to proceed from branched end groups to the core. Due to the hyperbranched character of dendrimers, the size of the structure, complexity and branching increases with each additional generation. The increase of each generation often results in the doubling the number of end groups with the possibility that intricate branched structures can be formed. Eventually dendrimers reach a generation number where end-group crowding results in restricted chemical accessibility self-interruption of synthesis [18]. The conformation of dendrimers at lower generation numbers can be different from higher number generation dendrimers and their conformation in solution is highly dependent on chemical structure and requires analysis on a case-by-case basis. For example, generations 1 and 3 of equilibrated triazine dendrimers in water assume a globular-like configuration with a dense core and flexible surface, generation 5 becomes less spherical but having still dense core, while generations 7 and 9 are porous and open to the penetration to the solvent [19]. As with most polymers, the behaviour of end groups depends on the nature of the terminal groups, in some cases the end groups may be flexible [19] or folded into the interior of the dendrimer [18].

The size and flexibility of structures of many dendrimers prevent the unequivocal determination of their conformation using common experimental techniques. Currently, there are only 15 reported structures of dendrimers determined by X-ray crystallography in the protein data bank (PDB) [20]. There is a significant progress in analysis of dendrimers using NMR from distinguishing the signals of different generations and signal assignments [21, 22]to understanding the dynamics and mobility of dendrimers [23]. However, the full three dimensional structure determination of dendrimers using NMR is still hindered due to the repetitive nature of monomers in each generation of the dendrimer that leads to ambiguous assignments of nOe signals. The use of biophysical and physicochemical techniques can provide information on complexes that dendrimers form with other molecules of interest [24], however the detailed structural information that can be obtained experimentally remains limited.

Due to the large variation in chemical structures of monomers that can be used to prepare dendrimers, it is increasingly difficult to envision their important properties, such as shape, flexibility and interior flexibility. Molecular modelling and simulations has become a powerful tool to probe and model molecular structural information. Molecular modelling has been increasingly used to rationally design dendrimers for biomedical applications with special focus on studying their structure and dynamic response to environment stimulus (e.g. pH change) as well as interactions of dendrimers with other molecules [23, 25–28]. Modelling studies of dendrimers have been conducted using different force fields developed for proteins and small molecules, including CHARMM[29], AMBER[30], CVFF[31], Dreiding[32], GROMOS[33], COMPAS [34] and OPLS[35]. Although these types of molecular modelling studies provide information that corroborate experimental data, it still remains difficult to generate the three dimensional models of relevant dendrimers *in silico*. This is particularly true for larger generations or for non-regular dendrimers or dendrons with complex structures. Several molecular modelling tools are suited to build regular dendrimers and hyper-branched polymers, namely Starmaker (part of Silico toolkit)[36, 37], Dendrimer Building Toolkit[30], Dendrimer Builder in Materials Studio [38] and HBP builder [39]. However, there are still opportunities to automate process of generation of the structures of more complex dendrimers or dendrons. We have previously developed a method to describe dendrimer structures as a sequence of monomers accompanied by a “connectivity table” [40] for use in generating structure using XPLOR-NIH software [41]. This method still required the manual description of the sequence and the “connectivity table” and thus lacked the automation required for general use. Therefore, there is still a demand for a GUI that allows building frameworks of dendrimers and dendrons with different chemistries and being able to use different force fields in a reliable manner.

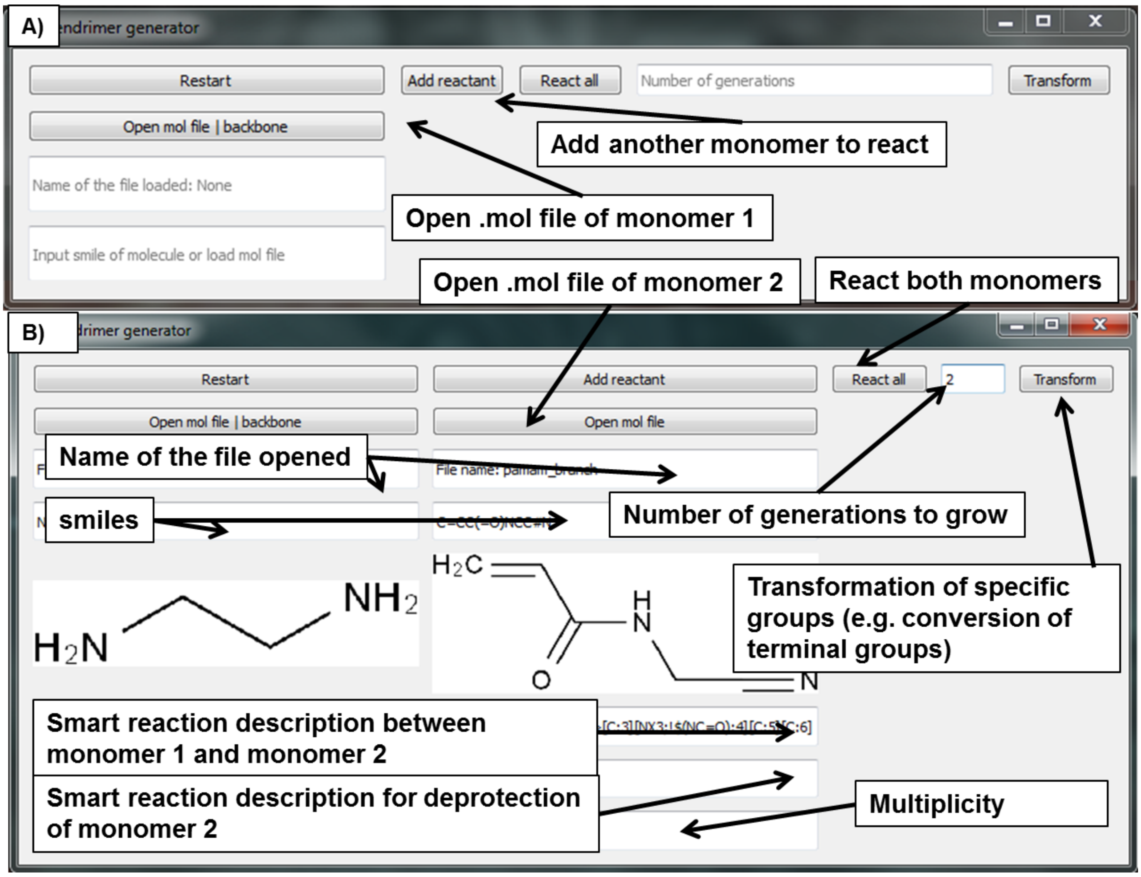
This manuscript concerns the description of an algorithm presented in a GUI to generate the sequence of monomers and connectivity tables to be exported to XPLOR to generate topology files of dendrimers and complete 3D structure to be used in MD simulation packages. Additionally, we also provide an automated method to generate 2D structure and smiles of dendrimers for fast prototyping and database management of these macromolecules. Both tools were shown to provide a faster and easier way to assemble dendrimers in preparation for further computational studies.

RESULTS AND DISCUSSION

Our approach aims to provide a framework of tools to build dendrimers and dendrons in a reliable and consistent way using two approaches according to the level of parameterization and complexity of the dendrimer required. As with other dendrimer building strategies, the principle is to construct the dendrimer framework from the set of monomers needed to increase the dendrimer generation number.

We focused on tools written in Python language that allow the design of different chemistry templates with varying complexity. In cases where parameterization for further MD simulation of dendrimers is not required and only the assembly is important we provide a GUI to generate 2D mol files of dendrimers (**Toolkit 1**). When parameterization is required to further model the dendrimers, we created a GUI to automate our previously reported method [40]. This new GUI generates the input file containing the sequence of monomers and their connectivity to be used in XPLOR to generate the pdb and psf files that can then be submitted to MD simulations (**Toolkit 2**). This tool is not restricted to build large dendrimers but was also focused on being able to build complex dendrimer and dendrons frameworks. For example, a more complex dendrimer may comprise a small number of generations derived from one type of monomer (e.g. flexible, hydrophobic), which is then followed by another type of monomer (e.g. rigid, hydrophilic) with variations of multiplicity and length of linear sequences.

Toolkit 1 - Quick generation of 2D mol files. While generating the 3D structure of dendrimers is a key goal when assembling dendrimers, there are several computational applications that do not require 3D coordinates. Having a quick and reliable way to generate 2D structure of dendrimers *in silico* is useful for various applications including database management for dendrimers, creation of images, rapid prototyping and in cheminformatics, such as QSAR, data mining and machine learning models. It is therefore useful to have an automated tool for building dendrimers without the need for more complex modelling tools and parameterization. Thus, we developed a GUI using PyQt4 (Figure 1) using the RDKit library in Python to have a flexible tool to rapidly generate 2D mol files of dendrimers, which include the smiles, a 2D image and a mol file with no coordinates.


A)
ndrimer generator
Restart
Add reactant
React all
Number of generations
Transform
Open mol file | backbone
Name of the file loaded: None
Input smile of molecule or load mol file
Add another monomer to react
Open .mol file of monomer 1
Open .mol file of monomer 2
React both monomers
B)
drimer generator
Restart
Add reactant
React all
2
Transform
Open mol file | backbone
Open mol file
Name of the file opened
smiles
Number of generations to grow
H_2N
NH_2
H_2C
H
N
O
N
Transformation of specific groups (e.g. conversion of terminal groups)
Smart reaction description between monomer 1 and monomer 2
Smart reaction description for deprotection of monomer 2
Multiplicity

File name: pamam_branch

C=CC(=O)NCC#N

[C:3][NX3:!$(NC=O):4][C:5][C:6]

Figure 1. Graphical user interface for assembling 2D structures of dendrimers showing a way how to assemble PAMAM dendrimers. The application utilizes mol files of the precursor's monomers and information on how these are reacted using SMARTS reaction description. A) Snapshot of the GUI of the first step in building a dendrimer and requires an entry to load the core; B) Snapshot of the GUI after adding a reactant and adding relevant information into entry boxes.

RDKit is an open-source cheminformatics software with a large collection of algorithms to perform *in silico* manipulation of molecules. In particular this library has functions to describe and apply chemical transformations using the SMARTS-based language similar to Daylight's Reaction SMILES [42]. This module is commonly used for virtual chemical library generation to quickly generate big libraries of compounds for virtual high throughput screening [43]. In order to "synthesize" a new compound, the algorithm requires the smiles of both reactants and the reaction between both described in SMARTS language. The functional groups required for the reactions are then mapped onto both precursors and linked accordingly to the description. For example, a generic peptide bond formation can be expressed as an amine ($-CNH_2$) reacting with a carboxyl (-COO) and would be described in SMARTS as [**C**X4][**N**H2].[**C**X3](=0)[**O**X1]>>[**C**X4][**N**X3][**C**X3](=O), where each reactant is separated by a "." and the final product of this reactions is given by the ">>" indicator (in this case -CNCO-). One of the difficulties observed in using this library to assemble dendrimers is that all mapped substructures (all functional groups) would react to give rise to all possible intermediate products which themselves could undergo further reaction. This results in a massive number of products, expensive computational time and it is difficult to retrieve the intended product from the pool of compounds that are generated.

To prevent this issue, we followed the same approach in the synthesis of dendrimers where branch monomers not allowed to react were conjugated with protecting groups (e.g. ester group of a carboxylic) or with groups that would not be mapped by the SMARTS identifier (e.g. a triple bond instead of a double). These protected groups do not undergo reaction to generate side products. Consequently, this allows application of transformation functions available in the RDKit library (.ReactionFromSmarts; .RunReactants; .ReplaceSubstructure) to link all mapped monomers that are allowed to react and obtain the product. This is followed by deprotection of the protecting groups in the newly formed layer, and repetition of the previous reaction until the desired generation is obtained. Since some functional groups can react more than once (e.g. amines), multiplicity has to be set to instruct how many times each group reacts. A specific examples is shown in Figure 2. A standard procedure goes as follows:

1. Click on open file to open a mol file of the monomer of the core. Once the file is loaded the name of the file is displayed on a text box as well as the smiles of the mol file and an image of the 2d structure of the input mol file. Alternatively, SMILES can be input directly in the text box.
2. Click add reagent and open the mol file of the monomer that will be attached to the core structure. The name of the file, the smiles and the d structure of the input mol file will be displayed.
3. Add the SMARTS reaction between the core and the branching monomers in the corresponding text box.
4. Add the SMARTS reaction for the deprotection reaction of the branching monomers.
5. Add the multiplicity of the reaction and the number of generations intended.
6. Press React all to create the smiles, 2D image of the dendrimer as well as a mol file without 3D coordinates.

Alternatively, a mol file of the dendrimer can be loaded to perform single modification, such as change of the terminal group chemistry by pressing the transform button and executing the SMARTS reaction for this modification. A key limitation of this tool is to achieve the balance between the use of protecting groups and types of reaction between the core and the branching monomers. A way to overcome these problems is by creating different layers of the dendrimer one by one and using it as the backbone input each time. Furthermore, the generation of more complex dendrimers and dendrons can be challenging, since the mol file generated does not have 3D coordinates, random generation of coordinates of higher generations dendrimers leads to entangled branches or monomers and thus not be suited for 3D visualization. Nevertheless, this tool proved to be highly flexible and allowed the rapid prototyping of differently types and families of dendrimers (Figure 3).

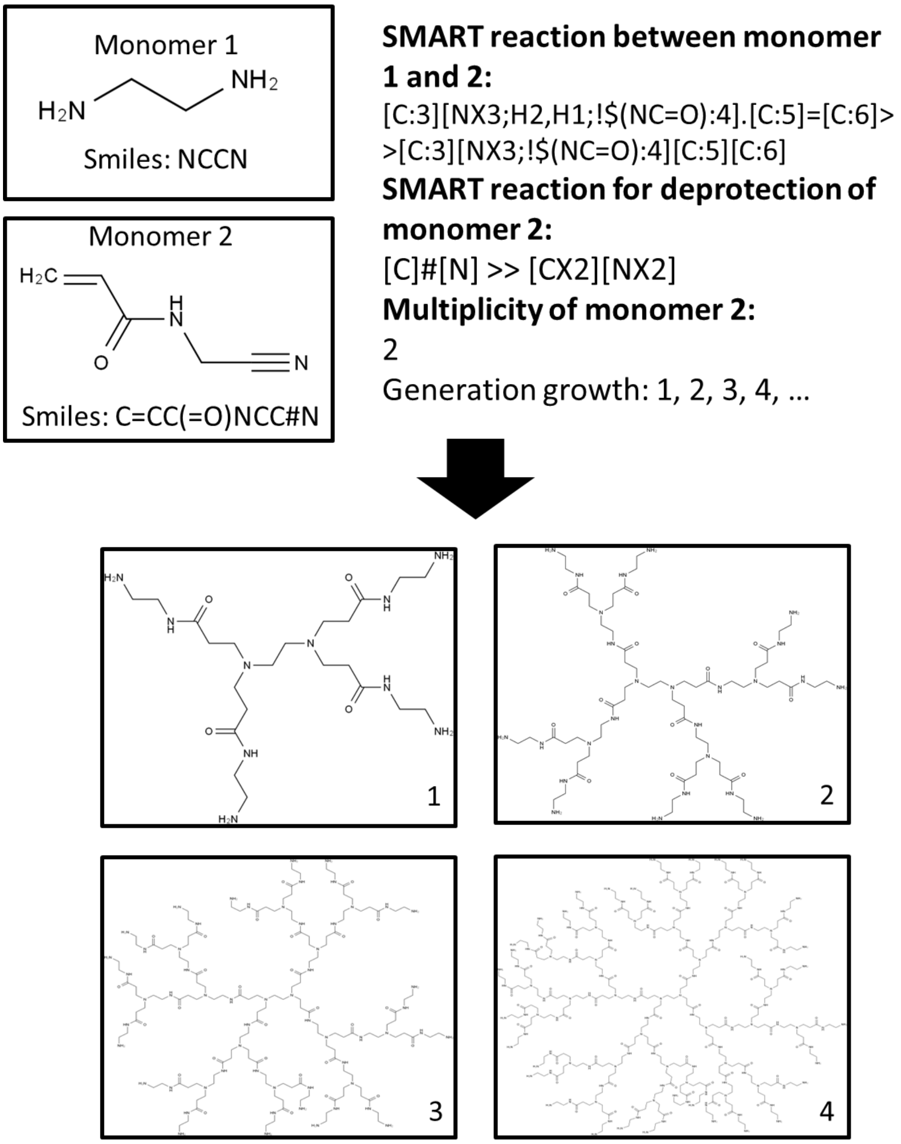

Monomer 1
H2N
NH2
Smiles: NCCN
Monomer 2
H2C
H
N
O
N
Smiles: C=CC(=O)NCC#N
SMART reaction between monomer 1 and 2:
[C:3][NX3;H2,H1;!$(NC=O):4].[C:5]=[C:6]>>[C:3][NX3;!$(NC=O):4][C:5][C:6]
SMART reaction for deprotection of monomer 2:
[C]#[N] >> [CX2][NX2]
Multiplicity of monomer 2:
2
Generation growth: 1, 2, 3, 4, …
1
2
3
4

Figure 2. Example of 2D structure of a dendrimer generated using the RDKit library. Smiles, SMARTS of the reactions, multiplicity and generation desired were used as the input to generate different generations of a dendrimer. The application generated the smiles, a mol file without coordinates and an image.

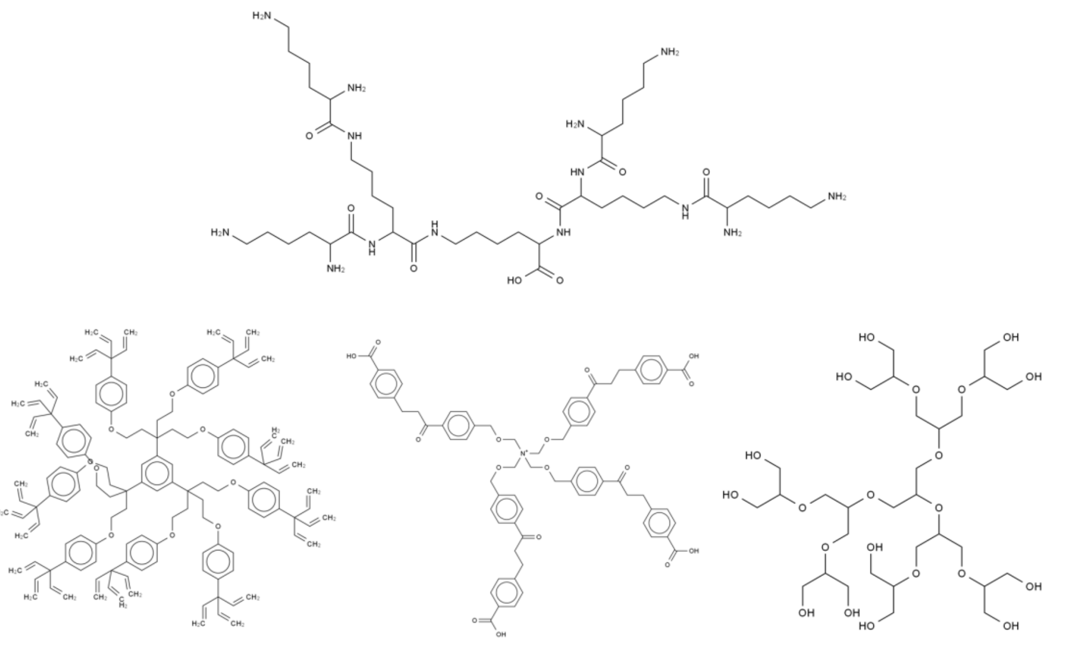
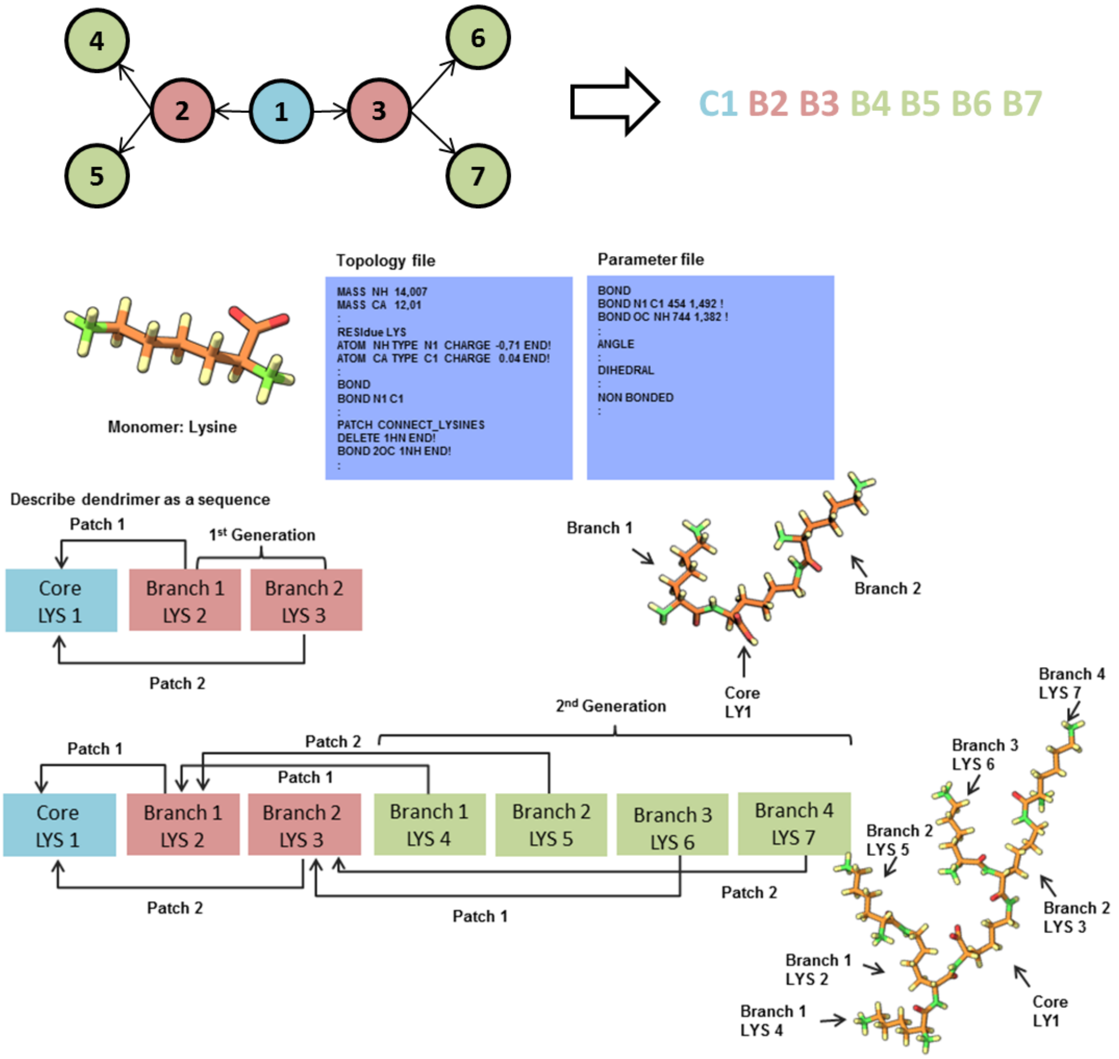


Figure 3. Examples of different types of dendrimers generated as 2D mol files using the RDKit module.

Toolkit 2 - Generation of 3D structures for MD simulations.

We previously reported a method where dendrimers were described as a linear sequence of monomers and were then assembled through the description of a connectivity table (similar to the creation of a disulfide bond in linear peptides, Figures 4 and 5) [40]. As shown in Figure 4, a generation 1 poly-lysine (PLL) dendrimer can be easily described by the core monomer and the two generations monomers translating into a sequence of “core; generation 0: monomer 1 and 2; generation 1: monomer 1,2,3 and 4” for a total of 7 monomers, lysine amino acids. However, contrary to a linear sequence, in order for XPLOR to understand how to assemble the dendrimer sequence, a “connectivity table” has to be provided. In this case of the PLL dendrimers, the monomer 1 (the core) was attached to the monomer 2 (branch 1) through the patch reference 1 (peptide bond through the amine group of the alpha carbon) and the monomer 3 (branch 2) through the patch reference 2 (peptide bond through the amine from the side chain). Then each of the branch monomers was further connected two times with another lysine monomer given the same patch reference. This can be repeated until the intended generation is obtained. Although this can be feasible to describe manually for small generations it becomes increasingly more difficult and prone to errors for higher generations. Consequently, in order to automate this process, we developed a python GUI application using PyQt4 library (Figure 5) that generates a text file containing the sequence and connectivity table to be used in XPLOR.

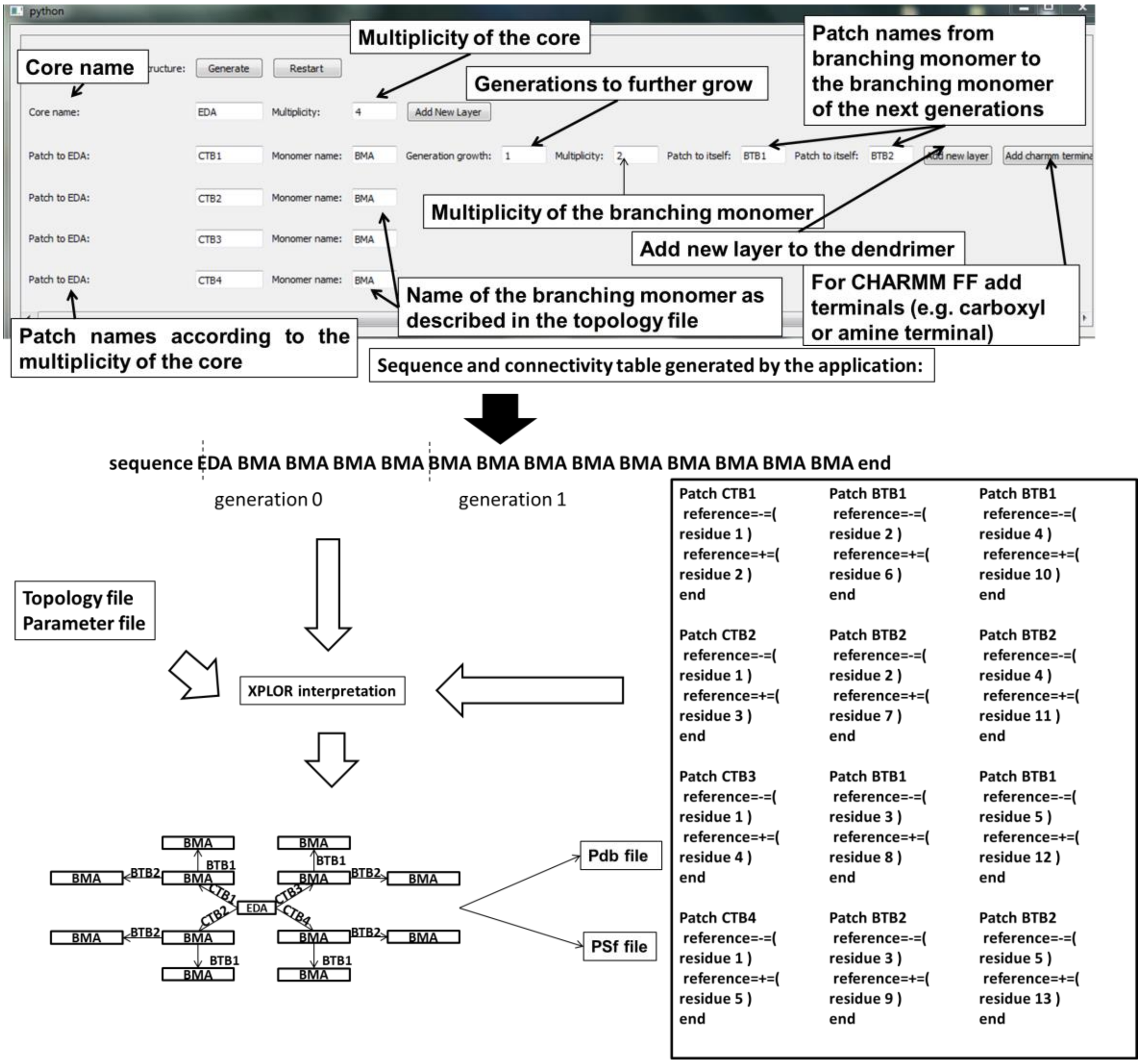


```
MASS  NH  14,007
MASS  CA   12,01
:
RESIdue LYS
ATOM   NH TYPE  N1  CHARGE  -0,71 END!
ATOM   CA TYPE  C1  CHARGE   0.04 END!
:
BOND
BOND N1 C1
:
PATCH  CONNECT_LYSINES
DELETE 1HN END!
BOND 2OC 1NH END!
:
```

```
BOND
BOND N1 C1 454 1,492 !
BOND OC NH 744 1,382 !
:
ANGLE
:
DIHEDRAL
:
NON BONDED
:
```

Figure 4. Method description for a dendrimer assembly using a linear sequence of monomers and describing how they connect to each other through a “connectivity table”.

Our python toolkit allowed a fast generation of the sequence and the “connectivity table” by using essential information only in an interactive way and resulting in a pdb file with preserved monomer names . Using a classical example, we firstly demonstrate the use of this toolkit on the assemble of PAMAM dendrimers (see Figure 5). Since XPLOR requires the topology and parameters of the monomers, these had to be generated in the first instance. PDB files of monomers were first generated in Avogadro and used as input on XPLO2D [44] to automatically generate a generic topology and parameters file. Simultaneously, monomers were used as input in ParamChem [45] to assign atom types, charges and bonded parameters of the CHARMM General Force Field (CGenFF). The topology and parameter files were then modified with the ParamChem data. Furthermore, patch references were manually assigned for individual connectivity (all functional groups) between monomers with the parameters given by ParamChem.



python
Generate
Restart
Add New Layer
Add new layer
Add charmm termina
Core name
Multiplicity of the core
Generations to further grow
Patch names from branching monomer to the branching monomer of the next generations
Multiplicity of the branching monomer
Add new layer to the dendrimer
For CHARMM FF add terminals (e.g. carboxyl or amine terminal)
Name of the branching monomer as described in the topology file
Patch names according to the multiplicity of the core
Sequence and connectivity table generated by the application:

ructure:

Core name: EDA Multiplicity: 4

Patch to EDA: CTB1 Monomer name: BMA Generation growth: 1 Multiplicity: 2 Patch to itself: BTB1 Patch to itself: BTB2

Patch to EDA: CTB2 Monomer name: BMA

Patch to EDA: CTB3 Monomer name: BMA

Patch to EDA: CTB4 Monomer name: BMA

```
sequence EDA BMA BMA BMA BMA BMA BMA BMA BMA BMA BMA BMA BMA end
```

generation 0
generation 1
Topology file
Parameter file
XPLOR interpretation
Pdb file
PSf file
EDA
BMA
CTB1
CTB2
CTB3
CTB4
BTB1
BTB2

```
Patch CTB1
 reference=-=(
residue 1 )
 reference=+=(
residue 2 )
end

Patch CTB2
 reference=-=(
residue 1 )
 reference=+=(
residue 3 )
end

Patch CTB3
 reference=-=(
residue 1 )
 reference=+=(
residue 4 )
end

Patch CTB4
 reference=-=(
residue 1 )
 reference=+=(
residue 5 )
end

Patch BTB1
 reference=-=(
residue 2 )
 reference=+=(
residue 6 )
end

Patch BTB2
 reference=-=(
residue 2 )
 reference=+=(
residue 7 )
end

Patch BTB1
 reference=-=(
residue 3 )
 reference=+=(
residue 8 )
end

Patch BTB2
 reference=-=(
residue 3 )
 reference=+=(
residue 9 )
end

Patch BTB1
 reference=-=(
residue 4 )
 reference=+=(
residue 10 )
end

Patch BTB2
 reference=-=(
residue 4 )
 reference=+=(
residue 11 )
end

Patch BTB1
 reference=-=(
residue 5 )
 reference=+=(
residue 12 )
end

Patch BTB2
 reference=-=(
residue 5 )
 reference=+=(
residue 13 )
end
```

Figure 5. Example for PAMAM G1 dendrimer generation using the GUI application to generate the sequence and connectivity table and how XPLOR interprets this information to generate the final 3D structure.

The sequence of monomers and the connectivity table were then created using our toolkit (Figure 5). The toolkit requires the name of the monomers and patch references, as described in the topology file, as well as the multiplicity of the monomers. A standard protocol goes as follows:

1. The first input command requires the name of the first monomer (described as the core) as given in the topology file and its multiplicity (for example 4 in the case of

PAMAM core) to which individual patch reference names should be provided. Then press New Layer.

2. Depending on the multiplicity provided in step 1, new entry boxes will open to match the multiplicity, each patch reference between the core and the branching monomers should have individual names since patches between monomers are described individually in the topology file because atoms are deleted and bonded individually.
3. After, the name of the branching monomer is provided, which in the case of the PAMAM dendrimers is the same for all branching points of the core (in this example named BMA, Figure 5). Since the toolkit recognizes all the branches monomers as the same name, it will only open a new entry for all of them, asking for the number of generations they are intended to grow. If different names are provided (e.g. different monomers used such as in the case of dendrons) it will open entries for each individual monomer and thus allowing for the versatility of building branched polymers or dendrons. It is important to note that the original names of the monomers are preserved in the final output files.
4. If more than one generation is to be grown then a multiplicity input entry pops up. In this case, multiplicity of 2 was given for the branching monomers and thus two new entry boxes appear to provide the patch reference to connecting the next layer.
5. Again, individual patching references should be given for individual functional group of the monomers.
6. The toolkit then produces an input text file for XPLOR containing the normal instructions for XPLOR to assemble a macromolecule and the sequence of monomers and how they are connected automatically (see Figure 5).

This protocol showed versatility in types of dendrimers and dendrons that could be built and the types of force fields to be used as parameterization. In particular, since amino acids were already described in XPLOR format using the CHARMM 22 FF and peptide dendrimers are relevant for biomedical applications, we implemented a feature where the caps for the amino acids (e.g. carboxyl or amine) can be added automatically for the terminal groups. To this, once the terminal layer is described we provided a "add charmm terminal" button that opens a new window for an input entry to specify the terminal cap, as described in the topology file.

Overall, this application extended on our previous approach to build dendrimers from sequence [40] allowing a flexible, fast, easy and reliable way to produce not only dendrimers, but also other linear or branched polymers such as dendrons. Depending on the practical use and type of modelling required, there are several ways where the use of this application is beneficial. Comparing to the original method, we improved the automation by making use of XPLO2D [41] to generate generic (indiscrimination of atom types) topology and parameters files with the Engh & Huber ball-park force field [46] (but lacking atom charges) and generation of the sequence and connectivity table automatically. This has considerably decreased the time necessary to assemble the dendrimer. Since XPLOR does not require a dedicated dendrimer force field just to assemble the dendrimer or dendrons, this means that the assembly can be performed just by updating the charges in the topology file provided by XPLO2D (generic atom charges) and by manually describing the patch references. A major advantage of using XPLOR is that during the handle of the simulated annealing protocols it allows atoms to cross over each other, and thus prevents the formation of entangled branches and, if specified, it can provide a fully extended dendrimer. Then once the dendrimer is assembled, the pdb generated can be imported to other software where force field development can be performed (e.g. MAESTRO, AMBER). On the other hand, the topology and parameter files of the monomers can be modified with the desired force field (e.g. CHARMM, OPLS, AMBER) at this early step generate the dendrimer and then use the same files in NAMD to run MD simulations (Figure 6).

-

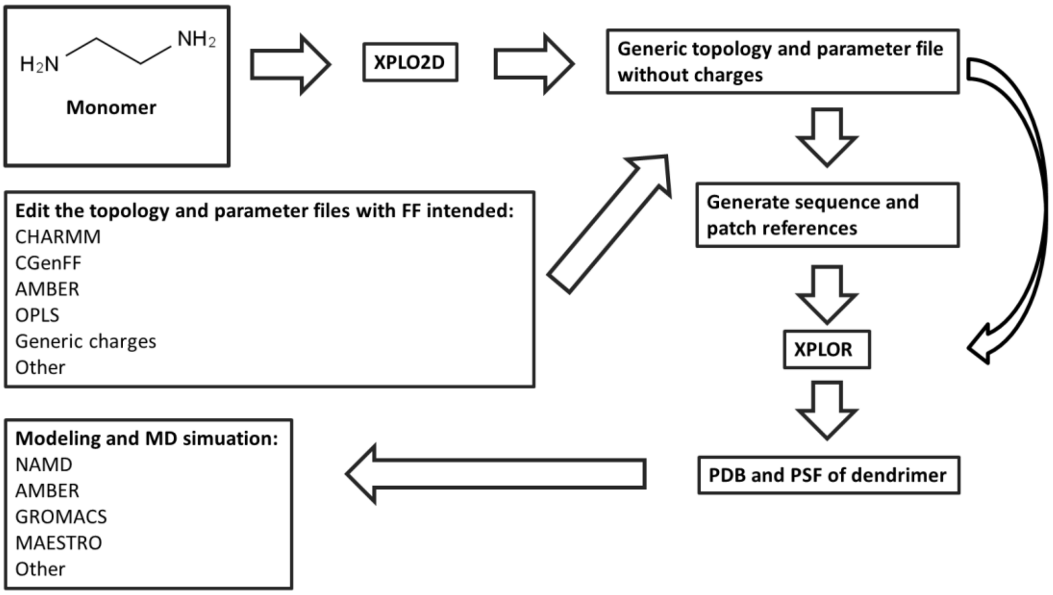

NH2
H2N
Monomer
XPLO2D
Generic topology and parameter file without charges
Generate sequence and patch references
XPLOR
PDB and PSF of dendrimer
Edit the topology and parameter files with FF intended:
CHARMM
CGenFF
AMBER
OPLS
Generic charges
Other
Modeling and MD simuation:
NAMD
AMBER
GROMACS
MAESTRO
Other

Figure 6. Schematic protocol making use of the XPLO2D, XPLOR and our application to quickly and reliably generate dendrimers for MD simulation.

Using this protocol (Figure 6) we were able to apply our scripts and generate the 3D structure of different classes of polymers (Figure 7). As depicted in figure 7, more complex structures could be generated including n-acetyl cysteine terminated PAMAM dendrimers [47] or peptide dendrimers [48] showing the versatility of this tool.

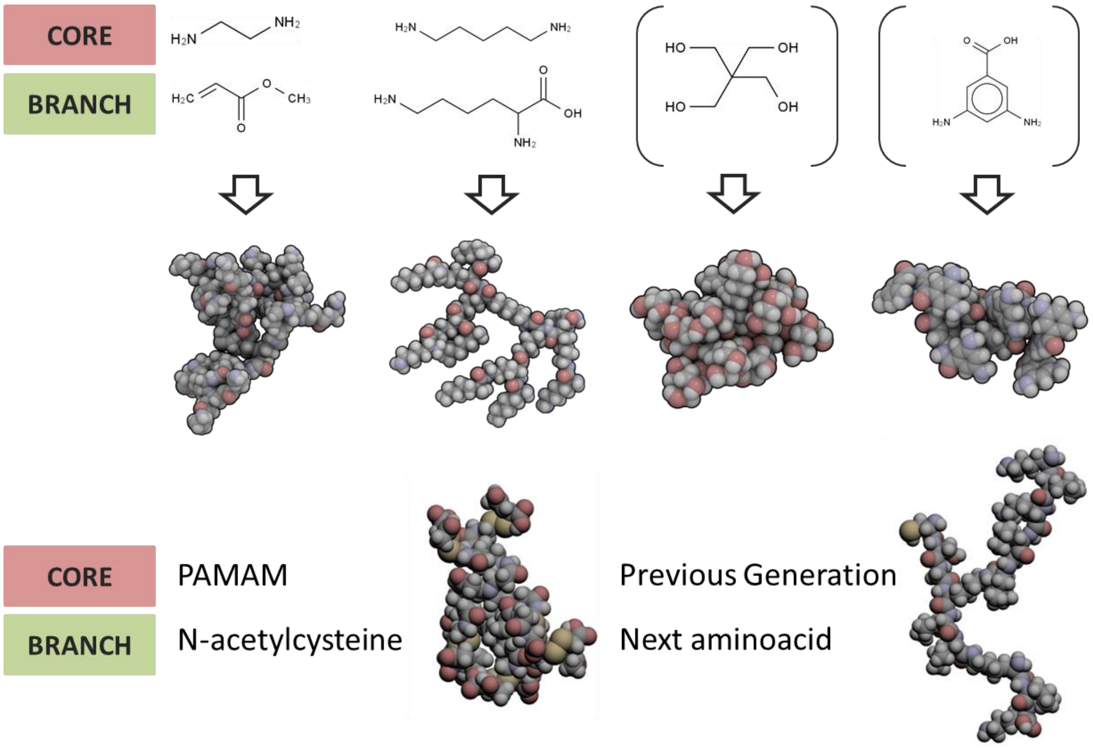


Figure 7. Different G0 and G1 dendrimers generated using our protocol. Even without specific force field development, different classes of dendrimers can be easily assembled to generate a pdb file that can then be subsequently used in modelling studies. More complex structures can be generated including differently terminated structures (N-acetylcysteine terminated PAMAM dendrimers, bottom left) and peptide dendrimers (bottom right) should be generated using toolkit 2.

These structures in a pdb can be readily used for further modelling studies and MD simulations using different software packages, such Desmond [49], NAMD [50] and XPLOR-NIH [41] with minimal modifications of generated structures and preparation of input files. The same files can be used in preparation of systems for simulation using AMBER [51]and Gromacs [52], albeit after generating adequate topologies and parametrization of the monomers and patches. Some of the examples of simulated systems and results of the simulations using Desmond and NAMD software are shown in the supplementary information (Figures S1-S6).

Although we focused on dendrimers, this application can be applied to other types of polymers including linear polymers and dendrons, as long as, a specific defined topology is required. In many respects our tools have similarity with already available tools for building dendrimers such as builder in Material Studio [38], Starmaker [36] and DBT [30] tools. These are also able to generate regular dendrimers but are limited when a molecule is not defined solely by a same monomer in higher generations of branches, in a case of dendrimers, or if one or more branches are completely different in cases of dendrons. Dendrimer Builder Tookit (DBT) is a Perl written graphical user interface that uses the antechamber module of AMBER to assemble the dendrimer with a main limitation as it can use only a maximum of three types of monomers to define a core, branches and terminal groups. This prevents building dendrimers with different building blocks in different generations. Starmaker can be used to overcome such limitations, however the attachment point has to be defined for each monomer that can be a complex task without a GUI.

Our toolkit is not suited for the generation of random copolymers, hyperbranched polymers with random branching and dendrigrafts. These features can be found in molecules builders available in Material Studio or HBP [39] and therefore those resources should be used instead.

Our toolkit is particularly useful in cases when only resources that are freely available to academia and industry can be used to generate 3D structures of dendrimers with different monomers in different generations; or to generate dendrons as it is possible to build a molecule comprising individual branches that have different structures.

CONCLUSION

Two different GUI applications using a Python were written to rapidly and reliably produce structures of dendrimers and dendrons. The first toolkit is intended for an easy use and intuitive way for rapid prototyping of 2D structures for *in silico* methods in chemoinformatics.

The second toolkit is aimed to further improve and automate our previously published method making it easier to produce 3D structures of higher dendrimer generations and different types of branched macromolecules without having to manually write the connectivity table. The major advantage of both applications is their flexibility for the assembly of monomers to obtain structures of dendrimers, dendrons or linear polymers with defined topologies for further molecular modelling studies.

METHODS

Generation of 2D mol files and smiles for fast prototyping and database management. A python GUI was built using the PyQt4 library to generate 2D mol files of dendrimers. The core assembly of this application follows the same principles of dendrimers synthesis-wise where protected monomers (e.g. ester forms of carboxyl groups) are used and subsequently deprotected for the next layer to be added. This interface takes mol files or smiles to describe the monomers and interprets them using the RDKit library package [43]. These were then subsequently reacted by providing the SMARTS reaction description [42] on how to connect both monomers and how to deprotect the monomers for subsequent layer. Once the SMARTS reaction description was mapped onto the monomers, it connected all identified groups and in cases where each group reacted multiple times (e.g. amines), a multiplicity value was provided. In cases where the modification of terminal groups was required (e.g. change of amines to carboxyl) the assembled dendrimer could then be imported again as a mol file, and the mapped groups changed to any other desired group. This application produced 2D mol files that could then be imported to different modelling software packages (e.g. Avogadro, Maestro/Desmond, Gabedit) where parameterization and molecular simulations could be performed. Furthermore, smiles as well as rendered png 2D images of the assembled dendrimers were produced, which can be used for database management of dendrimers.

Generation of 3D pdb and psf files of dendrimers using an automated tool for generation of a linear sequence and connectivity table. The assembly of 3D dendrimers followed a similar protocol previously published by our group [40]. Monomers' topology and parameter files in the XPLOR format were automatically generated using XPLO2D [53], and subsequently manually modified according to the CGenFF by ParamChem [45] or simply by adding charges calculated using the HF/6-31G* ESP method. For peptide dendrimers, the topology and parameters were already described in XPLOR format using the CHARMM22 force field and therefore were used without modification. The connection between monomers was described as patch references by describing the charges, atom types and the angles formed as previously reported [40].

A GUI application was written in Python using the PyQt4 library to produce the XPLOR input file containing the sequence and "connectivity table" of the desired dendrimers. The application takes the name of the core, the core's multiplicity, the name of the patches to connect to the core, the name of the branch, multiplicity of the branch monomers as well as the number of generations to grow. The text file created containing the instructions to XPLOR with the sequence of monomers and the "connectivity table" was used as input in XPLOR-NIH similarly to the previously published method [40]. The 3D structure of dendrimers was obtained as a pdb and psf file, which could be further simulated using Gromacs [52], NAMD [50] or MAESTRO using DESMOND [49].

ACKNOWLEDGMENTS

Nuno Martinho is thankful for the funding from FCT (Fundação para a Ciência e Tecnologia) with a doctoral fellowship (SFRH/BD/87838/2012) and iMed.ULisboa grant (UID/DTP/04138/2013). L.C. Silva acknowledges funding from Investigador FCT 2014 (IF/00437/2014), Portugal. Teresa Barata and Steve Brocchini are grateful for funding from the UK Engineering & Physical Sciences Research Council (EPSRC) for the EPSRC Centre for

Innovative Manufacturing in Emergent Macromolecular Therapies. Financial support from the consortium of industrial and governmental users for the EPSRC Centre is also acknowledged. Steve Brocchini is grateful for funding from the National Institute of Health Research (NIHR) Biomedical Research Centre at Moorfields Hospital NHS Foundation Trust and UCL Institute of Ophthalmology, Moorfields Special Trustees, the Helen Hamlyn Trust (in memory of Paul Hamlyn), Medical Research Council, Fight for Sight and Freemasons Grand Charity. Mire Zloh acknowledges support by University of Hertfordshire.

REFERENCES

1. Svenson S, Tomalia D a. (2005) Dendrimers in biomedical applications - Reflections on the field. Adv Drug Deliv Rev 57:2106–2129. doi: 10.1016/j.addr.2005.09.018
2. Astruc D, Boisselier E (2010) Dendrimers Designed for Functions: From Physical, Photophysical, and Supramolecular Properties to Applications in Sensing, Catalysis, Molecular Electronics, Photonics, and Nanomedicine. Chem Rev 110:1857–1959.
3. Dong S, Zheng B, Wang F, Huang F (2014) Supramolecular Polymers Constructed from Macrocycle-Based Host–Guest Molecular Recognition Motifs. Acc Chem Res 47:1982–1994. doi: 10.1021/ar5000456
4. Kalomiraki M, Thermos K, Chaniotakis NA (2016) Dendrimers as tunable vectors of drug delivery systems and biomedical and ocular applications. Int J Nanomedicine 11:1–12. doi: 10.2147/IJN.S93069
5. Ghosh S, Chakraborty P, Chakrabarti A, et al (2016) Biological activity of dendrimer–methylglyoxal complexes for improved therapeutic efficacy against malignant cells. RSC Adv 6:6631–6642. doi: 10.1039/C5RA23477H
6. Tanis I, Karatasos K (2009) Association of a weakly acidic anti-inflammatory drug (ibuprofen) with a poly(amidoamine) dendrimer as studied by molecular dynamics simulations. J Phys Chem B 113:10984–93. doi: 10.1021/jp9039176

7. Al-Jamal KT, Al-Jamal WT, Wang JTW, et al (2013) Cationic poly- L -Lysine dendrimer complexes doxorubicin and delays tumor growth in vitro and in vivo. ACS Nano 7:1905–1917. doi: 10.1021/nn305860k

8. Hsu H-J, Bugno J, Lee S, Hong S (2017) Dendrimer-based nanocarriers: a versatile platform for drug delivery. Wiley Interdiscip Rev Nanomedicine Nanobiotechnology 9:e1409. doi: 10.1002/wnan.1409

9. Luong D, Kesharwani P, Deshmukh R, et al (2016) PEGylated PAMAM dendrimers: Enhancing efficacy and mitigating toxicity for effective anticancer drug and gene delivery. Acta Biomater 43:14–29. doi: 10.1016/j.actbio.2016.07.015

10. Nguyen TTC, Nguyen CK, Nguyen TH, Tran NQ (2017) Highly lipophilic pluronics-conjugated polyamidoamine dendrimer nanocarriers as potential delivery system for hydrophobic drugs. Mater Sci Eng C 70:992–999. doi: 10.1016/j.msec.2016.03.073

11. Brown CW, Buckhout-White S, Díaz SA, et al (2017) Evaluating Dye-Labeled DNA Dendrimers for Potential Applications in Molecular Biosensing. ACS Sensors 2:401–410. doi: 10.1021/acssensors.6b00778

12. Cai X, Zhu H, Zhang Y, Gu Z (2017) Highly Efficient and Safe Delivery of VEGF siRNA by Bioreducible Fluorinated Peptide Dendrimers for Cancer Therapy. ACS Appl Mater Interfaces 9:9402–9415. doi: 10.1021/acsami.6b16689

13. Barata TS, Teo I, Brocchini S, et al (2011) Partially glycosylated dendrimers block MD-2 and prevent TLR4-MD-2-LPS complex mediated cytokine responses. PLoS Comput Biol. doi: 10.1371/journal.pcbi.1002095

14. Öztürk K, Esendağlı G, Gürbüz MU, et al (2017) Effective targeting of gemcitabine to pancreatic cancer through PEG-cored Flt-1 antibody-conjugated dendrimers. Int J Pharm 517:157–167. doi: 10.1016/j.ijpharm.2016.12.009

15. Jatczak-Pawlik I, Gorzkiewicz M, Studzian M, et al (2017) Sugar-Modified Poly(propylene imine) Dendrimers Stimulate the NF-κB Pathway in a Myeloid Cell Line. Pharm Res

34:136–147. doi: 10.1007/s11095-016-2049-3

16. Arima H, Motoyama K, Higashi T (2017) Potential therapeutic application of dendrimer/cyclodextrin conjugates with targeting ligands as advanced carriers for gene and oligonucleotide drugs. Ther Deliv 8:215–232. doi: 10.4155/tde-2016-0064

17. Li N, Li N, Yi Q, et al (2014) Amphiphilic peptide dendritic copolymer-doxorubicin nanoscale conjugate self-assembled to enzyme-responsive anti-cancer agent. Biomaterials 35:9529–9545. doi: 10.1016/j.biomaterials.2014.07.059

18. Jishkariani D, MacDermaid CM, Timsina YN, et al (2017) Self-interrupted synthesis of sterically hindered aliphatic polyamide dendrimers. Proc Natl Acad Sci U S A 114:E2275–E2284. doi: 10.1073/pnas.1700922114

19. Raut S, Enciso AE, Pavan GM, et al (2017) Intrinsic Fluorescence of Triazine Dendrimers Provides a New Approach to Study Dendrimer Structure and Conformational Dynamics. J Phys Chem C 121:6946–6954. doi: 10.1021/acs.jpcc.6b11110

20. Berman HM, Westbrook J, Feng Z, et al (2000) The Protein Data Bank. Nucleic Acids Res 28:235–42.

21. Zloh M, Ramaswamy C, Sakthivel T, et al (2005) Investigation of the association and flexibility of cationic lipidic peptide dendrons by NMR spectroscopy. Magn Reson Chem 43:47–52. doi: 10.1002/mrc.1508

22. Pinto LF, Riguera R, Fernandez-Megia E (2013) Stepwise Filtering of the Internal Layers of Dendrimers by Transverse-Relaxation-Edited NMR. J Am Chem Soc 135:11513–11516. doi: 10.1021/ja4059348

23. Pinto LF, Correa J, Martin-Pastor M, et al (2013) The Dynamics of Dendrimers by NMR Relaxation: Interpretation Pitfalls. J Am Chem Soc 135:1972–1977. doi: 10.1021/ja311908n

24. Alberto R, Joao R, de los Angeles M-F, et al (2017) Principal physicochemical methods used to characterize dendrimer molecule complexes used as genetic therapy agents,

nanovaccines or drug carriers. Curr Pharm Des 23:1–1. doi: 10.2174/1381612823666170220164535

25. Martinho N, Florindo H, Silva L, et al (2014) Molecular Modeling to Study Dendrimers for Biomedical Applications. Molecules 19:20424–20467. doi: 10.3390/molecules191220424

26. Kim SH, Lamm MH (2012) Multiscale Modeling for Host-Gues Chemistry of Dendrimers in Solution. Polymers (Basel) 4:463–485.

27. Bello M, Fragoso-Vázquez J, Correa-Basurto J (2017) Theoretical studies for Dendrimer-Based Drug Delivery. Curr Pharm Des 23:1–1. doi: 10.2174/1381612823666170228142429

28. Ahmed S, Vepuri SB, Kalhapure RS, et al (2016) Interactions of dendrimers with biological drug targets: reality or mystery – a gap in drug delivery and development research. Biomater Sci 4:1032–1050. doi: 10.1039/C6BM00090H

29. Barata TS, Shaunak S, Teo I, et al (2011) Structural studies of biologically active glycosylated polyamidoamine (PAMAM) dendrimers. J Mol Model 17:2051–60. doi: 10.1007/s00894-010-0907-1

30. Maingi V, Jain V, Bharatam P V., Maiti PK (2012) Dendrimer building toolkit: Model building and characterization of various dendrimer architectures. J Comput Chem 33:1997–2011. doi: 10.1002/jcc.23031

31. Quintana A, Raczka E, Piehler L, et al (2002) Design and function of a dendrimer-based therapeutic nanodevice targeted to tumor cells through the folate receptor. Pharm Res 19:1310–6.

32. Liu Y, Bryantsev VS, Diallo MS, Goddard W a (2009) PAMAM dendrimers undergo pH responsive conformational changes without swelling. J Am Chem Soc 131:2798–9. doi: 10.1021/ja8100227

33. Javor S, Reymond J-L (2009) Molecular dynamics and docking studies of single site

esterase peptide dendrimers. J Org Chem 74:3665–74. doi: 10.1021/jo802743c

34. Razmimanesh F, Amjad-Iranagh S, Modarress H (2015) Molecular dynamics simulation study of chitosan and gemcitabine as a drug delivery system. J Mol Model 21:165. doi: 10.1007/s00894-015-2705-2

35. Roberts BP, Krippner GY, Scanlon MJ, Chalmers DK (2009) Molecular Dynamics of Variegated Polyamide Dendrimers. Macromolecules 42:2784–2794. doi: 10.1021/ma8021579

36. Chalmers DK, Roberts BP Silico: A Perl Molecular Toolkit. In: http://silico.sourceforge.net.

37. Naylor AM, Goddard W a., Kiefer GE, Tomalia DA (1989) Starburst Dendrimers. 5. Molecular Shape Control. J Am Chem Soc 111:2339–2341.

38. Maly M, Mata FJ De, Gómez R, et al (2016) Antiviral mechanism of polyanionic carbosilane dendrimers against HIV-1. Int J Nanomedicine 11:1281–1294.

39. Yu C, Ma L, Li S, et al (2016) HBP Builder : A Tool to Generate Hyperbranched Polymers and Hyperbranched Multi-Arm Copolymers for Coarse-grained and Fully Atomistic Molecular Simulations. Sci Rep 6:1–15. doi: 10.1038/srep26264

40. Barata TS, Brocchini S, Teo I, et al (2011) From sequence to 3D structure of hyperbranched molecules: application to surface modified PAMAM dendrimers. J Mol Model 17:2741–9. doi: 10.1007/s00894-011-0966-y

41. Schwieters CD, Kuszewski JJ, Tjandra N, Marius Clore G (2003) The Xplor-NIH NMR molecular structure determination package. J Magn Reson 160:65–73. doi: 10.1016/S1090-7807(02)00014-9

42. Daylight SMARTS - A language for Describing Molecular Patterns.

43. Landrum G RDKit: Open-source chemoinformatics.

44. Kleywegt G, Jones T (1997) Model-building and refinement practice. Methods Enzymol 277:208–230.

45. ParamChem.

46. Kleywegt G (2005) XPLO2D manual.

47. Mishra MK, Beaty CA, Lesniak WG, et al (2014) Dendrimer Brain Uptake and Targeted Therapy for Brain Injury in a Large Animal Model of Hypothermic Circulatory Arrest. 2134–2147.

48. Kwok A, Eggimann GA, Reymond J-L, et al (2013) Peptide Dendrimer / Lipid Hybrid Systems are Efficient DNA Transfection Generations Reagents : Relationships Highlight the Role of Charge Distribution Across Dendrimer. ACS Nano 7:4668–4682.

49. Bowers KJ, Sacerdoti FD, Salmon JK, et al (2006) Molecular dynamics---Scalable algorithms for molecular dynamics simulations on commodity clusters. Proc 2006 ACM/IEEE Conf Supercomput - SC '06. doi: 10.1145/1188455.1188544

50. Phillips JC, Braun R, Wang W, et al (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26:1781–802. doi: 10.1002/jcc.20289

51. Case DA, Cheatham TE, Darden T, et al (2005) The Amber biomolecular simulation programs. J Comput Chem 26:1668–1688. doi: 10.1002/jcc.20290

52. Berendsen HJC, van der Spoel D, van Drunen R (1995) GROMACS: A message-passing parallel molecular dynamics implementation. Comput Phys Commun 91:43–56. doi: 10.1016/0010-4655(95)00042-E

53. Kleywegt G, Jones T (1998) Databases in protein Crystallograhy. Acta Cryst D54:1119–1131.