The Ducque Manual

Jérôme Rihon[1,*]

[1]KU Leuven, Rega Institute for Medical Research, Medicinal Chemistry, Herestraat 49 - Box 1041, 3000 Leuven, Belgium
[*]*Corresponding author, Maintainer*

Contents

Supervisor : prof. dr. Eveline Lescrinier
Co-promotor : prof. dr. Vitor Bernardes Pinheiro
Co-promotor : prof. dr. Matheus Froeyen

dr. Rinaldo Wander Montalvao for his guidance on the fundamentals of linear algebra.
dr. Charles-Alexandre Mattelaer for his guidance on Quantum Mechanics and without his experimental work, Ducque could have never been conceived.

1 Introduction

> There are no mistakes, only happy little accidents.
>
> Bob Ross

A software for the purpose of building native and synthetic nucleic acid duplexes.

Ducque (IPA : /dʌk/) stands for :

D Acronyms
U Are
C Rather
Q
U
E Tedious

The name was inspired much in the same way that the ORCA Quantum Mechanics package[1,2] was named. At the time of development, I had different name for Ducque. During one of my evaluation moments, one of the members of my jury pointed out that with that name, I would probably not get enough traction, since Google searches gaves millions of results. Given that fact, I tried coming up with a new name for the model builder. Weeks later, as I was working, my hat that was resting on my desk in front of me caught my eye. You see, my hat has a tiny duck embroided on the front. With this idea, I "researched" ways to rewrite the word and get as little search engine results as possible, without degenerating the word itself and keeping it phonetically somewhat correct. Thusly, Ducque was born (a second time).

1.1 Modules

Ducque has five modules the user can access.

1. The **build** module, which is the primary module for to use. This will build us the duplex structures we want to use for further
2. The **transmute** will be used, together with an elaborate input, to convert the ‘pdb‘ file to a suitable input for Ducque to use as a building block. The appropriate format has been settled to be the very simple ‘json‘ format.
3. For testing purposes or just to be able to generate randomised sequences, there is a **randomise** module, which generates a randomised duplex structure based on a set of given inputs.
4. To be able to convert ‘xyz‘ formatted molecule structures to a ‘pdb‘ format, I have included a **xyz_pdb** module. Since I am an avid ORCA user and the ‘xyz‘ format is often outputted, I wanted to make it more managable to format a ‘pdb‘ file for myself during development.
5. To have clickable objects for the standard non-terminal user, there is a simple **GUI** that can be used to employ the following modules : build, transmute and randomise. The reason that the xyz_pdb is not included is because I did not find a nice way to design a simple gui. Secondly, other QM programs might have different outputs of molecule structures, the ‘xyz‘ format can be rather niche and therefor not worth to effort to design a good gui for. Chimera and PyMol can definitely convert from one format to the other if one cannot work through the CLI.

1.2 Package structure

path/to/Ducque

→ **bin/:** contains executable to run Ducque.

→ **docs/:** contains this pdf and its LaTeX sourcefiles.

→ **json/:** contains building blocks requires by Ducque to build duplex sequences. The name for the json file is derived from the inputs of the `--transmute` INPUTFILE. The filename is there for classification and is thereby non-trivial.

→ **puckerdata/:** contains all optimised *xyz* and *pdb* files, needed by `--transmute` to be converted to *json*.

→ **src/:** contains all the Ducque source code.

→ **transmute/:** contains the `--transmute` input files, just as an example. The name of the file itself is trivial.

→ **xyz/:** contains the `--xyz_pdb` input files, just as an example. The name of the file itself is trivial.

→ LICENSE .

→ README.md .

→ setup.sh : required to make Ducque executable.

2 Usage

2.1 Build

```
$ Ducque --build INPUTFILE
      The inputfile is read in and the sequence is built accordingly.
      ' * ' : mandatory
      ' + ' : additional in the GUI

INPUTFILE : [
--sequence SEQUENCE *
  Only valid input in the file just a string of nucleotides (comma-delimited).
      Example: --sequence   dT, dC, dA, dA, dC, dG, dG, dT, dA

--complement COMPLEMENT *
  The complement flag denotes the sequence of the complementary strand.
  A list of nucleotides is also a valid input (comma-delimited).
      Example: --complement homo
      Example: --complement rA, rG, rT, rT, rG, rC, rC, rA, rT

--pdbname PDBNAME
  To prompt the name of the pdbfile for the outputted structure.
  If none is given, this defaults to the name of the INPUTFILE.
        Example: --pdb foobar.pdb
]
GUI : [
--filename FILENAME +
  The name of the INPUTFILE to write to.
]
```

The **Build** module allows to build a structure with a simple query. The *sequence* and *complement* flags are mandated. The *pdbname* allows you to name the produced pdb structure. Defaults to the name of the prompted INPUTFILE.
The GUI module has an added flag *filename*, to which you need to prompt a filename, which is the INPUTFILE that is used to build the pdb structure. Defaults to *random_sequence*.

2.2 Transmute

```
$ Ducque --transmute INPUTFILE
The inputfile is read in and the json file is formatted accordingly.
      ' * ' : mandatory
      ' + ' : additional in the GUI

INPUTFILE : [
--pdb PDB *
  The name of the file of the structure you want to convert to json
      Example: --pdb dna_A.pdb

--chemistry ID *
  The chemistry that defines the given nucleo(s)(t)ide
      Example: --chemistry DNA

--comformation CONFORMATION *
  The conformation that denotes the nucleic acid.
   Allows multiple conformers for a given chemistry.
  Used to name the output json file.
  The complementary strand is then fitted onto the leading strand.
      Example: --conformation 2endo
      Example: --conformation 1-4boat

--moiety MOIETY *
  The moiety that the structure defines. Should either be "nucleoside" or "linker"
      Example: --moiety nucleoside

--bondangles ALPHA, BETA, GAMMA, DELTA, EPSILON, ZETA, CHI *
  The backbone angles, DEGREES (comma-delimited).
      Example: --bondangles 101.407, 118.980, 110.017, 115.788, 111.943, 119.045, 126.013

--dihedrals ALPHA, BETA, GAMMA, DELTA, EPSILON, ZETA, CHI *
  The backbone dihedrals, DEGREES (comma-delimited)
      Example: --dihedrals -39.246, -151.431, 30.929, 156.517, 159.171, -98.922, -99.315
]
```

2.3 Randomise

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4 XYZ → PDB

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.5 Graphical User Interface (GUI)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3 Installation

3.1 Install Ducque

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.2 Environments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

References

[1] Frank Neese, Frank Wennmohs, Ute Becker, and Christoph Riplinger. The ORCA quantum chemistry program package. *J. Chem. Phys.*, 152(22):224108, jun 2020.

[2] Frank Neese. Software update: The ORCA program system—version 5.0. *WIREs Computational Molecular Science*, 12(5), mar 2022.