Daniele Iovinelli

Signed 26/04/2022

Università degli Studi di Siena

DIP. BIOTECNOLOGIE CHIMICA E FARMACIA
DIPARTIMENTO DI ECCELLENZA 2018-2022
UNIVERSITÀ DI SIENA 1240

Ph.D. in Biochemistry and Molecular Biology BiBiM 2.0 XXXIV Ciclo

Structure-based approaches for the rapid identification of tumor microenvironment nutrients, inhibitors and allosteric modulators against soluble and membrane proteins

Ph.D. Supervisor

Prof. Ottavia Spiga

Ph.D. Student

Mr. Daniele Iovinelli

Preface

A solid tumor is a dysfunctional neoplastic tissue characterized by uncontrolled growth and chaotic histological organization and it is composed, in addition to cancer cells, by heterogeneous subsets of non-transformed cells, such as mesenchymal stem cells, fibroblasts, endothelial cells, adipocytes and immune cells, establishing a complex tumor microenvironment (TME) with peculiar structural and biophysical characteristics. The features of the neoplastic parenchyma are well instructed through a complex interplay between cancer and stromal cells, orchestrated by soluble factors, metabolites, extracellular vesicles (EVs), as well as cell-to-cell interaction. All those processes will involve some peculiar proteins interacting with different type of molecule that could be studied by using computational approaches.

1. INTRODUCTION

In recent years, bioinformatics is becoming more and more prevalent especially in pharmaceutical chemistry studies for the discovery of new and more innovative drugs aimed at different types of targets according to the different types of diseases.

Bioinformatics is an interdisciplinary field that develops methods and software tools for understanding biological data, in particular when the data sets are large and complex. As an interdisciplinary field of science, bioinformatics combines biology, computer science, information engineering, mathematics and statistics to analyze and interpret the biological data. Bioinformatics has been used for in silico analyzes of biological queries using mathematical and statistical techniques.

Bioinformatics includes biological studies that use computer programming as part of their methodology, as well as a specific analysis "pipelines" that are repeatedly used, particularly in the field of genomics. Common uses of bioinformatics include the identification of candidates genes and single nucleotide polymorphisms (SNPs), thanks to the most varied genomics sequencing techniques such as NGS sequencing[1]. Often, such identification is made with the aim of better understanding the genetic basis of disease, unique adaptations, desirable properties (esp. In agricultural species), or differences between populations. In a less formal way, bioinformatics also tries to understand the organizational principles within nucleic acid and protein sequences, called proteomics. [1]

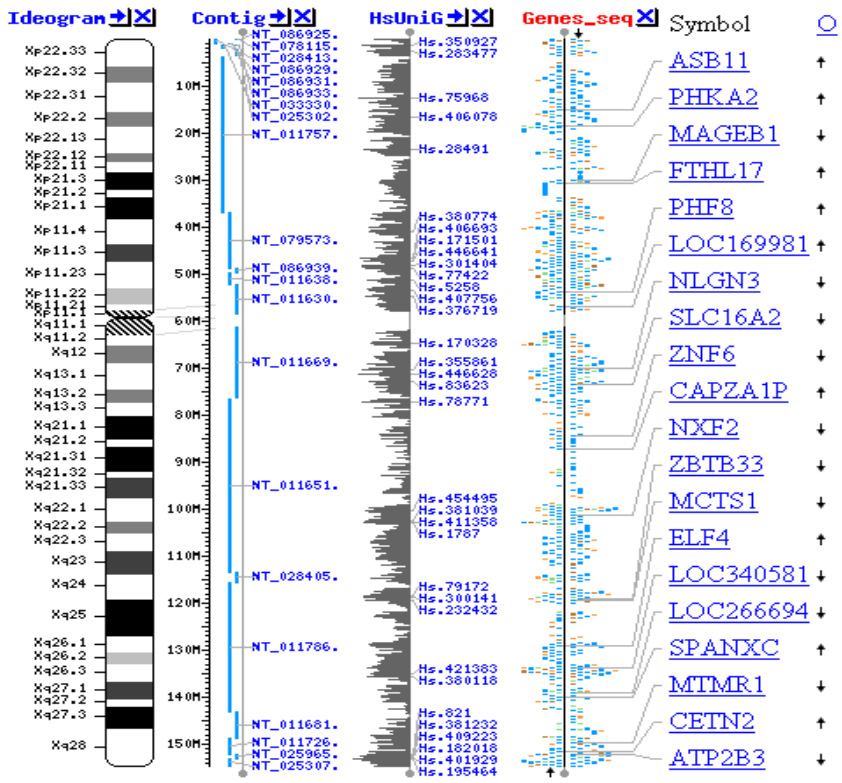

Figure 1. Map of the human X chromosome (from the National Center for Biotechnology Information website)

In general, however, we have different areas of bioinformatics. For example we have the bioinformatics of omics[2], in which the studies are carried out purely on data coming from genomics[1], transcriptomics[1], proteomics[1], metabolomics[1] and other "omics"; structural bioinformatics, in which we study in depth the protein structures , Nucleic Acids structures and all small molecules , known and unknown , and in particular we study the physico-chemical interactions that occur between protein and ligands or between proteins and Nucleic Acids or between proteins themselves. We have other fields of bioinformatics more similar to bioengineering and basic computer science, in which Artificial Intelligence[70] and machine learning[70] are applied, to predict, for example, the druggability of a protein based on the chemical-physical data present in databases of proteins that they are known to be druggable ; protein folding prediction(see Alpha-Fold of Google -

https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery) ; omics big data analysis; classification of tumor images also of real patients ; analysis of human-machine interactions ; precision medicine applied to different diseases, including rare and genetic diseases ; robotics applied to biomedicine ; and much more.

Bioinformatics has become an important part of many areas of biology. In experimental molecular biology, bioinformatics techniques such as image and signal processing allow extraction of useful results from large amounts of raw data.

In the field of genetics, it aids in sequencing and annotating genomes and their observed mutations. It plays a role in the text mining of biological literature and the development of biological and gene ontologies to organize and query biological data. It also plays a role in the analysis of gene and protein expression and regulation. Bioinformatics tools aid in comparing, analyzing and interpreting genetic and genomic data and more generally in the understanding of evolutionary aspects of molecular biology. At a more integrative level, it helps analyze and catalog the biological pathways and networks that are an important part of systems biology. In structural biology, it aids in the simulation and modeling of DNA, [2][5] RNA, [2][5],proteins [4] as well as biomolecular interactions. [5] [6] [7] [8]

One branch of bioinformatics that deals with this is structural bioinformatics. In this sector it is possible to identify, after many steps that we will illustrate, thanks to the knowledge of the receptor or in any case of the protein target, a small number of potential drugs, starting from libraries potentially composed of thousands or hundreds of thousands of molecules or small molecules.

After they have been identified, the best candidate molecules will be subjected to subsequent biological studies, such as "*in vitro*" and "*in vivo*" tests, to confirm the veracity of the computational simulations.

It is therefore useless to underline how important it is today to use bioinformatics and all the tools of which it is composed, to begin a study whose purpose is the research

and discovery of new drugs for a disease that is not yet known, or whose therapy itself is to be improved with new and more selective drugs that avoid the so-called "side effects" that we hear so much in this time of SARS-Cov-2 pandemic.

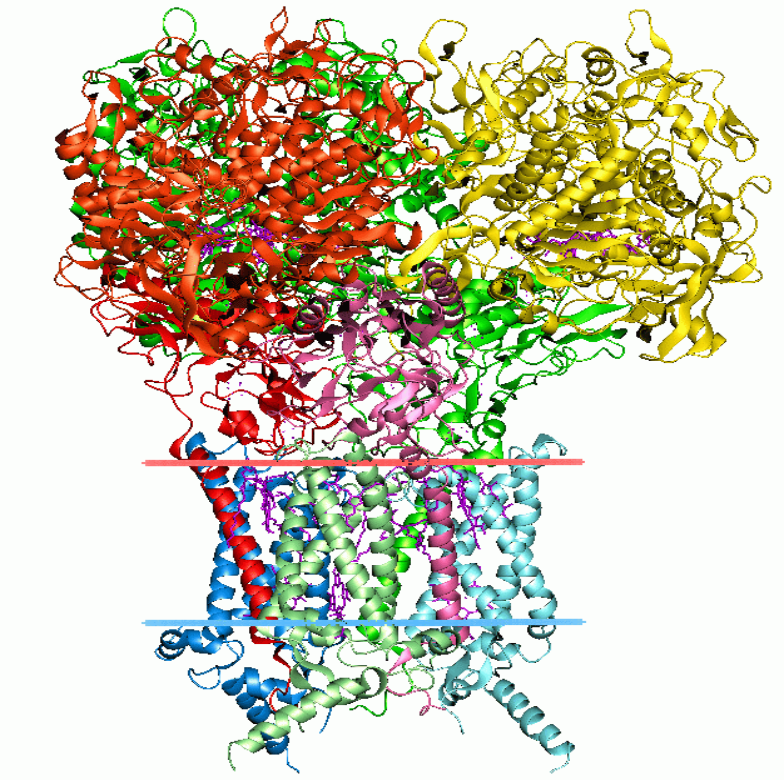

Figure 2. 3-dimensional protein structure of Formate Dehydrogenase (PDB: 1KQF).

Bioinformatics oncology is an integrative discipline that incorporates knowledge from the mathematical, physical and computational fields to promote biomedicine understanding of cancer. Before providing a deeper insight into the bioinformatics approach e utilities involved in oncology, we must understand what a system biology framework is and the genetic connection, due to the high heterogenicity of the background of the people approaching precision medicine. Indeed, it is essential to provide general theoretical information on genomics, epigenomics and

transcriptomics to understand the phases of the multi-omics approach. We consider how to create a multi-omic model. And we have analyzed the new frontiers and the future perspectives of this field.

In the coming age of omics technologies, next gen sequencing, proteomics, metabolomics, and other high throughput techniques will become the usual tools in biomedical cancer research. However, their integrative approach is not trivial due to the broad diversity of data types, dynamic ranges and sources of experimental and analytical errors characteristic of each omics. The multi-omics systematic study of cancer found many different factors involved in the development / maintenance of the malignant state such as genetic aber rations, epigenetic alterations, changes in the response to signaling pathways, metabolic alterations, and many others . The advent of high-throughput technologies has permit ted the development of systems biology. The system biology paradigm tries to analyze cancer as a complex and intricate pathology and to gain insight into its molecular origin by taking into account the different contributions like DNA mutations, deregulation of the gene expression, metabolic abnormalities, and aberrant pathway signaling [5].

The essential basis of systems biology is to consider a biological phenomenon as a system of interconnected elements such as many complex molecular and environmental components interacting with each other at different levels. For example, tumor behavior is determined by a combination of changes in genomic information possibly associated with abnormal gene expression, protein profiles, and different cellular pathways. In this scenario, the complex interaction of DNA and proteins in replication, transcription, metabolic, and signaling networks are considered the decisive causes for cancer cells dis-functioning [5].

The integration of multi-omics data provides a platform to connect the genomic or epige nomic alterations to transcriptome, proteome, and metabolome networks underling the cellular response to a perturbation. Powerful and sophisticated computational tools canidentify the interconnection between genomic aberrations

with differentially expressed mRNAs, proteins, and metabolites associated with cancer-driven cellular perturbation [5].

If on the one hand this aspect provides an opportunity to better study the cellular response, on the other hand it poses a challenge for systems biology-driven modeling. Therefore, the next step of systems biology approach focuses on dynamic models that can deal with thousands of mRNA, protein, and metabolite changes developing effective strategies toadminister personalized cancer therapy [5]. Summarizing, the main goal of the systems biology research driven by multi-omics data is to develop predictive models that are refined by experimental validations in order to select patients based on personalized multi-omics data and stratifying them to determine who are most likely to benefit from targeted therapies [5]. Definition and detection of cancer-distinctive features allow the investigation of the transition process of a normal cell to malignancy. Generally, the hallmarks involve phenotypic and molecular changes in several metabolic pathways such as uncontrolled proliferation by blocking growth suppressors, reprogramming of energy metabolism, evading immune destruction, resisting cell death, angiogenesis, and metastasis [5]. These variations in cellular machinery are driven by molecular aberration in several omics layers such as genome, epigenome, transcriptome, proteome, and metabolome within cancer cells. Specifically, by applying next generation sequencing to cancer cell genomes, it is possible to reveal how mutations in proliferative genes like B-raf drives the activation of mitogen-activated protein- (MAP-) kinase signaling pathway underlying an uncontrolled cell proliferation. Molecular aberrations leading to cancer are involved not only in genomic mutational events but also in the epigenome. In particular, aberrant epigenetic mechanisms can be responsible for silencing of certain cancer suppressor genes [5]. The multistep processes of invasion and metastasis require a transition of epithelial cell toward mesenchymal phenotype to colonize distant sites. Recent studies have revealed that epithelial-mesenchymal transition is induced by specific transcription factors that coordinate the invasion and metastasis processes [5]. By applying transcriptomics techniques it is possible to investigate the

transcription factors involved in transcription regulatory networks assumed to be activated in malignancy. Moreover, manifestations of cancer hallmarks also affected cellular metabolism, in fact tumor cells can reprogram glucose metabolism and energy production pathways detectable with a metabolomics approach [1].

2. MATERIALS AND METHODS

In this and the next few chapters, I will set out to clarify the methods used in different structural bioinformatics studies. In particular, I will illustrate the techniques, the pipelines and protocols used during the projects ,the softwares and tools used for simulations and analysis.

We will analyze the results obtained in several studies recently published in various scientific journals. We will then finally draw conclusions and underline the strengths and weaknesses of bioinformatics.

2.1 BIOINFORMATICS TOOLS AND SOFTWARES

The starting point, which is common to many areas of research, is a detailed literature research about the disease and the receptor involved. In fact, it is necessary to check all the publications of the desease and recepor we want to study. We have to take care if there is a binding site of the protein or in any case an allosteric site and then we have to identify the amino acids of the pocket. If we know a binding site from literature, we need to analyze the molecules that experimentally bind on the receptor

and then carrying out studies of structurally similar molecules, with different chemical substituents, in order to improve the therapy and avoid possible unwanted effects. Furthermore, it is necessary to check whether studies similar to ours have not already been done previously, so as not to repeat studies already done in the past.

After having identified the right bibliography, we move on to practical study. First we have to go and search for the protein target on the most famous databases such as PDB[9] and Uniprot[10]. These platforms present databases of proteins and nucleic acids that have been obtained by X-Ray crystallography[11], NMR[12] or Cryo-EM [13] .

We need to analyze the protein target with Molecular visualization softwares such as PyMol[14] , Chimera [15] , VMD [16] , Jmol [17], etc.

The analysis that must be carried out to check if there are some amino acid "gaps", or if there are missing amino acids in the primary sequence. Usually the vast majority of the "gaps" are found in the "protein-loops", as these are high motility protein folding, so it is possible that the resolution techniques already mentioned above, may not detect some amino acids present in these portions.

2.2 HOMOLOGY MODELING

Therefore, if the gaps are present in the protein , we will have to perform a technique called *"Homology Modeling"* [18]. During this phase we must have the primary sequence of the receptor in FASTA format [19] available, and access to the Blastp website (https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins). In this

website, after pasting the primary squence into the main box, and after selecting the main research database (PDB in our case), the organism on which to carry out the in vivo tests (for example "rattus norvegicus") and others options, we are going to search for a possible "*structuretemplate*" on which processing the Homology modeling. The two most important parameters to consider are the *"query cover"* and the *"identity percentage"*. The protein with the highest values of these parameters or a fair compromise will be selected as our working template.

Subsequently, to carry out the actual homology modeling, having the template and the primary sequence available, we use a PyMol plugin called PyMod [20]. Inside this tool we can use the main software of Homology Modeling which is MODELLER[20] where, after following all the instructions, the protein chain will be generated and all the missing amino acids will also be present.

2.3 STRUCTURAL BIOLOGY SIMULATIONS

After carrying out homology modeling we can move on to the next two steps, namely *Molecular Docking simulations*[21] and *Molecular Dynamics simulations* [22].

By molecular docking we mean the "virtual screening" of the target, where we analyze the interactions and the binding poses of the ligands within the binding site.

The most commonly used docking softwares are Autodock[23] and Autodock Vina[24] for small molecules, and Haddock[25] and Rosetta[26] for larger molecules such as peptides or cyclic peptides.

As mentioned, we usually start with thousands of molecules, and then arrive at two or three candidate molecules. The best 2-3 molecules are selected on the basis of the interaction energy calculated by Autodock Vina (in our case). Then the poses are analyzed on sight, superimposing them on the crystal experimental ligand, if any, for a more accurate study.

The "final" step of an entire structural bioinformatics study is usually a technique called Molecular Dynamics simulation (**MD**) [22], where on the basis of *"Newton's dynamics equations"*[27] and *"quanto mechanics equations"*[28], we analyze the motility of the entire protein-ligand complex. This motility is an index of activity or not of the protein, in fact it is compared with the motility of the protein in the free state. If there are substantial differences between the free state and the complexed state, we can say that amolecule inside the protein has a probable activity. The software most used in the study of molecular dynamics was Gromacs[29].

The Force Fields [30] most used in the studies were Amber[31] and Charmm[32] Forcefields. Especially with regard to charmm, the Charmm-GUIsite [33] was used very frequently. The energy minimizations [34] were almost always set at 5000 steps, the temperature equilibration took place at 303.15 K or at 310.15 K depending on whether the subsequent study was performed "in vitro" or "in vivo".

The pressure was obviously set to 1 Bar. Molecular dynamics were performed on average of 100 nanoseconds (ns), with a writing interval (dt) of 2 femtoseconds (fs).

Some variations of molecular dynamics simulations have also been studied on our study systems. Two types in particular were taken into consideration:

- Steered Molecular Dynamics simulations(**SMD**)[35]
- Supervised MolecularDynamics simulations(**SuMD**)[36]

With the steered molecular dynamics simulations , we can calculate the force of interaction between the ligand and the protein and above all therefore the "*residence time*" of the molecule inside the pocket. To obtain this simulation, a pull force bias must be indicated in the protocol. The average repulsion force applied in the study protocols was on average of 500 KJ mol^{-1} nm^{-2} .

In a SUMD, instead, we try to simulate the approach of a ligand, which has been separated by tens of Angstroms, from the original binding site, and if it can simulate what happens in the experimentally obtained crystal.

The SUMDs studies are still under review, as there is still no well-defined protocol.

If on the one hand, for protein-protein interactions, as simulated for interactions between Spike protein and ACE2 [36] in the article of SARS-Cov-2 [36] , it seems to work properly, on the other hand for protein-ligand interactions, it seems that other parameters , such as background noise of the binding site , time and mode of binding site gating , force interactions between ligand and aminoacids in and out of the binding site , degrees of freedom , and other parameters must be taken into consideration in order to correctly optimize the simulations.

What we are trying to understand with this technique is whether, given an interaction site, a ligand can enter in that site and engage in chemical-physical interactions as it would happen in an experimental crystal. The aim therefore would be to

computationally simulate what actually happens biologically. It is trivial to admit that it is not yet possible to carry out studies of this type with satisfactory answers. In any case, the SUMDs have been studied thanks to an home-made script written in Python programming language [37] in our laboratory.

2.4 SIMULATIONS ANALYSIS PARAMETERS

The parameters that are taken into consideration during the analysis of a molecular dynamics are in particular RMSD[38] , RMSF [39] , Radius of gyration [40] , and the Gibbs free energy[41].

In bioinformatics, the root-mean-square deviation of atomic positions, or simply root-mean-square deviation (RMSD),

$$\mathrm{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^{\,2}}$$

Equation 1. RMSD Calculation

is the measure of the average distance between the atoms (usually the backbone atoms) of superimposed proteins. Note that RMSD calculation can be applied to other, non-protein molecules, such as small organic molecules. In the study of globular protein conformations, one customarily measures the similarity in three-dimensional structure by the RMSD of the Cα atomic coordinates after optimal rigid body superposition.

When a dynamical system fluctuates about some well-defined average position, the RMSD from the average over time can be referred to as the RMSF or root mean square fluctuation. The size of this fluctuation can be measured, for example using Mössbauer spectroscopy or nuclear magnetic resonance, and can provide important physical information. The Lindemann index is a method of placing the RMSF in the context of the parameters of the system.

A widely used way to compare the structures of biomolecules or solid bodies is to translate and rotate one structure with respect to the other to minimize the RMSD. Coutsias, et al. presented a simple derivation, based on quaternions, for the optimal solid body transformation (rotation-translation) that minimizes the RMSDbetween two sets of vectors. [42] They proved that the quaternion method is equivalent to thewell-known Kabsch algorithm. [43] The solution given by Kabsch is an instance of the solution of the d-dimensional problem, introduced by Hurley and Cattell. [44] The quaternion solution to compute the optimal rotation was published in the appendix of a paper of Petitjean. [45] This quaternion solution and the calculation of the optimal isometry in the d-dimensional case were both extended to infinite sets and to the continuous case in the appendix A of another paper of Petitjean. [46]

The Radius of gyration aims to analyze the angles of twisting that occur in the bonds between heavy atoms , such as $C_\alpha$in Proteins , and it is also an index of flexibility and motility of the protein.

In thermodynamics, Gibbs free energy (or Gibbs energy) is a thermodynamic potential that can be used to calculate the maximum reversible work that can be done by a thermodynamic system at constant temperature and pressure. Gibbs free energy

$$\Delta G = \Delta H - T\Delta S$$

Equation 2. Gibbs Free energy equation

(measured in joules in SI) is the maximum amount of work not of expansion that can be extracted from a thermodynamically closed system (which can exchange heat and work with the surrounding environment, but it doesn't matter). This maximum can only be achieved in a fully reversible process. When a system reversibly transforms from an initial state to a final state, the decrease in Gibbs free energy is equal to the work done by the system on the surrounding environment, minus the work of the pressure forces. [47]

The Gibbs energy (symbol G) is also the thermodynamic potential that is minimized when a system reaches chemical equilibrium at constant pressure and temperature. Its derivative with respect to the reaction coordinate of the system vanishes at the equilibrium point. Therefore, a reduction of G is required for a reaction to be spontaneous at constant pressure and temperature.

So finally, if we find a negative energy in our protein-ligand complex, it is possible that the binding occurs spontaneously and therefore that there is an interaction between the ligand and the binding site of the target.

The software used to perform this calculation has been MM/PBSA [48].

2.5 PROGRAMMING LANGUAGES : THEORY AND PROJECTS APPLICATION

A programming language [71] is a formal language comprising a set of strings that produce various kinds of machine code output. Programming languages are one kind of computer language, and are used in computer programming to implement algorithms.

Most programming languages consist of instructions for computers. There are programmable machines that use a set of specific instructions, rather than general programming languages. Since the early 1800s, programs have been used to direct the behavior of machines such as Jacquard looms, music boxes and player pianos. The programs for these machines (such as a player piano's scrolls) did not produce different behavior in response to different inputs or conditions.

Thousands of different programming languages have been created, and more are being created every year. Many programming languages are written in an imperative form (i.e., as a sequence of operations to perform) while other languages use the declarative form (i.e. the desired result is specified, not how to achieve it).

The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning). Some languages are defined by a specification document (for example, the C programming language is specified by an ISO Standard) while other languages (such as Perl) have a dominant implementation that is treated as a reference. Some languages have both, with the basic language defined by a standard and extensions taken from the dominant implementation being common.

Programming language theory is a subfield of computer science that deals with the design, implementation, analysis, characterization, and classification of programming languages.

```c
/*
 * This line basically imports the "stdio" header file, part of
 * the standard library. It provides input and output functionality
 * to the program.
 */
#include <stdio.h>

/*
 * Function (method) declaration. This outputs "Hello, world\n" to
 * standard output when invoked.
 */
void sayHello(void) {
    // printf() in C outputs the specified text (with optional
    // formatting options) when invoked.
    printf("Hello, world!\n");
}

/*
 * This is a "main function". The compiled program will run the code
 * defined here.
 */
int main(void)
{
    // Invoke the sayHello() function.
    sayHello();
    return 0;
}
```

Figure 3. The source code for a simple computer program written in the C programming language. The gray lines are comments that help explain the program to humans in a natural language. When compiled and run, it will give the output "Hello, world!".

During my PhD the most used programming language has been Python[72].

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. [71]

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-

oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. [72]

```
n = int(input('Type a number, and its factorial will be printed: '))

if n < 0:
    raise ValueError('You must enter a non negative integer')

factorial = 1
for i in range(2, n + 1):
    factorial *= i

print(factorial)
```

Figure 4. An example of factorial program written in Python

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. [72] Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. [72]

Python consistently ranks as one of the most popular programming languages.

Python's large standard library, commonly cited as one of its greatest strengths, [73] provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules

for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, [74] manipulating regular expressions, and unit testing.

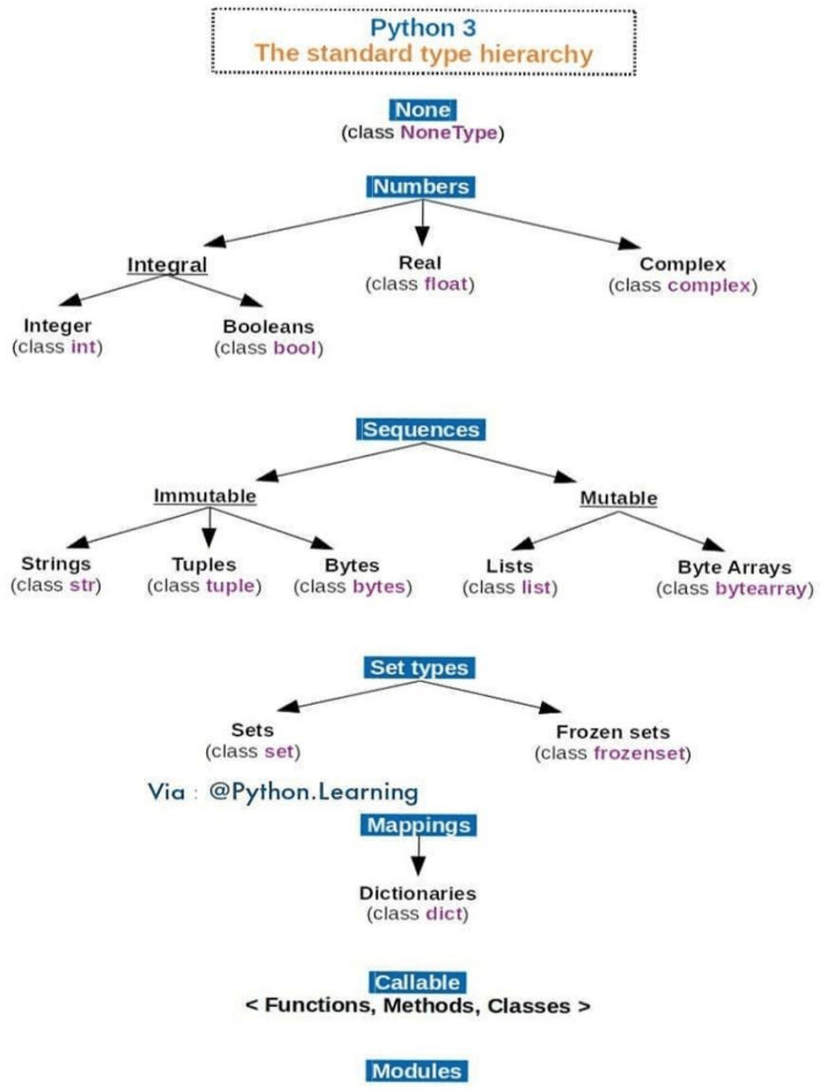

Figure 5. The standard type hierarchy in Python 3– Python Documentation

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333 [75]), but most modules are not. They are specified by their code, internal

documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2021, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 290,000 [72] packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile App
- Multimedia
- Computer networking
- Scientific computing
- System administration

Python can serve as a scripting language for web applications, e.g., via mod_wsgi for the Apache web server. [182] With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used

as a data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

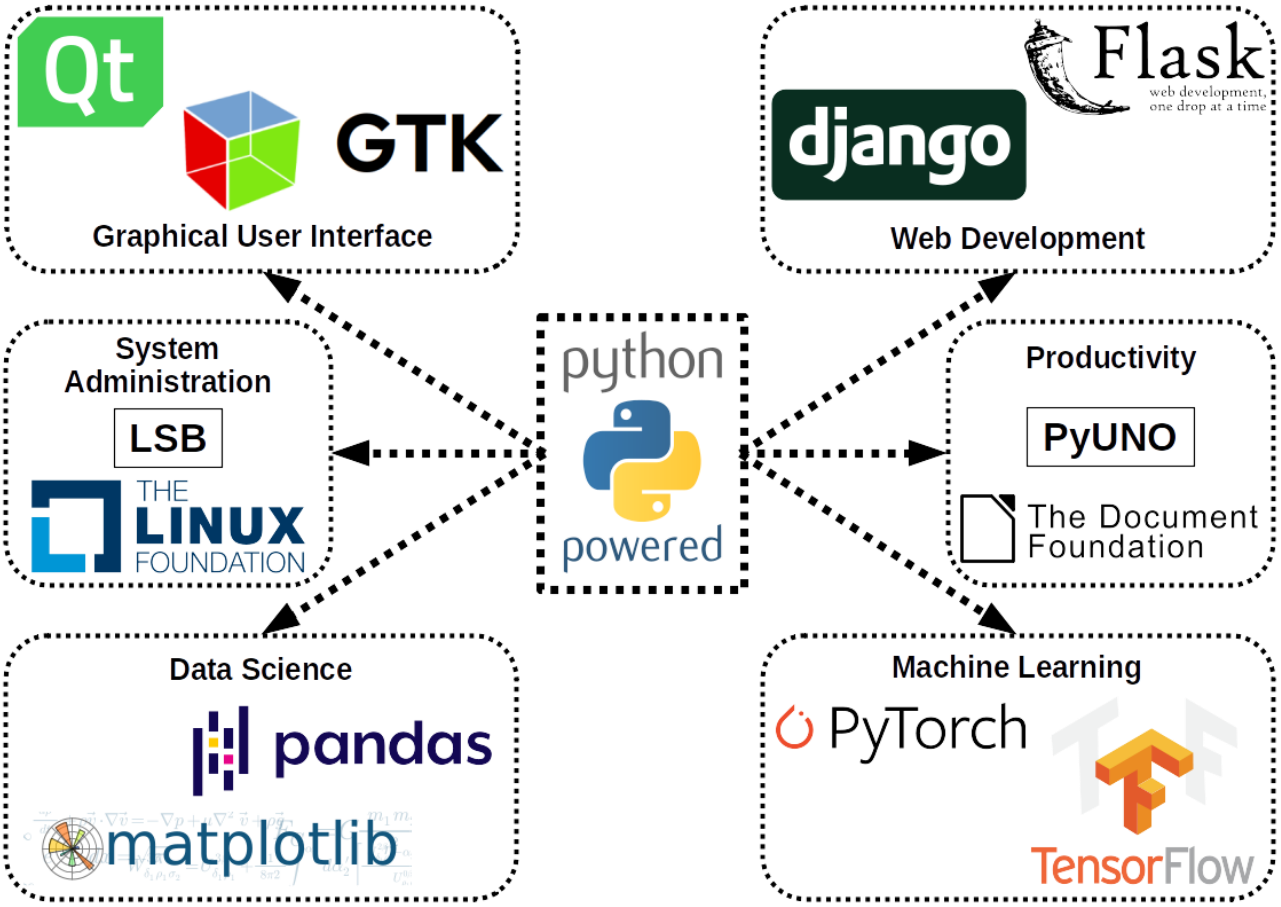

Figure 6. Python Powered - Wikipedia

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a computer algebra system with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number

theory, and calculus. [76] OpenCV has python bindings with a rich set of features for computer vision and image processing. [77]

Python is commonly used in artificial intelligence projects and machine learning projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. [78] . As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing. [71]

Many operating systems include Python as a standard component. It ships with most Linux distributions, [79] AmigaOS 4 (using Python 2.7), FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in exploit development. LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature [81] since Version 4.0 from 7 February 2013.

There is a project of Python , called Biopython [80], which is widely used by the bioinformatics society, especially with regard to the analysis of genomics and genetics, and differend kind of biological data in general.

The Biopython project is an open-source collection of non-commercial Python tools for computational biology and bioinformatics, created by an international association of developers. It contains classes to represent biological sequences and sequence annotations, and it is able to read and write to a variety of file formats. It also allows for a programmatic means of accessing online databases of biological

```
>>> # This script creates a DNA sequence and performs some typical manipulations
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import IUPAC
>>> dna_sequence = Seq('AGGCTTCTCGTA', IUPAC.unambiguous_dna)
>>> dna_sequence
Seq('AGGCTTCTCGTA', IUPACUnambiguousDNA())
>>> dna_sequence[2:7]
Seq('GCTTC', IUPACUnambiguousDNA())
>>> dna_sequence.reverse_complement()
Seq('TACGAGAAGCCT', IUPACUnambiguousDNA())
>>> rna_sequence = dna_sequence.transcribe()
>>> rna_sequence
Seq('AGGCUUCUCGUA', IUPACUnambiguousRNA())
>>> rna_sequence.translate()
Seq('RLLV', IUPACProtein())
```

Figure 7. A core concept in Biopython is the biological sequence, and this is represented by the Seq class. [80] A Biopython Seq object is similar to a Python string in many respects: it supports the Python slice notation, can be concatenated with other sequences and is immutable. In addition, it includes sequence-specific methods and specifies the particular biological alphabet used.

information, such as those at NCBI. Separate modules extend Biopython's capabilities to sequence alignment, protein structure, population genetics, phylogenetics, sequence motifs, and machine learning. Biopython is one of a number of Bio * projects designed to reduce code duplication in computational biology. [5]

```
>>> # This script downloads genomes from the NCBI Nucleotide database and saves them in a FASTA file.
>>> from Bio import Entrez
>>> from Bio import SeqIO
>>> output_file = open('all_records.fasta', "w")
>>> Entrez.email = 'my_email@example.com'
>>> records_to_download = ['FO834906.1', 'FO203501.1']
>>> for record_id in records_to_download:
...     handle = Entrez.efetch(db='nucleotide', id=record_id, rettype='gb')
...     seqRecord = SeqIO.read(handle, format='gb')
...     handle.close()
...     output_file.write(seqRecord.format('fasta'))
```

Figure 8. Through the Bio.Entrez module, users of Biopython can download biological data from NCBI databases. Each of the functions provided by the Entrez search engine is available through functions in this module, including searching for and downloading records.

During the three years of Ph.D. , the main focus was to help our research group, thanks to the automation of many processes and methods, which were previously carried out individually, by hand. My main task was to create pipelines and algorithms in order to allow the automatic operation of many Molecular Docking and Molecular Dynamics simulation studies through, the Gromacs and Autodock Vina XBsoftwares, and the interactions and the retrieval of information and data with the Webthanks to the principal Python Parsing Web Scraping libraries such as **Urllib** [72] and **Selenium**[72] , in order to be able to retrieve a lot of data automatically on the browser. The most frequent procedure, for example, was the automatic generation of the chemical-physical parameters of the protein-ligand complexes, both soluble and membrane complexes , on the CHARMM-GUI website[33].

A very important task was also to create a tabu-like algorithm (shown in the Appendix section) for the study and analysis of Supervised Molecular Dynamics simulations (SuMD) [82].

Finally, another very difficult task was to collaborate with the research group of prof. Niccolai, in order to be able to search for all the possible genes that report an X / Gly mutation [66] and the related disease on the ClinVar Data Base. Subsequently my task was to go and search for each gene mutated in Glycine, through the Python Web Scraping libraries (Selenium and Urllib) [37] , the corresponding protein structure in all possible conformations, on the UniProt and Protein Data Bank databases.

Subsequently, analyzing many factors such as the quality of the protein, expressed in Angstrom units , the length of the primary sequence, the deposition date of the crystal, and other factors, including the elimination of redundancies contained in the different output files , we arrived to a final database of a small number of proteins. All the studies were of course automated with Python programming language.

3. RESULTS AND CONCLUSIONS

During the three years of doctorate we found ourselves facing studies aimed at looking for inhibitors or allosteric modulators, both against soluble and membrane proteins.I will show below all my studies.

In this chapter I am going to present the most important results obtained during my three years of PhD. All of these studies have been published in various scientific journals.

I would like to thanks in this occasion all the members of our Laboratory and in particular Ottavia Spiga , Neri Niccolai, Prischi Filippo , Annalisa Santucci, Lorenza Trabalzini , Alfonso Trezza , Francesco Pettini , Anna Visibelli , Vittoria Cicaloni , Andrea Bernini , for their collaboration in the different projects , and all the other professors, students and researchers of our department who collaborated in our studies.For any detail I refer you to references section.

3.1 SARS-Cov-2 reversible inhibitors

SARS-CoV-2 RBD [49] and hACE2 binding is mostly driven by polar interaction, with an overall~900 $Å^2$ buried surface area. A close analysis of the interface reveals the absence of cavities on RBD in the interaction surface. We performed MD simulations to account for the protein conformational fexibility and detected 1,029 transient pockets. Based on the druggability features of the cavities, i.e. volume,

depth, polarity, and proximity to the hACE2 binding site, we detected a cluster of 9 transient pockets. In order to identify possible PPI inhibitors the transient pocket that contained key residues involved in hACE2 recognition and binding (Fig. 3A) was selected and used for the virtual screening of 1582 FDA-approved drugs.This curated library of drugs, whose structures were freely downloaded from DrugBank[50], represents a reservoir of bioactive molecules that could be repurposed for COVID-19 treatment relatively fast, due to their pre-existing approval for use in humans.

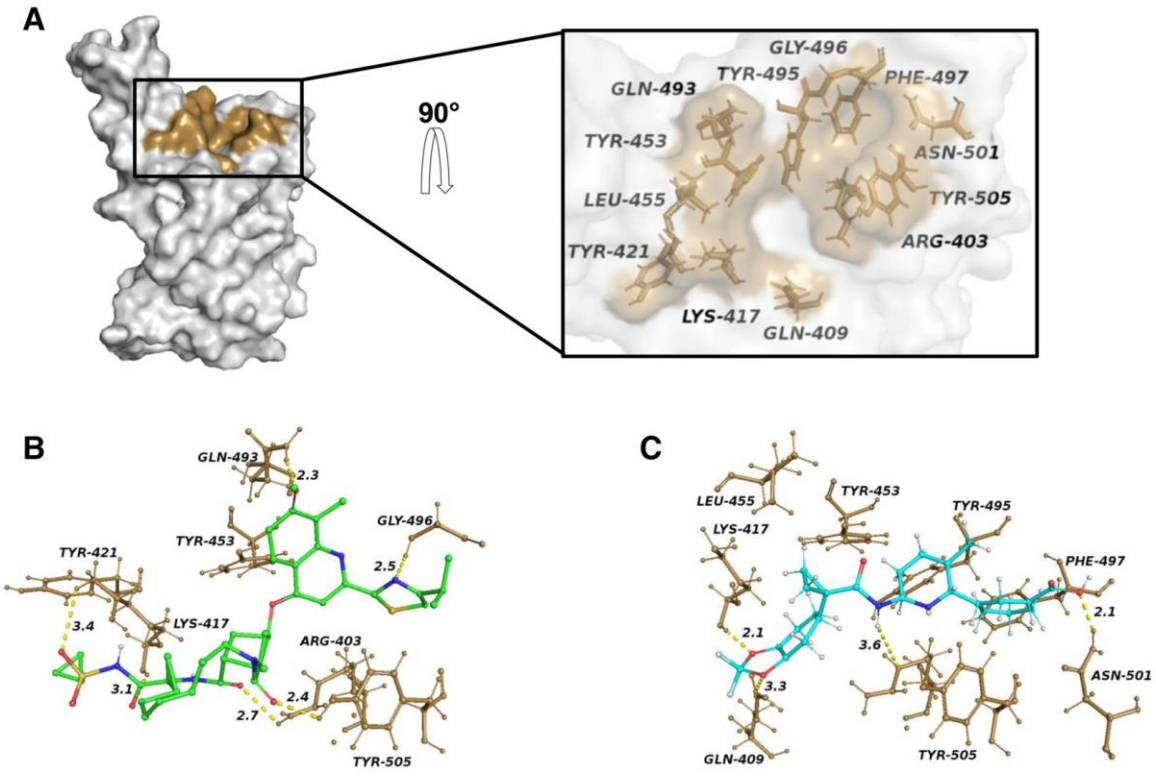

Figure 3. RBD binding pocket and drugs bindg site. (A) Surface representation of the structure of the RBD of the S protein having an open pocket conformation. Te transient pocket surface patch is depicted in brown. In the zoomed region it is possible to see a detailed structural representation of the open pocket conformation. Residues laying on the pocket surface have been labelled and are shown in stick. (B, C) Structural representations of the (B) RBD-Simeprevir and (C) RBD-

Lumacafor complexes resulting from docking simulations. Residues forming direct interactions with the drugs are shown as brown sticks. Hydrogen bonds are indicated with yellow dashed lines

The 10 best compounds (Lumacafor, Paritaprevir, Dihydroergotamine, Trypan blue, Midostaurin, Dihydroergotoxine, Simeprevir, Lurasidone, Spinosyn D, Olaparib) showed high binding free energy scores (− 9.4 to − 8.5 kcal/mol). The compound with the highest binding energy (− 9.4 kcal/mol) was Lumacafor, a CFTR corrector that traffic the mutant protein to the plasma membrane. An analysis of the quality of interactions of the 10 best compounds revealed that Simeprevir had the highest number of polar bonds with side chains of residues in the RBD binding pocket. Simeprevir, a second-generation HCV NS3/4A protease inhibitor [51] , has been reported to be both a SARS-CoV-2 main protease inhibitor [52] and a S protein-RBD interaction inhibitor [53]

In order to understand if Simeprevir and Lumacaftor are able to interfere and prevent the binding between the S glycoprotein and ACE2, we ran a Supervised Molecular Dynamics (SuMD) simulations. Using SuMD it is possible to simulate the full binding process of ACE2 to RBD in presence of Simeprevir or Lumacaftor in an unbiased way (i.e. independently from starting relative positions), taking into account hydration patterns and drug binding-unbinding events. We first validated the SuMD protocol by simulating the binding process of RBD with ACE2. The resulting relative position of ACE2 bound to RBD is comparable to that in the crystal structure.

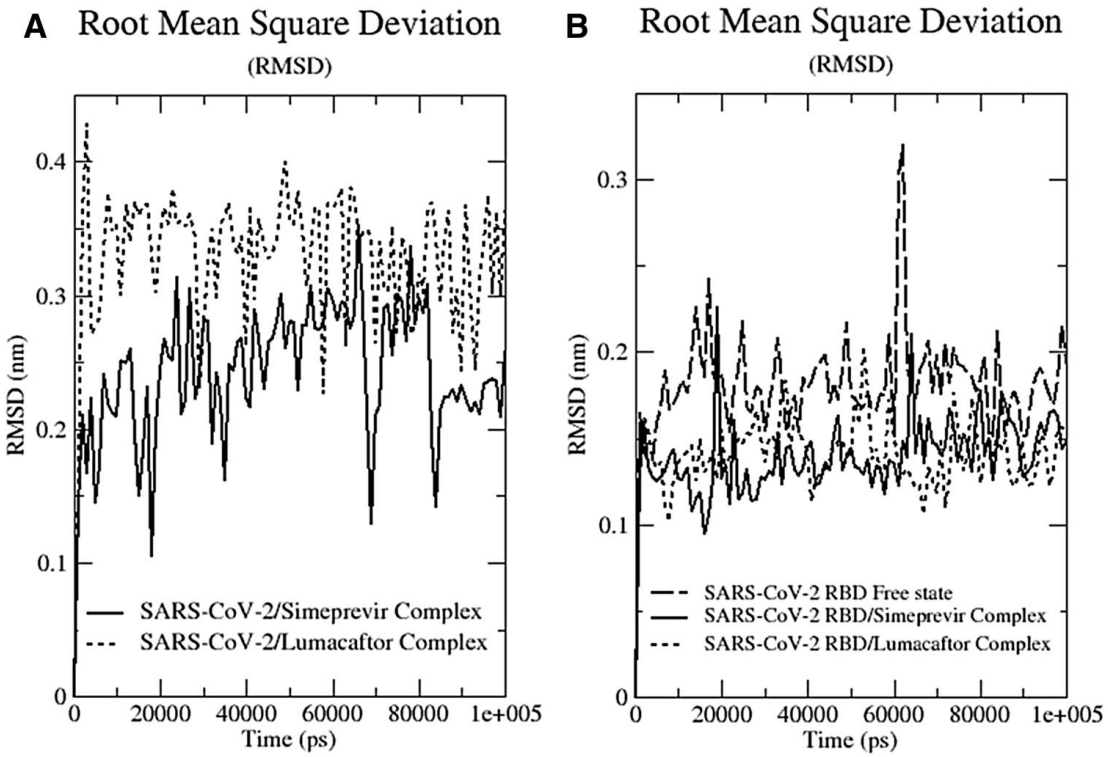

Figure 4. Root mean square deviation (RMSD) plots. (**A**) The RMSD profile of drugs and protein backbone,(**B**) relative to the initial frame against simulation time.

During the SuMDs drugs were allowed to move and find a more energetically favourable pose in the binding pocket. We noticed very limited movements of Simeprevir and Lumacafor and, to confrm binding stability, we performed 100 ns cMD simulations(Fig. 4) of RBD alone and in complex with the drugs.

To quantify the strength of the interaction between Simeprevir and Lumacafor on RBD, we computed the interaction energy between the protein and the two drugs. Te total interaction energy for Simeprevir and Lumacafor was − 75.58± 4.2 kJ/mol and − 63.42± 13.8 kJ/mol respectively. Taken together these data suggest that Simeprevir and Lumacafor bind spontaneously to the target with high affinity.

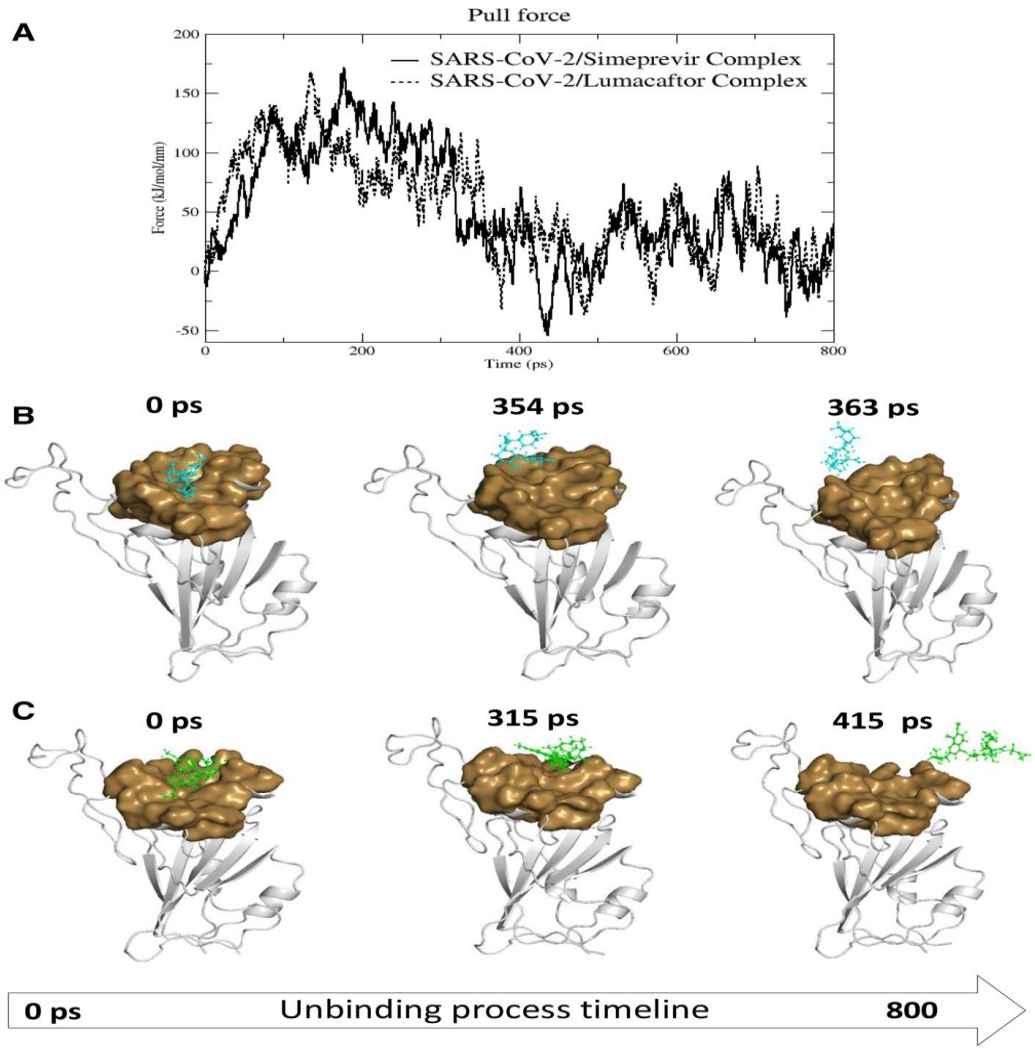

Figure 5. Steered Molecular Dynamics simulations. (**A**) Force profiles of drugs pulled out of the RDB transientpocket along the unbinding pathway, Lumacaftor (dotted line) and Simeprevir (continuous line). (**B**, **C**)Structural representations showing position of Lumacaftor (cyan ball-and-stick) and Simeprevir (green balland-stick) on RBD (white cartoon) during the different stages of the unbinding process from the RBD binding
pocket (brown surface).

To further characterise the recognition process of the two drugs to the S glycoprotein we performed Steered Molecular Dynamics (SMD) simulations. We ran a 800 ps SMD simulation on RBD in complex with both Simeprevir and Lumacafor, and the time-averaged force profles during the unbinding simulation of complexes is shown in Fig. 5A.

Both drugs have a steady increase of the applied forces on the first ~ 150 and ~ 200 ps of the simulation, respectively for Lumacaftor and Simeprevir, until they reach the maximum, which corresponds to the rupture force of Lumacaftor and Simeprevir unbinding along this dissociation pathway. The force then quickly decreases and stays constant until the end of the simulation. In the first step, between 0 and 315 ps of the simulation for Simeprevir and 0 and 354 ps for Lumacaftor, the two drugs slowly detach and move away from the transient pocket; in the second step, between 316 and 800 ps of the simulation for Simeprevir and 355 and 800 ps for Lumacaftor, they move away from the protein and enter the solvent region (Fig. 5 B,C). The comparable rupture forces reflect similarity in the unbinding from RBD in line with our binding energy data.

3.2 Natural h-ERG Channels blockers

The cardiac action potential is regulated by several ion channels. Drugs capable to block these channels, in particular the human ether-a-go-go ` -related gene (hERG) channel[54], also known as KV11.1 channel, may lead to a potentially lethal ventricular tachyarrhythmia called “Torsades de Pointes”. Thus, evaluation of the hERG channel off-target activity of novel chemical entities is nowadays required to safeguard patients as well as to avoid attrition in drug development. Flavonoids, a large class of natural compounds abundantly present in food, beverages, herbal medicines, and dietary food supplements, generally escape this assessment, though consumed in consistent amounts. Continuously growing evidence indicates that these compounds may interact with the hERG channel and block it. The present review, by

examining numerous studies, summarizes the state-of-the-art in this field, describing the most significant examples of direct and indirect inhibition of the hERG channel current operated by flavonoids. A description of the molecular interactions between a few of these natural molecules and the Rattus norvegicus channel protein, achieved by an in silico approach, is also presented.

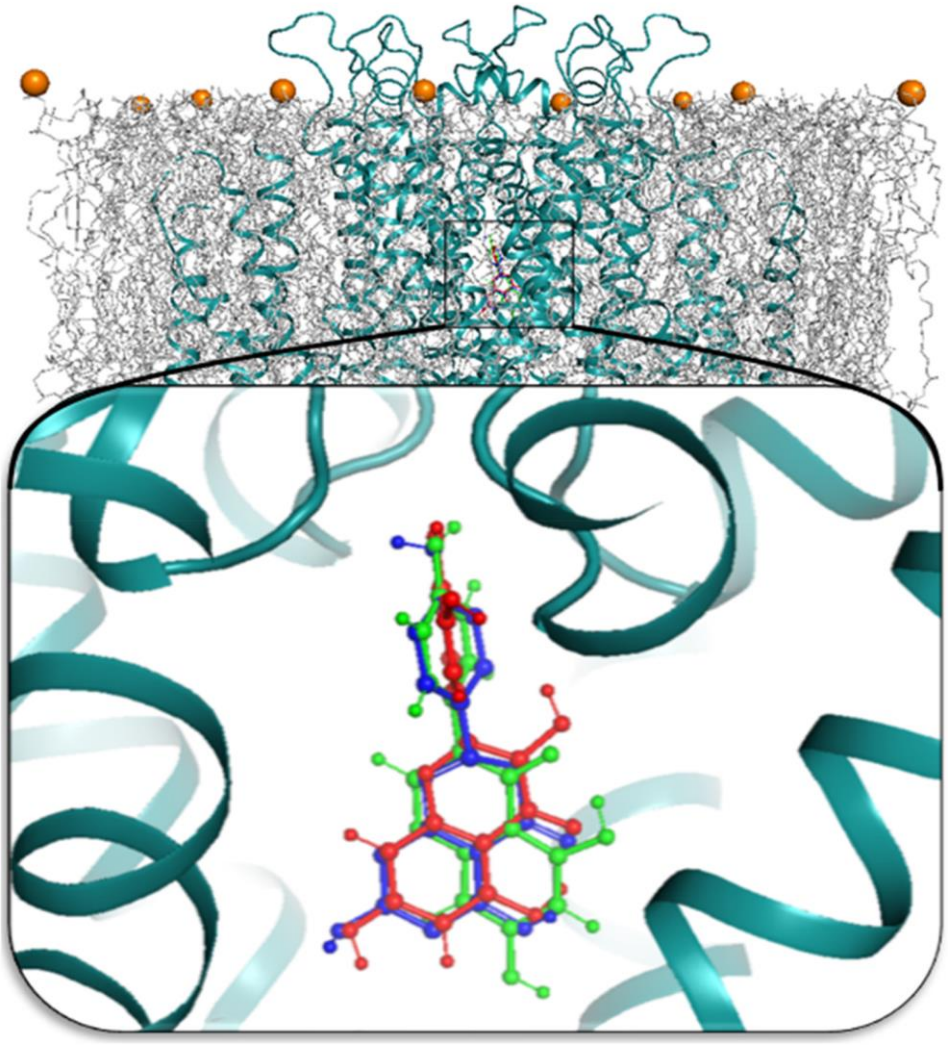

Figure 6. Best docked poses of genistein, (±)-naringenin, and quercetin. Top view: the KV11.1 channel is depicted in deep teal cartoon. Bottom, enlarged view of the small central cavity with the binding poses of genistein, (±)-naringenin, and quercetin represented in blue, green, and red sticks

and balls. The molecules bound within the same KV11.1 binding pocket, sharing a very similar binding pose.

The results were analysed by using GROMACS 2019.3 package and displayed with GRACE. PyMOL2.3 (Schrodinger, ¨ USA) was used as molecular graphical interface. Docking results exhibited different potential binding poses for each compound, though all compounds located in the same binding region. The best pose for each ligand, generated by Autodock/VinaXB (Fig. 6), was selected according to the affinity to and interaction network on the target, taken as initial conformation for molecular dynamics simulations. The structural stability of each biological system was determined from the deviation produced during the simulation runs, which resulted inversely related to the deviation extent. The Root Mean Square Deviation (RMSD) profiles, reported in Fig. 7A, support the validity of the molecular dynamics simulation protocol, which allows to exclude computational artefacts. In Fig. 7B, the RMSD profiles related to ligands inside the KV11.1 binding pocket are reported. RMSD trends showed that all compounds achieved a good stability along the entire molecular dynamics run, thus confirming the strength and correctness of the selected binding poses. The molecular mechanics energies combined with the Poisson–Boltzmann - or generalized Born - continuum solvation surface area (MM/PBSA and MM/GBSA) give rise to methods commonly

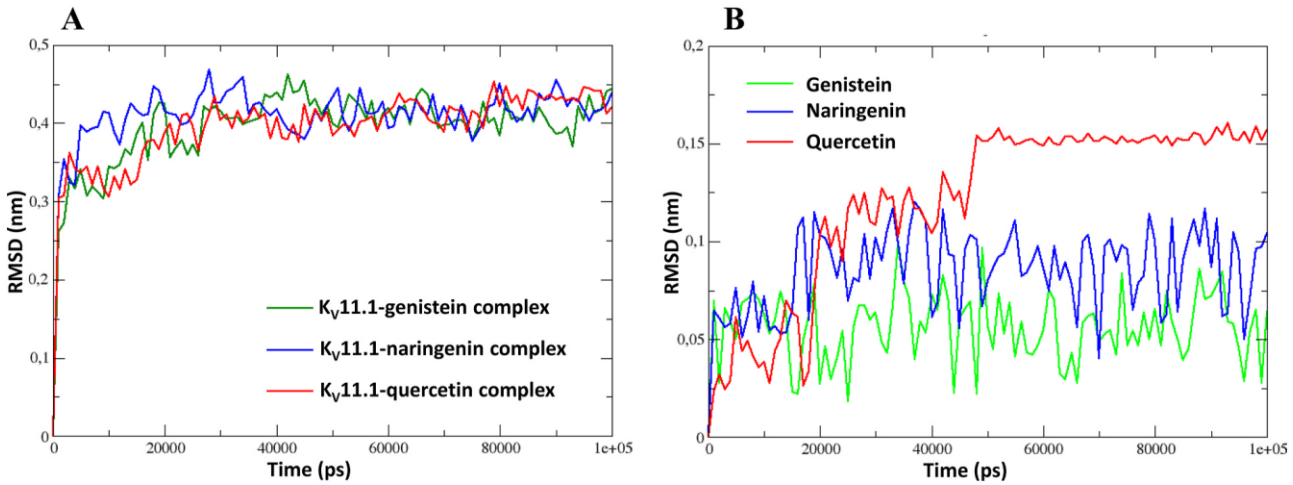

Figure 7. Molecular dynamics simulation root mean squared deviation profiles for the KV11.1 channel-best ligand complexes. Root mean squared deviation (RMSD) values (nm) and time of the molecular dynamics run (ps) are reported on y and x axis, respectively. (A) RMSD of KV11.1 channel in complex with its ligands. All complexes achieved a good stability after 25 ns. (B) RMSD of ligands within the KV11.1 channel binding pocket showed to be stable during molecular dynamics run, confirming a stable binding pose. The complex RMSD profiles are indicated with coloured lines.

used to estimate the free energy of binding of small ligands to macro-molecules. In the case of the MM/PBSA method, the free energy of a certain state arises from the following equation [55]:

$$G = Ebnd + Ecl + EvdW + Gpol + Gnp – TS$$

Equation 3. MM/PBSAFree Energy Calculation[55]

where the first three are standard MM energy terms from bond (bond, angle, and dihedral), electrostatic, and van der Waals interactions; Gpol and Gnp are the polar and non-polar contributions to the solvation free energy. Gpol is normally obtained by solving the Poisson–Boltzmann equation, whereas Gnp is estimated from a linear

relationship to the solvent accessible surface area (SASA). Thereafter, contributions of solvent molecules are removed from each snapshot, since the PBSA solvent model already takes into account the solvation energy. This may lead to some margin of error in the estimation of the binding energy, as both the simulation and energy calculation do not use the same energy function, thus requiring reweighting of the final energy values[56]. We used MMPBSA method to compute the binding free energy of flavonoid ligand-channel complexes. The large binding free energy values of KV11.1 channel-flavonoid complexes for genistein (− 71.50 kcal/mol), (±)-naringenin (− 83.47 kcal/mol), and quercetin (− 93.29 kcal/mol), calculated by the MM/PBSA method, indicate that these flavonoids bind spontaneously inside the pocket of the channel with high affinity. Finally, the energy contribution per residue was also evaluated: the pore helical Ser-626 (segment S5), Tyr-654, and Phe-658 (segment S6) of the Rattus norvegicus KV11.1 channel bound the flavonoids with high energy(Fig. 8), in agreement with previous findings. In fact, several authors [57,58 ,59] demonstrated that the pore helical Ser-624 (segment S5), Tyr-652, and Phe-656 (segment S6) of the human KV11.1, corresponding to the above mentioned Rattus norvegicus residues , are crucial for KV11.1 channel activity. This provides the molecular mechanism accounting for the ability of flavonoids to modulate rat hERG channel current. Since these flavonoids bind inside the pocket of KV11.1 channel with high affinity and good stability, it may be of great interest to investigate by the co-docking procedure (Kadioglu et al., 2016) at hERG channel from one side, and by recording KV11.1 current in HEK293 cells incubated with both flavonoids and drugs known to block hERG channel on the other, the mutual interplay of these agents at this pharmacological target. Taken together, these findings provide the molecular basis to elucidate the activity of flavonoids at the hERG channel. This model, which coincides by a 95.4% of sequence identity and a cover of 72% with the human hERG channel, may serve to flag unwanted effects in humans caused by natural compounds ingested as dietary supplements.

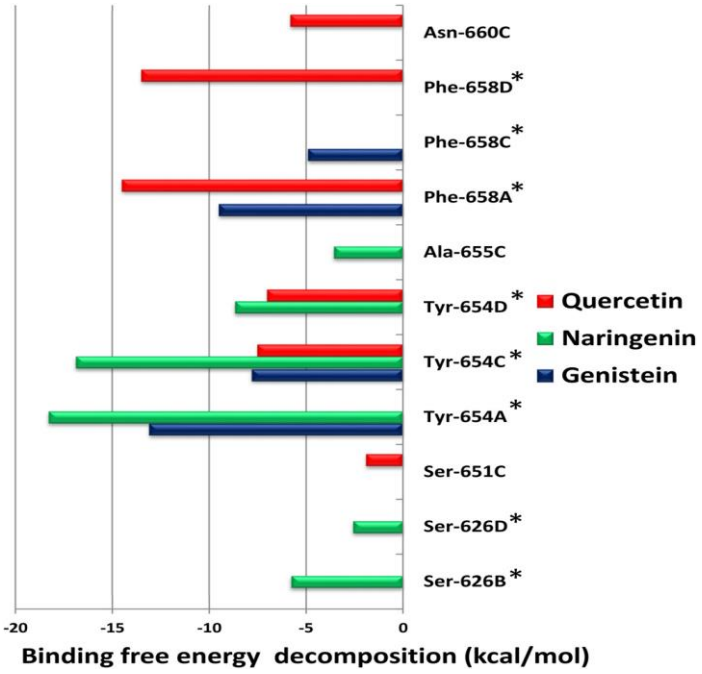

Figure 8. KV11.1 channel binding residues-ligand binding free energy decomposition. The energetic contribution (cal/mol) of KV11.1 channel binding-residues with its ligand is reported on the ordinate scale. Crucial binding residues for KV11.1 channel biological activity are marked with a star. All ligands showed a large interaction network within the KV11.1 channel binding pocket, establishing bonds with high energy.

3.3X/Gly MUTATIONS AND TRANSIENT POCKETS FORMATION

It has been recently suggested that amino acid replacements with Gly can modify the shape of protein surfaces and, hence, protein dynamics and functions. We have browsed ClinVar[60], the database of all the reported variants of clinical relevance, to identify all the proteins having X/Gly mutations that determine Mendelian disorders. We have found 959 benign and 875 pathogenic X/Gly substitutions. Pathogenicity origins were initially searched in the distribution profiles of replaced amino acids. These profiles indicated that Mendelian disorders that include Gly replacement arise

mainly from substitutions of hydrophobic amino acids that reduce protein core stability. In the case mutated proteins were structurally defined, we could give a deeper insight on pathogenicity mechanisms, as Gly-mutants might modify protein surfaces thus interfering with physiological protein-protein interaction processes. In many cases, we have found that pathological Gly-mutants exhibit protein surfaces with additional pockets. These new pockets (Fig. 9) could be the target of a pharmacological strategy for Mendelian disorder remediation.

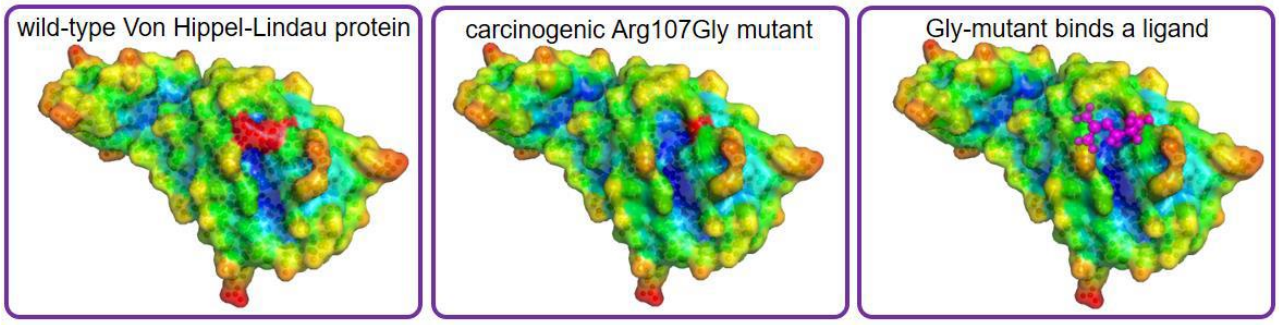

Figure 9. Wild type, carcinogenic mutant and ligand complexed carcinogenic mutant VHL protein. PDB : 4AJY

ClinVar is a very large public database of reported associations between human variants and phenotypes. To select the relevant information for the present study, we applied the following filters through the web interface of ClinVar: “type of variation” = “single nucleotide variant”; “molecular consequence” = “missense”; “review status” = “at least one star”. Then, by applying the “clinical significance” filter, we extracted two separate mutation data sets with “benign” and “pathogenic” filter values. Finally, we selected only those mutations that lead to X/Gly substitutions, in both benign and pathogenic data sets.

We have searched for X/Gly missense mutations among all the items collected in ClinVar variants. ClinVar, an online daily updated service that collects variants directly or from other humanmutation databases, as of October 3, 2020, reported 25,579 benign and 22,901 pathogenic missense mutations. By parsing the information

contained in the latter ClinVar missense mutation items, thanks to a python script written in our laboratory, we have found 959 benign and 875 pathogenic X/Gly substitutions. It must be noted first, that all the eight possible X/Gly replacements arising from single-nucleotide codon changes are present, but to a very different extent, see Fig. 10.

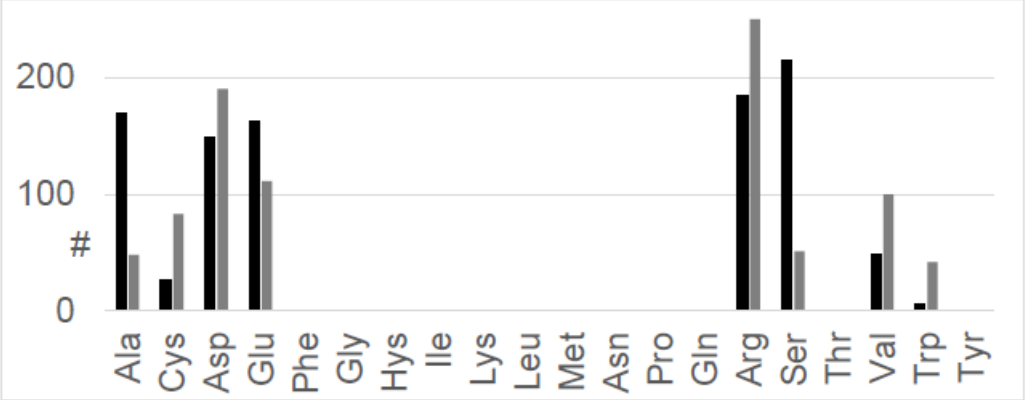

Figure 10. Distribution of X/Gly replacements referenced by ClinVar. Black and gray histograms respectively refer to Gly-substitution in benign and pathogenic missense mutations.

Gly-pipe[61], a software that we have recently developed to find those Gly mutations occurring at the protein surface and, hence, causing a change in its shape [61]. This software compares native and Gly-mutated structures to find new pockets close to the replaced amino acids. Thus, *Gly-pipe*, once applied to our data set of 255 X/Gly elements, indicates that only 96 mutations are located on protein surfaces, making them potentially responsible for functional modulations due to solvent dynamics changes. As *Gly-pipe* is a machine learning procedure to evaluate also the druggability score of new or enlarged pockets determined by X/Gly replacements, we restricted our analysis to those protein surface changes that are predicted as druggable

ones, *i.e.* with DS > 0.5. Thus, only 20 proteins of our data set have surface pockets induced by X/Gly substitutions with druggability scores suitable for ligand binding. Table 1 summarizes our results, indicating the 20 Mendelian disorders that are associated with Gly-mutations and exhibit a druggability score higher than 0.5.

Table 1

Mendelian disorders related to Gly-mutations yielding druggable surface pockets

Mendelian Disorder	OMIM Code	Protein PDB ID	X_nGly	DS[a]
Permanent neonatal diabetes mellitus 3	#618857	6C3O	Val86Gly	0.670
Ataxia-oculomotor apraxia type 1	#208920	6CVQ	Val263Gly	0.745
Early infantile epileptic encephalopathy 2	#300672	4BGQ	Trp195Gly	0.703
Deafness, autosomal dominant 9	#601369	1JBI	Val66Gly	0.674
Mirror movements 1	#157600	4PLO	Val793Gly	0.544
Congenital myasthenic syndrome 13	#614750	6FWZ	Val264Gly	0.729
Waardenburg syndrome, type 4A	#277580	6IGK	Ala183Gly	0.518
Craniofrontonasal syndrome	#304110	6P7S	Trp37Gly	0.513
Deficiency of UDPglucose-hexose-1-phosphate uridylyltransferase	#230400	6GQD	Trp154Gly	0.766
Alkaptonuria	#203500	1EYB	Val300Gly	0.519
Charcot-Marie-Tooth disease, axonal, type 2S	#616155	4B3F	Val373Gly	0.563
Long QT syndrome 2	#613688	4HQA	Glu58Gly	0.719
Gastrointestinal stromal tumor	#606764	2VIF	Val569Gly	0.519
Multiple endocrine neoplasia, type 1	#131100	6O5I	Glu45Gly	0.666
Lynch syndrome I	#120435	3THX	Val163Gly	0.622
Neurodevelopmental disorder with behavioral abnormalities, absent speech, and hypotonia	#618718	3ZYG	Trp107Gly	0.600
Amyotrophic lateral sclerosis 18	#614808	4X1L	Cys71Gly	0.750
Noonan syndrome 4	#610733	3KSY	Arg552Gly	0.560
Spastic paraplegia 4, autosomal dominant	#182601	5Z6R	Asp444Gly	0.724
Pheochromocytoma	#171300	4AJY	Trp117Gly Arg107Gly Ser111Gly	0.758 0.674 0.674

a) Druggability scores were determined with Gly-pipe (9).

3.4Multi-omics applied to cancer therapy

In general, cancer disrupts cellular relations and results in the dysfunction of vital genes. This disturbance is affective in the cell cycle and enhances abnormal proliferation [2]. There are three main types of cancer genes that control cell growth and can cause cancer to develop:

1. Oncogenes. These, when mutated, actively promote cell proliferation. They are formed when proto-oncogenes that promote cell division are improperly activated, so they are not known to be inherited. They may lead to increased/dysregulated expression of the gene in a new location or to production of fusion proteins with new functions [2]. Two common oncogenes are HER2 and RAS.

2. Gatekeeper genes. These are protective genes, also known as tumor suppressor genes. Normally, they negatively control cell growth by monitoring and controlling the cell phases or repairing mismatched DNA. Autosomal recessive mutations in tumor suppressor gene cause loss of function effect at the cellular level, inducing cells to grow uncontrollably, which may eventually form a tumor. Examples of tumor-suppressor genes include BRCA1, BRCA2, and p53 or TP53. Germline mutations in BRCA1 or BRCA2 genes increase a woman's risk of developing hereditary breast or ovarian cancers and a man's risk of developing hereditary prostate or breast cancers. They also increase the risk of pancreatic cancer and melanoma in women and men [2]. The most mutated gene in people with cancer is p53 or TP53. More than 50% of cancers

involve a missing or damaged p53 gene. Most p53 gene mutations are acquired. Germline p53 mutations are rare, but patients who carry them are at a higher risk of developing many different types of cancer.

4 Carekeeper genes. These fix the mistakes made when DNA is copied. Many of them function as tumor suppressor genes. BRCA1, BRCA2, and p53 are all DNA repair genes. If a person has an error in a DNA repair gene, mistakes remain uncorrected. Int. J. Mol. Sci. 2021, 22, 5751 4 of 21 Then, the mistakes become mutations. These mutations may eventually lead to cancer, particularly mutations in tumor suppressor genes or oncogenes. Mutations in DNA repair genes may be inherited or acquired. Lynch syndrome is an example of the inherited kind. BRCA1, BRCA2, and p53 mutations and their associated syndromes are also inherited [2].

As said before genetic changes that promote cancer can be inherited from our parents if the changes are present in germ cells, which are the reproductive cells of the body (eggs and sperm). Such changes, called germline changes, are found in every cell of the offspring. Cancer-causing genetic changes can also be acquired during one’s lifetime and are called somatic (or acquired) changes [2].

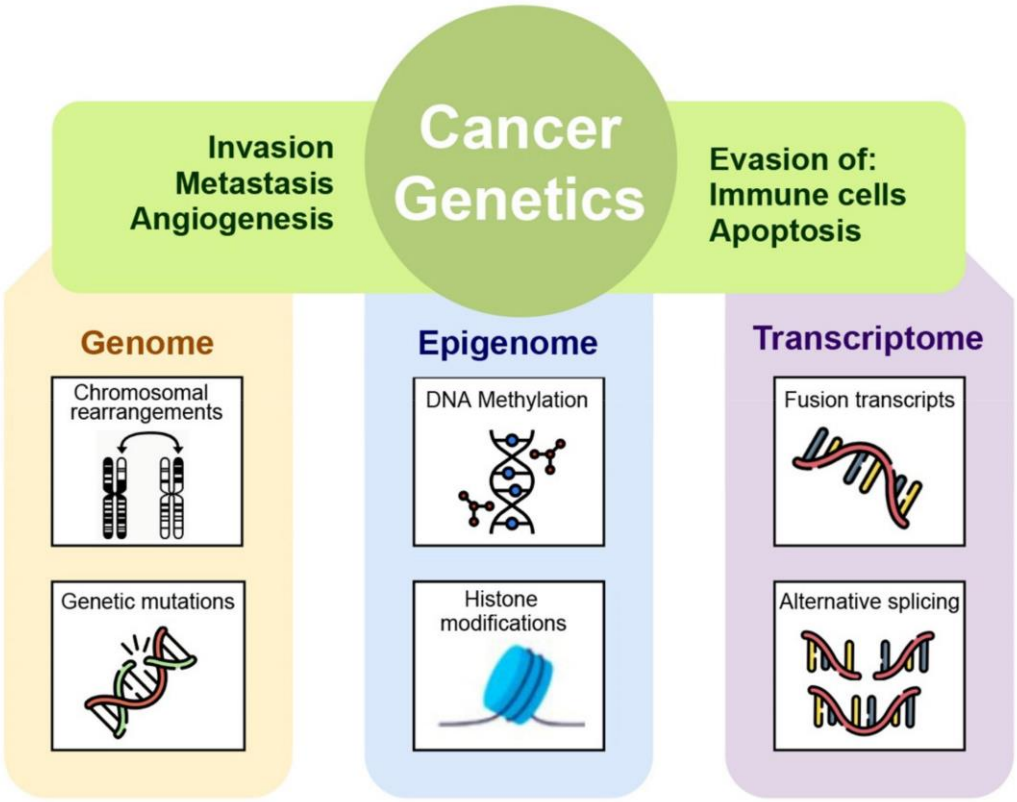

Figure 11. The many levels of interactions found in a cancer system, that can be measured via the different omics technologies, such as genomics, epigenomics, transcriptomic, and proteomic

It is largely proved that genomic instability is a reductive model; studies demonstrated

epigenetic errors resulting in aberrant gene silencing/activation [2]. According to the

definition, epigenetics is a dynamic situation in the study of cell fate, that alter the structureof DNA without directly affecting and mutating its sequence. In fact, mutationsoccurred in the elements that regulate the expression or repression of the genome, suchas transcription factors and noncoding RNAs, with a consistent effect on the coordination of multiple biological processes. These elements can be divided into three roles: “writers”and “erasers” refer to enzymes that transfer or remove chemical groups to or from DNAor histones, respectively; “readers” are proteins that can recognize the modified DNA orhistones [2].In tumor tissues, different tumor cells show various patterns of histone modification,genome-wide or in individual genes,

demonstrating that epigenetic heterogeneity existsat a cellular level and suggesting that tumorigenesis is the consequence of the combinedaction of multiple epigenetic events [2]. For example, the repression of gatekeeper genesis usually caused by DNA modification in the methylation of CpG islands together withhypoacetylated and hypermethylated histones [2]. Gene silencing experiments identifiedseveral hallmarks of epigenetic events, including histone H3 and H4 hypoacetylation,histone H3K9 methylation, and cytosine methylation [2]. Major epigenetic modificationsare classified as DNA modifications, histone modifications, effects of non-coding RNA(Figure 2).

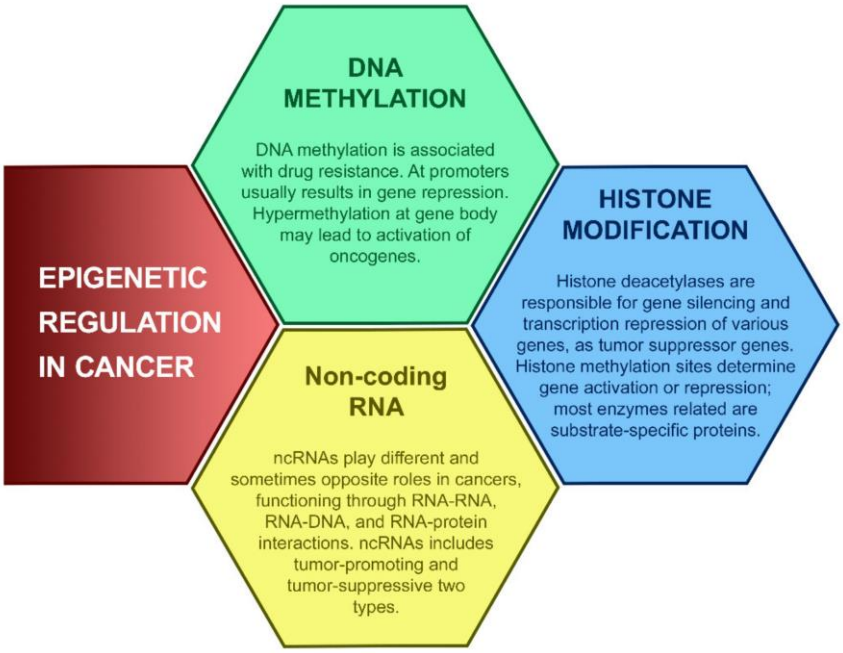

Figure 12. Epigenetic regulations in cancer. Alterations in epigenetic modifications in cancer regulate various cellular responses, including cell proliferation, apoptosis, invasion, and senescence. Through DNA methylation, histone modification, and noncoding RNA regulation, epigenetics play an important role in tumorigenesis. These main aspects of epigenetics present reversible effects on gene silencing and activation via epigenetic enzymes and related proteins.

Computational approach plays central roles not only in the analysis of high-throughput experiments, but also in data acquisition, in processing of raw file derived from several instruments, in storage and management of large streams of omics information and in the data model integration. Bioinformatics workflow management systems can be used in developing and in application of a certain pipeline. Examples of such systems include Galaxy [2], Snakemake [2], Nextflow [2], and the general-purpose Common Workflow Language [2]. Several tools for omics data studies are available in Bioconductor project as packages for the R language [2] and in Biopython project [2] [37].

All the omics technologies have a specific role to figure out the complex phenotype of cells especially in complex diseases like cancer. Knowledge of the biological molecular basis of different cellular signaling pathways does not involve only genes and transcripts, in fact, proteins and metabolites are particularly important to predict the phenotypic alterations for diagnosis and prognosis of cancer, and for this reason, in this chapter, we will spend some words about them. Table 1 represents a summary of the applications of different NGS-based and mass spectrometry-based techniques which are at the basis of different omics data acquisition approaches.

To date, genomics approach has highly sustained the finding and investigation of variations at both the germline and somatic levels thanks to many progresses in genome-exome sequencing techniques, for instance from the Sanger sequencing-based approaches to the NGS-based sequencing. Bioinformatics has always had a central role in the analysis of downstream genetic data. For example, in the multiscale scale project “The Cancer Genome Atlas” (TCGA), researchers used NGS sequencing associated to bioinformatics tools with the aim to discover somatic mutational landscape across thousands of tumor samples and to understand the complexity underlying different cancer types [56,57]. For the analysis of NGS data a sequence aligner tool is used on the sequence data (stored in FASTQ format). Some

popular aligners are the stand-alone BWA , Bowtie ,Bowtie2 , and SNAP [2] , with aligned sequences being stored in SAM (Sequence Alignment Map, text-based) or BAM (Binary Alignment Map) files.

Table 2. Summary of the applications of different techniques for sequencing, which are at the basis of different omics data acquisition approaches. Genomics, epigenomics, and transcriptomics are based on NGS techniques, whereas proteomics and metabolomics are driven by mass-spectrometric (LC-MS/MS) method. The main goal of genomics, epigenomics, and transcriptomics is the screening of genome-wide mutations, the identification of altered epigenomic modifications, and exploring differential RNA expression, while for proteomics and metabolomics is the identification of differentially regulated proteins and metabolites (reprinted from reference [6]).

OMICS	TYPE	PRINCIPLE	APPLICATION	BIOINFORMATICS TOOLS
GENOMICS	Whole exome sequencing	NGS	Exome-wide mutational/analysis	BWA Bowtie Bowtie2 SNAP SAM BAM
	Whole genome sequencing	NGS	Genome-wide mutational/analysis	
	Targeted gene/exome sequencing	Sanger sequencing	Mutational analysis in targeted gene/exon	
EPIGENOMICS	Methylomics	Whole genome bisulfite sequencing	Genome-wide mapping of DNA methylation pattern	Methylation-Array-Analysis SICER2 PeakRanger GEM MUSIC PePr DFilter MACS
	ChIP-sequencing	NGS	Genome-wide mapping of epigenetic marks	
TRANSCRIPTOMICS	RNA-sequencing	NGS	Genome-wide differential gene expression analysis	Bowtie STAR kallisto Salmon
	Microarray	Hybridization	Differential gene expression analysis	
PROTEOMICS	Deep-proteomics	Mass-spectrometry	Differential protein expression analysis	MaxQuant Perseus
METABOLOMICS	Deep-metabolomics	Mass-spectrometry	Differential metabolite expression analysis	Metab metaRbolomics Lipidr

The detection and quantification of RNA transcripts (mRNA, noncoding RNA and microRNAs) is possible owing to the employment of several transcriptomics techniques. Differently from the static nature of genome, transcriptome dynamically changes as consequence of temporal cellular and extracellular stimuli. Microarray was the technique of choice to detect alterations in cellular mRNA levels in a high-throughput manner owing to its ability to quantify the relative abundance of mRNAs for thousands of genes at the same time. Microarrays are widely used to facilitate the identification of genes with differential expression between normal and cancer conditions. With the advent of NGS, the identification of the presence and the abundance of RNA transcripts in genome-wide manner became possible. In contrast to microarrays technique, RNA-seq does not depend on the transcript-specific probes and thus can effectively perform an unbiased detection of novel transcripts, also the less abundant, with high specificity and sensitivity. Starting points for RNA-seq bioinformatics analysis include alignment-based methods, such as Bowtie [2], and STAR [2], or alignment-free methods, such as kallisto [2] and Salmon [2]. Cancer-related omics experiments often rely on specific, tailor-made analytic pipeline. TCGA and other repositories give the great opportunity to analyze the omics data by a pan-cancer approach where different types of cancers can be compared in terms of genomic and transcriptomic landscapes [2].

Given the high complexity and dynamic range of proteins, their identification and quantification in large scale are significantly challenging. Proteomic analyses are applied to identify and quantify the set of proteins present within a biological system of interest. Progressions of the tandem mass-spectrometry (LC-MS/MS) techniques in terms of resolution, accuracy, quantitation, and data analysis have made it a solid instrument for both the identification and quantification of cells proteome. Recently, the advent of cutting edge high-resolution “Orbitrap” mass-spectrometer instruments associated with powerful computational tools (i.e., MaxQuant [2] and Perseus [2]) simplified the genomewide detection of all expressed proteins in human cells and

tissues paving the way for a first draft of the human proteome [2]. MS-based proteomics techniques have been extensively applied also to investigate the proteome alteration in several human cancer tissues [2]. In particular, the study of cancer proteomes is a promising path for biomarkers and therapeutic targets identification because proteins are the molecular unit from which cellular structure and function arise [2]. The application of MS techniques is not restricted to proteomics but rather can be extended to smaller molecules such as metabolites. Metabolomics is characterized by the quantifications of metabolites that are synthesized as products of cellular metabolic activities, such as amino acids, fatty acids, carbohydrates, and lipids. Their levels can be dynamically altered in disease states reflecting aberrant metabolic functions in complex disorders like cancer. Indeed, metabolic variations are significant contributors to cancer development . This is the reason why cancer metabolomics has become an important research topic in oncology , with the aim to get new insights on cancer progression and potential therapeutic targets. Lipidomics is a subset of metabolomics [84], specifically cancer lipidomics has recently led to the identification of novel biomarkers in cancer progression and diagnosis [2]. Metabolomics is still an ongoing field with the potential to be highly effective in the discovery of biomarkers, especially in cancer. This is possible due to the support of bioinformatics tools like metab package [86], which provides an analysis pipeline for metabolomics derived from gas chromatography-MS data, or metaRbolomics package [2], which is a general toolbox that goes from data processing to functional analysis. Similarly, the lipidr package [2] is an analogous framework focused on lipidomics data processing.

In recent years, machine learning has been proved to be capable of solving many biomedical problems. These mathematical models can represent the relationships between observed variables and provide a useful description of biological phenomena. A ML tool can perform several tasks, including classification task in which the input data are divided into two or more classes and the learning system

produces a model capable of assigning one class among those available to each input. These models have important biomedical applications [2], because they are capable of discriminating between health and disease, or between different diseases outcomes [2]. In a regression task instead, the output belongs to a continuous rather than discrete domain. These models provide insights into the molecular mechanisms driving physiological states, reveal interactions between different omics, and have been used in prognostic tools [2]. In this context, due to the large amounts of heterogeneous data, the removal of non-informative characteristics whichsimplifies the model, increases its performance, and makes it less expensive to measure, reveals to be a crucial process [2]. Feature selection algorithm is a process which selects the variables that contribute most to the prediction, removing the irrelevant or less important features that can negatively contribute to the performance of the model. Both classification and regression ML techniques combined with feature selection algorithms have been widely used for cancer prognosis and prediction [2]. Moreover, many packages, which combine exploratory, supervised, and unsupervised tools, have been recently implemented in oncology. Table 4 provides a list of some of these new tools.

Table 3. Main packages tools implemented in oncology for machine learning

Package Tools	Description
mixOmics	R package for the multivariate analysis of biological datasets with a specific focus on data exploration, dimension reduction, and visualization [116].
DIABLO	Package for the identification of multi-omic biomarker panels capable of discriminating between multiple phenotypic groups. It can be used to understand the molecular mechanisms that guide a disease [117].
MOFA	Package for discovering the principal sources of variation in multi-omics data sets [118].
Biosigner	Package for the identification of molecular signatures from large omics datasets in the process of developing new diagnostics [119].
omicRexposome	Package that uses high-dimensional exposome data in disease association studies, including its integration with a variety of high-performance data types [120].
OmicsLonDA	Package that identifies the time intervals in which omics functions are significantly different between groups [121].
Micrographite	Package that provides a method to integrate micro-RNA and mRNA data through their association to canonical pathways [122].
pwOmics	Package for integrating multi-omics data, adapted for the study of time series analyses [123].

The interpretation of a ML model results could be a difficult task. A strategy that can provide readily interpretable results consist in mapping omic data on functional characteristics, in order to make them more informative and to associate them with a wider body of biomedical knowledge. Some functional enrichment approaches are listed below:

• Over-Representation Analysis (ORA) [2];

• Gene-Set Enrichment Analysis (GSEA) [2];

• Multi-Omics Gene-Set Analysis (MOGSA) [2];

• Massive Integrative Gene Set Analysis (MIGSA) [2];

• Exploratory Data Analysis (PCA) [2];

• Divergence Analysis [2].

3.5 MMP-14 Cyclic-peptides inhibitors

Matrix Metallo Proteinase 14 (MMP 14) [84] is known to be involved in many types of bone cancers [84] . After many researches we have found the protein in complex with Collagen (PDB:2MQS) [84].

Figure 13.MMP14 in complex with Collagen.PDB: 2MQS

Thanks to a collaboration with the University of Essex, that obtained an experimental crystal structure of MMP14 in complex with a cyclic peptide inhibitor, we have conducted computational studies on this target. We have performed Classical Molecular Dynamics simulations and meta-dynamics simulations , such as Supervised Molecular Dynamics simulations. The results werenot so good and celar , in fact after having performed SuMD simulation of the experimental crystal , we have observed that the experimental cyclic peptide , prevently moved about 20 Angstroms away from the binding pocket , did not reach the right pose as in the

experimental crystal of MMP14 in complex with the cyclic peptide itself. This problem is known since, as we have already said, small molecules or peptides and cyclic peptides can have different distortions and a consistent probabilistic variability , some of these are the noise and the large number of degrees of freedom.Anyway we are studying different algorithms, in collaboration with many universities and scientists , that could avoid all these problems and distortions.

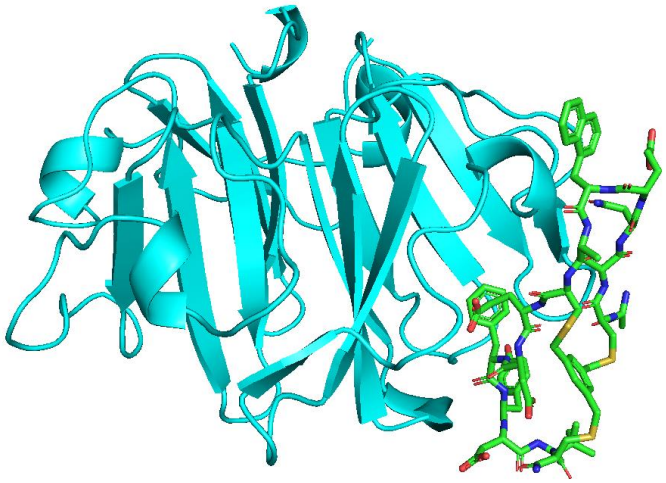

Figure 14. Experimental crystal of MMP14-cyclic peptide inhibitor obtained in the crystallography laboratory of the University of Essex.

3.6Conclusions

Regarding the study on potential drugs against SARS-CoV-2 Spike Protein we can say that SARS-CoV-2 invades human cells via ACE2, a transmembrane protein expressed on the surface of alveolar cells of the lungs. Upon binding of ACE2, viral and host cell membranes fuse and the virus enters into the host cell. This results in the development of an infectious disease, called COVID-19, which is associated with a major immune infammatory response. Deaths are caused by respiratory failure, which have been linked to a cytokine storm with high serum levels of pro-infammatory cytokines and chemokines[62].

The aim of this proof of principle study was to propose a robust in silico protocol that overcame limitations of classic virtual screening studies. The role of hydration patterns in target recognition and binding is completely absent in docking simulations. Furthermore, in most virtual screenings, while the ligand is fexible, proteins are only semi-fexible, which afects both the resulting pose of the ligand and the scoring system[63].

More reliable information can only be obtained by MD simulations, which, despite being computationally expensive, allow to take into account macromolecules' unique features, such as conformational fexibility, charge distribution, and hydration patterns in target recognition, drug binding, and drug unbinding[64].

Our results show the importance of taking into account the full structural features of a protein–ligand complex and how a combination of MD simulations may help predict the validity of a proposed inhibitor. Our work suggests that Simeprevir and Lumacafor could be potential initial compounds able to prevent and treat SARS CoV-2 infection.

Regarding the study on flavonoids against h-ERG receptors , we can say that the widespread use of plant-derived medications, functional foods, and dietary supplements results in a remarkable daily intake of flavonoids. The recent observation that the combination of quercetin and dasatinib decreases the number of

senescent cells in humans [65] is expected to generate great appeal of flavonoid use among the general public. Beyond their well-known beneficial effects on overall cardiovascular mortality, accumulating in vitro and in vivo evidence point out to some flavonoids as potential hERG channel blockers . Noticeably, unlike prescription-drugs, phyto-therapeutic products and dietary supplements are mostly used without prescription by a healthcare professional. Finally, a further concern is represented by the concomitant intake of flavonoids and QT prolonging drugs, due to the potential impact of these natural compounds on drug pharmacokinetics and bioavailability. Though the topics covered here deserve further investigations to better clarify the clinical relevance of these interactions, they have important implications that impose cautiousness in phytotherapy indication and dietary recommendation of flavonoids, especially for patients susceptible to LQTS and Torsades de Pointes owing to a reduced repolarization reserve.

Regarding the study on Mendelian Disorders , we can say that Gly-mutations can be responsible for the formation of new pockets on the protein surface if some conditions are fulfilled, as in the case that amino acids involved in the replacement bear bulky side chains and are located near to the protein surface [66].
This feature can be exploited to design Glymutants for proteins whose activity we like to control through suitable ligands. In the present report, we explored how evolution designed Gly-mutants as witnessed in the human genome by ClinVar. ClinVar, as of October 3, 2020, classified 959 benign and 875 pathological missense Gly-mutations,offering a large repertoire for understanding the way these mutations can causeMendelian disorders.

By crossing ClinVar and PDB data banks, the structural feature that originates a Mendelian disorder can be analyzed and, in case the Gly-mutation determines the formation of a potential new binding site, remediation could be found by discovering

ligands that shield the latter binding site. Thus, the possibility that a pill can fix a Mendelian disorder seems to be practicable.

As a final remark, it must be underlined that a severe bottleneck for the present investigation has been due to the lack of structural information, reducing the 875 proteins with pathological Glymutations to 225 structurally characterized systems. The continuous progress in protein structure predictions, like the ones recently proposed (67-68), will solve in part the problem. In general, AI will yield more and more powerful tools to predict which and how Mendelian disorders can be cured. The possibility of very fast tailoring of ligands for Gly-induced pockets through molecular graph generation with graph neural networks is just a recent example of AI-related advancements [69].

Regarding the sudies and reviews about possible applications of Multi-omics in Cancer therapy we can say that these computational approaches play a central role in improving our current cancer diagnostic capabilities . The understanding of the cancer progression, the new therapeutic interventions, and the discovery of novel cancer biomarkers need to adopt and integrate different omics strategies at multiple levels. To achieve this aim, there are five essential challenges in the omics integration workflow: (1) experimental challenges, (2) individual omics datasets, (3) integration issues, (4) data issues, and (5) biological knowledge.

1. Experimental challenges: an accurate sample preparation in a multi-omics perspective becomes one of the major experimental challenges, with the aim to achieve a universal sample collection and preparation protocol for generating multiple omics datasets.

2. Individual omics datasets: data preprocessing is also another significant challenge. This process can be performed on each omic dataset independently before merging significant results or after the production of a unique merged dataset. Moreover, the information included in each individual omic dataset requires very different standardization and scaling approaches, operating in different numerical and time scales
3. Integration issues: data integration issues increases the difficulty of accounting for false positives in merged datasets. Additional problems include the management of rigorous approaches based on statistical models with respect to less rigorous approaches that include a biological interpretation. In comparison to a single omics study, a multi-omics approach has the benefit to allow a deeper understanding ofhow the tumoral transformation is affecting the flow of information from different omics levels resulting in a bridge between cancerous genotype and the phenotype.
4. Data issues: the storage of omics data is very important for reproducibility. To this end, new omic platforms are being developed to provide essential clinical data for insights into the prognosis and diagnosis of diseases.
5. Biological knowledge: the interpretation of the outputs of computational models requires a deep knowledge of the biological system under study, in order to discriminate results that are not biologically relevant.

Despite these challenges the application of bioinformatics data integration and analysis, as well as the use of molecular modeling algorithms, allow to formulate many predictions of drug–target interactions to greatly facilitate guided drug development and guided drug resistance prevention [2].

Artificial intelligence (AI) approaches act on many aspects related to cancer therapy, including drug discovery and development and how these drugs are clinically validated and ultimately administered to patients [2]. The convergence of AI and

cancer therapy has led to multiple benefits in terms of cost and time reduction. AI methods, ranging from regression models to neural networks can accelerate drug discovery, harness biomarkers to accurately match patients to clinical trials, and truly customize cancer therapy using only patients' own data. In conclusion, the design and development of methods that integrate different multi-omic computational approaches in order to create robust and reliable models can lead to enormous advances in understanding the biology of cancer. As bioinformatics tools evolve, they must become user-friendly, interconnected, interoperable, and powerful for intensive analyses. In this context, integrated omics is not just an ensemble of computational tools, but a cohesive paradigm for deeper biological interpretation of multi-omics datasets that will potentially reveal novel details into cancer investigation. Although this field is still under development, many advances are constantly being made, with the development of new updated algorithmic approaches.

The results in all studies have been satisfactory and in some case are in progress. For a more accurate understanding, I refer you to our scientific publications which you can find in the references section.

In conclusion, it must be emphasized that, even if all these computational studies are promising for the futureperspectives, and are of great help for the scientific community, as they reduce time and labor costs, computational simulations actually, can't never replace what happens naturally in the organism. Therefore, in order to provide a detailed study to the scientific community, it is necessary, in my opinion, that in silico simulations should always be confirmed by experimental studies, such as *"in vitro"* and "in *vivo"* tests.

In any case, it is always better to check, before conducting a computational study, if there exsists an experimental confirmation of the target we are studying , for example in the databases we use or in scientific publications , in order to have a very important reference from which to start our computational study.

4. APPENDIX

In the appendix section I want to report two useful Python scripts that were used in the two recently published projects. The first Python script is the one that was used for the X / Gly mutation project with Professor Neri Niccolai. Instead the second script is the one that was used for the simulation of the Supervised Molecular Dynamics of the interaction of the two proteins involved in the pathogenicity mechanism of the SARS-Cov-2 virus, namely the RBD Spike Protein and the ACE 2 (Angiotensin Converting Enzyme 2) . I report below the two entire scripts.

1. X/Gly Script (WebScraping research of X/Gly mutations , and related deseases and protein structures)

```
#!/usr/bin/python

import urllib2

import re

import httplib

from urllib2 import urlopen
```

```
import shutil
import time
import subprocess
import os
import time
import argparse
import selenium
import openpyxl
import csv
from socket import error as SocketError
import errno

#Creazione lista ClinVar

with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/clinvar_result.txt','r') as clin :
    lista_clin = clin.readlines()

# Ricerca soltanto delle mutazioni in glicina
lista_mutazioni_gly = []
for i in lista_clin:
```

```
        try:
          if i.split()[1][-4:-1] == 'Gly':
             lista_mutazioni_gly.append(i)
        except IndexError:
          pass


lista_mutazioni_gly_senza_doppioni = []
for gen in lista_mutazioni_gly :
   if gen not in lista_mutazioni_gly_senza_doppioni:
       lista_mutazioni_gly_senza_doppioni.append(gen)


# Eliminiamo la ridondanza e scriviamo il file Mutations_without_redundance
file_mutazioni_gly_redundance = open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_linee_geni.csv', 'w')
for rigo in (lista_mutazioni_gly_senza_doppioni) :
    file_mutazioni_gly_redundance.write(str(rigo)+'\n')
file_mutazioni_gly_redundance.close()


with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_TOTALI.csv', 'w') as file_mutazioni_tot:
       for m in lista_mutazioni_gly_senza_doppioni :
```

```
        file_mutazioni_tot.write(m.split()[0]+'\t'+m.split()[1]+'\n')



with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_TOTALI.csv', 'r') as file_mutazioni_tot:
    lista_file_mut_tot =  file_mutazioni_tot.readlines()
    lista_file_mut_tot = list(set(lista_file_mut_tot))
    for h in lista_file_mut_tot:
      for u in  lista_file_mut_tot:
        if h.split('(')[0] != u.split('(')[0] and h.split('(')[1] == u.split('(')[1]:
          lista_file_mut_tot.remove(h)


with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_TOTALI.csv', 'w') as file_mutazioni_tot_no_redund:
    file_mutazioni_tot_no_redund.writelines(lista_file_mut_tot)


with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_linee_geni.csv','r') as file_mut_pdb :
    with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutations_without_redundance.csv','a') as final_file :
```

```
try:
  lista_file_mut_pdb = []
  lista_doc_unico_gene = []
  lista_file_mut_pdb_ini = file_mut_pdb.readlines()
  for i in lista_file_mut_pdb_ini:
     if i == '\n':
          lista_file_mut_pdb_ini.remove(i)
  #lista_file_mut = file_mut.readlines()

  for i in lista_file_mut_pdb_ini :
    lista_file_mut_pdb.append(i.split('(')[1].split(')')[0])
  lista_file_mut_pdb = list(set(lista_file_mut_pdb)) #elimino ridondanza dei geni al fine di diminuire il tempo di calcolo
  lista_file_mut_pdb.sort() # ordino geni in ordine alfabetico
  # creazione lista con tutti i geni unitari
  for i in lista_file_mut_pdb :
    if i[0] == 'g':
        lista_file_mut_pdb.remove(i)


  for l in lista_file_mut_pdb :
```

```
            for j in  lista_file_mut_pdb_ini:
                if l == j.split('(')[1].split(')')[0]:
                    lista_doc_unico_gene.append(j)
                    break
        final_file.writelines(lista_doc_unico_gene)

    except (IndexError , NameError , urllib2.HTTPError ,urllib2.URLError , httplib.BadStatusLine ) :
        pass

time.sleep(1)

#Ricerca totale di tutti i possibili codici pdb su uniprot e crazione di un database di mutazioni per ogni gene --
with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutations_without_redundance.csv', 'r') as file_mutazioni_without_redundance:
  with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutations_&_pdb_without_redundance.csv', 'a') as mutations_pdb:

        lista_mutazioni_gly_no_redundance = file_mutazioni_without_redundance.readlines()
```

```
for i in lista_mutazioni_gly_no_redundance :
    i = i[0:-1]
    try:
      ref_seq_code = i.split('(')[1].split(')')[0] # creo variabile ref_seq_code
    #print  ref_seq_code
    except IndexError:
      pass
    time.sleep(1)
    try:
      url= "https://www.uniprot.org/uniprot/?query=" +ref_seq_code +('&sort=score')  #ref_seq_code without NM only code number
      time.sleep(1)
    #print ref_seq_code,url
    #download uniprot search - parsing to take uniprot accession number

      response=urllib2.urlopen(url)
      time.sleep(1)
      webcontent=response.read()
    except (urllib2.HTTPError , urllib2.URLError , httplib.BadStatusLine , SocketError):
      pass

    out_wc=open("/home/libra/out_wc","a")
```

```
out_wc.write(webcontent)


f1 = open("/home/libra/out_wc" ,'r')
lines1 = f1.readlines() #legge tutte le linee e le mette in una lista


for lin1 in lines1:


    ucode=re.findall('<a href="/uniprot/\w+',lin1)
    tl=len(ucode)
    if tl > 0:
        #print ucode


        unacc=ucode



#print unacc
#print acc


out_wc.close()
f1.close()


os.system("rm /home/libra/out_wc")
```

```
info_uni= "PDB;"

lista_pdb = []

for elemento in unacc :
  acc = elemento.split('/')[2]
  try:
    url1="https://www.uniprot.org/uniprot/"+acc+".txt" #download uniprot search - parsing to take uniprot accession number
    time.sleep(1)
    response1=urllib2.urlopen(url1)

  except (urllib2.HTTPError , urllib2.URLError , httplib.BadStatusLine , SocketError):
    pass

  webcontent1=response1.read()

  out_wc1=open("/home/libra/out_wc1","a")
  out_wc1.write(webcontent1)

  f2 = open("/home/libra/out_wc1" ,'r')
```

```
lines2 = f2.readlines() #legge tutte le linee e le mette in una lista

mem=[]

try:
  for lin2 in lines2:

        if ".join(lin2).split()[0] == "DR" and ".join(lin2).split()[1] == info_uni:

            k=info_uni+"_"+".join(lin2).split()[2]
            mem.append(k)

except IndexError:
   pass

#print mem
```

```
    out_wc1.close()
    f2.close()

    os.system("rm /home/libra/out_wc1")
    time.sleep(2)

    if len(mem) > 0:
       for pdb in mem:
            lista_pdb.append(pdb)

    lista_pdb = list(set(lista_pdb))



if  lista_pdb > 0 :
     mutations_pdb.write(str(i)+'\t'+'PDB : '+str(lista_pdb)+'\n\n')
else:
     mutations_pdb.write(str(i)+'\t'+'\n\n')
```

```
time.sleep(1)

# Ricerchiamo il codice pdb referente al gene - la risoluzione - la data di deposizione della struttura - la lunghezza della sequenza

lettura = open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutations_&_pdb_without_redundance.csv','r')
lista_lettura = lettura.readlines()
lettura.close()

with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_e_pdb_uniprot_corrisposti_gene.csv','a') as file_mutazioni :

   for i in lista_lettura:
```

```
    if i == '\n':
        lista_lettura.remove(i)


for i in  lista_lettura:
    lista_pdb = []
    i = i[0:-1]
    print i
    ref_seq_code = i.split('(')[1].split(')')[0]


    linea = i.split('\tPDB')[1]
    lista_linea = linea.split()
    print (lista_linea)



    for elemento in lista_linea:
        try:
            elemento = elemento.split('_')[1].split(';')[0]
            lista_pdb.append(elemento)
        except IndexError:
            pass


    lista_finale_pdb = []
```

```
    if len (lista_pdb) > 0:
      for pdb in lista_pdb:
        try:
          #pdb = pdb.split('_')[1][0:-1]
          url3 = 'https://files.rcsb.org/view/'+pdb+'.pdb'
          time.sleep(1)
          try:
             urlopen(url3)

          except:
             pass

          try:
            response3 = urllib2.urlopen(url3)

          except (urllib2.HTTPError , urllib2.URLError, httplib.BadStatusLine , SocketError):
            pass

          webcontent2 = response3.read()
          out_wc3=open("/home/libra/DANIELE/out_wc3.txt","w")
          out_wc3.writelines(webcontent2)
```

```
out_wc3.close()
with open('/home/libra/DANIELE/out_wc3.txt','r') as testo:
  try:
    for linea in testo:
      linea = ''.join(linea).split()
      for p in linea:
        if  p == ref_seq_code  or p == ref_seq_code+'_HUMAN' or p == ref_seq_code+'_RAT' or p == ref_seq_code+'_MOUSE' or p == ref_seq_code+';'  or p == ref_seq_code+'_HUMAN;' or p == ref_seq_code+'_RAT;' or p == ref_seq_code+'_MOUSE;' or p == ref_seq_code+',' or p == ref_seq_code+'_HUMAN,' or p == ref_seq_code+'_RAT,' or p == ref_seq_code+'_MOUSE,':
          #print 'yes'
          # Scrivere codice per greppare : Data deposizione pdb , Risoluzione pdb e Lunghezza sequenza pdb
          lista_finale_pdb.append(pdb)
          break
  except IndexError:
    pass
except  (urllib2.HTTPError ,  urllib2.URLError, httplib.BadStatusLine):
```

```
        pass



lista_finale_pdb_senza_copie  = []
for pdb in lista_finale_pdb :
        if pdb not in lista_finale_pdb_senza_copie:
                lista_finale_pdb_senza_copie.append(pdb)
print i


if len (lista_finale_pdb_senza_copie) > 0 :
        file_mutazioni.write(str(i.split('\tPDB')[0])+'\t'+'PDB :'+'  ')
        try:
          for single in lista_finale_pdb_senza_copie:
              url4 = 'https://files.rcsb.org/view/'+single+'.pdb'


              try:


                urllib2.urlopen(url4)
                time.sleep(1)
                response4 = urllib2.urlopen(url4)
                time.sleep(1)
```

```
except  (urllib2.HTTPError , urllib2.URLError ,SocketError) :
  pass


webcontent3 = response4.read()
pdb_single = open("/home/libra/DANIELE/pdb.txt","w")
pdb_single.writelines(webcontent3)
pdb_single.close()
with open('/home/libra/DANIELE/pdb.txt','r') as testo:
     lista_file_testo = testo.readlines()
     file_mutazioni.write(lista_file_testo[0]+'\t')
     for rigo in lista_file_testo:
       rigosplit = ''.join(rigo).split()


       if   rigosplit[0] == 'SEQRES'  :
                file_mutazioni.write('sequence_lenght '+rigosplit[3]+'     ')
                break




     for rigo in lista_file_testo:
```

```
                    if rigo.split()[0] == 'REMARK' and 'RESOLUTION.' in rigo.split():
                        file_mutazioni.write(rigo+'       \n')
                        #break

        except (IndexError , urllib2.HTTPError , urllib2.URLError , NameError , httplib.BadStatusLine):
            pass
            file_mutazioni.write('\t\t')
        file_mutazioni.write('\n\n')
    else :
        file_mutazioni.write(str(i.split('\tPDB')[0]) +'\n\n')


time.sleep(1)

# Ricerco infine tutte le mutazioni per ogni gene -- presenti nel file Mutazioni_TOTALI.csv -- e le faccio scrivere in un file finale -- denominato Mutations_pdb_final.csv -- avente tutte le informazioni per ogni gene

with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_TOTALI.csv', 'r') as file_mutation_read:
```

```
  with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutazioni_e_pdb_uniprot_corrisposti_gene.csv','r') as file_mutazioni_pdb_read:
      with open ('/home/libra/Scrivania/progetto_niccolai/mutazioni/prova_script_finale/Mutations_pdb_final.csv','a') as final_file_append:
          lista_file_mutation_read = file_mutation_read.readlines()
          lista_file_mutazioni_pdb_read = file_mutazioni_pdb_read.readlines()
          for b in lista_file_mutazioni_pdb_read :
            try:
              if b.split('_')[0] == "NM" or b.split('_')[0] == "NC":
                lista_nuove_mutazioni = []
                for h in lista_file_mutation_read  :
                  if b.split('(')[1].split(')')[0] == h.split('(')[1].split(')')[0] :
                    lista_nuove_mutazioni.append(h.split()[1])
                final_file_append.write('Gene  :  '+b.split('(')[1].split(')')[0]+'    Mutazioni  : '+str(lista_nuove_mutazioni)+'  Gene e Malattie : '+(b)+'\n')
              else :
                final_file_append.write(b)
            except IndexError:
              pass
```

2. SuMD RBD-ACE2

```
#!/usr/bin/python
#!/usr/local/bin/python




import shutil
import time
import subprocess
import os
import time
import argparse


input_command = ('export GMX_MAXCONSTRWARN=-1')
subprocess.Popen(input_command,shell=True).wait()
os.system('echo 1 |gmx pdb2gmx -f Complex.pdb -o Complex.gro -p topol.top -v  -water tip3p')
os.system('gmx editconf -f Complex.gro -o KALP_newbox.gro  -bt cubic -d 1.6 -c') # Con Membrana  gmx editconf -f Complex.pdb -o KALP_newbox.gro -c -box x y z
with open ('box.txt','r') as file_read:
        for linea in file_read:
```

```
        if ''.join(linea).split()[1] == 'size':
            os.system('gmx editconf -f Complex.pdb -o KALP_newbox.gro -c -box '+''.join(linea).split()[3]+' '+''.join(linea).split()[4]+' '+''.join(linea).split()[5])
time.sleep(2)
os.system('rm *#')
time.sleep(2)
```

Solvatazione

```
input_command = ('gmx solvate -cp KALP_newbox.gro -cs spc216.gro -o solv.gro -p topol.top')
subprocess.Popen(input_command,shell=True).wait()
```

Ioni

```
input_command = ('gmx grompp -f ../em.mdp -c solv.gro -p topol.top -o ions.tpr -r solv.gro -maxwarn 1000')
subprocess.Popen(input_command,shell=True).wait()
```

```
input_command = ('echo 15 |gmx genion -s ions.tpr -o solv_ions.gro -p topol.top -neutral')
subprocess.Popen(input_command,shell=True).wait()
```

```
# avvio minimizzazione
input_command = ('gmx grompp -f ../em.mdp -c solv_ions.gro -p topol.top -o em_0.tpr -r solv_ions.gro -maxwarn 1000')
subprocess.Popen(input_command,shell=True).wait()



input_command = ('gmx mdrun -v -deffnm em_0 -nt 32')
subprocess.Popen(input_command,shell=True).wait()

# Controllo minimizzazione
i = 0
while True :
    lista_em = []
    os.system('echo Potential| gmx energy -f em_'+str(i)+'.edr > em.txt')
    time.sleep(2)
    with open ('em.txt','r') as em:
        for line in em:
            lista_em.append(line)
```

```
        for l in lista_em:
          if l == '\n':
            lista_em.remove(l)
        time.sleep(2)
        if ''.join(lista_em[4]).split()[0] == 'Potential' and float(''.join(lista_em[4]).split()[1]) >= 0:
                os.system('gmx grompp -f ../em.mdp -c em_'+str(i)+'.gro -r em_'+str(i)+'.gro -p topol.top -o em_'+str(i+1)+'.tpr -maxwarn 1000')
                os.system('gmx mdrun -v -deffnm em_'+str(i+1)+'  -nt 32')
        else :
                break
        i += 1
    os.system('rm *#')


# avvio nvt
#input_command = ('gmx grompp -f nvt.mdp -c em_'+str(i)+'.gro -r em_'+str(i)+'.gro -p topol.top -o nvt_0.tpr -maxwarn 1000')
input_command = ('gmx grompp -f nvt.mdp -c em.gro -r em.gro -p topol.top -o nvt_0.tpr -maxwarn 1000')
subprocess.Popen(input_command,shell=True).wait()
input_command = ('gmx mdrun -deffnm nvt_0  -nt 32')
subprocess.Popen(input_command,shell=True).wait()
```

```
# Controllo nvt
i = 0
while True :
    lista_nvt = []
    os.system('echo Temperature| gmx energy -f nvt_'+str(i)+'.edr > nvt.txt')
    time.sleep(2)
    with open ('nvt.txt','r') as nvt:
        for line in nvt:
          lista_nvt.append(line)
        for l in lista_nvt:
          if l == '\n':
            lista_nvt.remove(l)
        time.sleep(2)
        if ''.join(lista_nvt[4]).split()[0] == 'Temperature' and float(''.join(lista_nvt[4]).split()[1]) > 310 or float(''.join(lista_nvt[4]).split()[1]) < 309 :
                os.system('gmx grompp -f nvt.mdp -c nvt_'+str(i)+'.gro -r nvt_'+str(i)+'.gro -t nvt_'+str(i)+'.cpt -p topol.top -o nvt_'+str(i+1)+'.tpr -maxwarn 1000')
                os.system('gmx mdrun -deffnm nvt_'+str(i+1)+'  -nt 32')
        else :
                break
        i += 1
```

```
    os.system('rm *#')


#avvio npt
input_command = ('gmx grompp -f npt.mdp -c nvt_'+str(i)+'.gro -r nvt_'+str(i)+'.gro -t nvt_'+str(i)+'.cpt  -p topol.top -o npt_0.tpr -maxwarn 1000')
subprocess.Popen(input_command,shell=True).wait()
input_command = ('gmx mdrun -deffnm npt_0  -nt 32')
subprocess.Popen(input_command,shell=True).wait()


# controllo npt
i = 0
while True :
    lista_npt = []
    os.system('echo Pressure| gmx energy -f npt_'+str(i)+'.edr > npt.txt')
    time.sleep(2)
    with open ('npt.txt','r') as npt:
        for line in npt:
          lista_npt.append(line)
        for l in lista_npt:
          if l == '\n':
            lista_npt.remove(l)
        time.sleep(2)
```

```
        if ".join(lista_npt[4]).split()[0] == 'Pressure' and float(".join(lista_npt[4]).split()[1]) < 0.900 or float(".join(lista_npt[4]).split()[1]) > 1.100 :
                os.system('gmx grompp -f ../npt.mdp -c npt_'+str(i)+'.gro -r npt_'+str(i)+'.gro -t npt_'+str(i)+'.cpt -p topol.top -o npt_'+str(i+1)+'.tpr -maxwarn 1000')
                os.system('gmx mdrun -deffnm npt_'+str(i+1)+'  -nt 32')
        else :
                break
        i += 1
        #os.system('rm npt_'+str(i-1)) Rimuovo tutti i file npt antecedenti
    os.system('rm *#')



input_command = ('gmx grompp -f md_1ns.mdp -c npt_'+str(i)+'.gro -r npt_'+str(i)+'.gro -t npt_'+str(i)+'.cpt -p topol.top  -o md_0.tpr -maxwarn 1000')
subprocess.Popen(input_command,shell=True).wait()
input_command = ('gmx mdrun -deffnm md_0  -nt 32')
subprocess.Popen(input_command,shell=True).wait()


# inizializzo variabile md_input
```

```
md_input = 'md_0'

#creare index corretto es  os.system("echo -e '0 & 19 \n q' |gmx make_ndx -f npt_7.gro -o index.ndx") esempio ligand and not system


os.system("echo -e '19 \n 20 \n' |gmx pairdist -f md_0.xtc -n index.ndx")


#creare var diff_iniz (diff tra ultimo e primo valore)

with open('dist.xvg','r') as dist:
    lista_dist = dist.readlines()

for i in lista_dist:
 if i.split()[0] == '0.000':
    primo_valore = float(i.split()[1])

ultimo_valore = float(lista_dist[-1].split()[1])


# Analisi della distanza ottenuta dalla prima dinamica - Avvio algoritmo SuMD

if float(ultimo_valore) >= float(primo_valore) or (float(ultimo_valore) < float(primo_valore) and float(ultimo_valore) > 0.27):
   #ciclo while True
   i = 0
   md_input = 'md_'+str(i)
   md_out = 'md_'+str(i+1)
   while True :
```

```
os.system('gmx grompp -f md_1ns.mdp -c '+md_input+'.gro -p topol.top -r '+md_input+'.gro  -o '+md_out+'.tpr -n index.ndx -maxwarn 1000')

os.system('gmx mdrun -v -deffnm  '+ md_out+' -nt 32')

os.system("echo -e '19 \n 20 \n' |gmx pairdist -f  "+md_out+".xtc -n index.ndx")

with open('dist.xvg','r') as dist:

    lista_dist = dist.readlines()

lista_distanze_tempi = []

for val in lista_dist:

  if val.split()[0] == '0.000':

      primo_valore_out =  float(val.split()[1])

  if val[0] != '#' and val[0]!= '@' and  val.split()[0] != '0.000' and len(val.split()) == 2:

      lista_distanze_tempi.append(val.split())




lista_distanze_tempi.sort()


lista_only_distance = []

for dist in lista_distanze_tempi:

  lista_only_distance.append(float(dist[1]))
```

```
        minim_value_dist = min(lista_only_distance)
        if minim_value_dist < primo_valore_out:
          for el in lista_distanze_tempi:
            if float(el[1]) == float(minim_value_dist):
              frame = el[0]
          md_input = 'md_'+str(i+1)
          input_command = ('echo 0 |gmx trjconv -f '+md_out+'.xtc -s '+md_out+'.tpr -o
'+md_input+'.pdb  -b '+frame+'  -e  '+frame)
          subprocess.Popen(input_command,shell=True).wait()
          os.system('gmx editconf -f '+md_input+'.pdb  -o  '+md_input+'.gro')

          md_out = 'md_'+str(i+2)
          os.system('rm mdout.mdp')
          i += 1

        if float(minim_value_dist) <= 0.27 :
              #scrivere col trjconv dal primo  fino al frame piu basso
              break

        else :
           md_input = 'md_'+str(i)
           md_out = 'md_'+str(i+1)
```

```
        i = i-1
    i += 1
    os.system('rm *#')

if float(ultimo_valore) < primo_valore and float(ultimo_valore) <= 0.27:
    None
os.system('gmx grompp -f md.mdp -c '+md_input+'.gro -p topol.top -r '+md_input+'.gro   -o md_cmd.tpr -maxwarn 1000')
os.system('gmx mdrun -deffnm md_cmd -v -nt 32')
```

5. REFERENCES

1. *Lesk, A. M. (26 July 2013). "Bioinformatics". Encyclopaedia Britannica*
2. Pettini F, Visibelli A, Cicaloni V, Iovinelli D, Spiga O. Multi-Omics Model Applied to CancerGenetics. *International Journal of MolecularSciences*. 2021; 22(11):5751. https://doi.org/10.3390/ijms22115751
3. *Dawson, W. K.; Maciejczyk, M.; Jankowska, E. J.; Bujnicki, J. M. (2016). "Coarse-grainedmodeling of RNA 3D structure". Methods. **103**: 138–56. doi:10.1016/j.ymeth.2016.04.026. PMID 27125734.*
4. *Kmiecik, S.; Gront, D.; Kolinski, M.; Wieteska, L.; Dawid, A. E.; Kolinski, A. (2016). "Coarse-GrainedProteinModels and Their Applications". ChemicalReviews. **116** (14): 7898–936. doi:10.1021/acs.chemrev.6b00163. PMID 27333362.*
5. Wong, K. C. (2016). *ComputationalBiology and Bioinformatics: Gene Regulation*. CRC Press/Taylor & Francis Group. ISBN 9781498724975.
6. *Joyce, A. P.; Zhang, C.; Bradley, P.; Havranek, J. J. (2015). "Structure-basedmodeling of protein: DNA specificity". Briefings in FunctionalGenomics. **14** (1): 39–49. doi:10.1093/bfgp/elu044. PMC 4366589. PMID 25414269.*
7. Spiga, E.; Degiacomi, M. T.; Dal Peraro, M. (2014). "New Strategies for Integrative DynamicModeling of Macromolecular Assembly". In Karabencheva-Christova, T. (ed.). *BiomolecularModelling and Simulations*. Advances in ProteinChemistry and StructuralBiology. **96**. Academic Press. pp. 77–111. doi:10.1016/bs.apcsb.2014.06.008. ISBN 9780128000137. PMID 25443955.
8. Ciemny, Maciej; Kurcinski, Mateusz; Kamel, Karol; Kolinski, Andrzej; Alam, Nawsad; Schueler-Furman, Ora; Kmiecik, Sebastian (4 May 2018). "Protein–peptide docking: opportunities and challenges". *DrugDiscoveryToday*. **23** (8): 1530–37. doi:10.1016/j.drudis.2018.05.006. ISSN 1359-6446. PMID 29733895.
9. *PDB : https://www.rcsb.org/*
10. *Uniprot :https://www.uniprot.org/*
11. *X-RayCristallography : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1186895/*
12. Damadian, R. V. "TumorDetection by NuclearMagneticResonance," Science, 171 (March 19, 1971): 1151-1153
13. Xiao, C., Fischer, M.G., Bolotaulo, D.M., Ulloa-Rondeau, N., Avila, G.A., and Suttle, C.A. (2017) "Cryo-EM reconstruction of the Cafeteriaroenbergensis virus capsidsuggestsnovelassemblypathway for giantviruses". *Scientific Reports*, **7**: 5484. doi:10.1038/s41598-017-05824-w.
14. *https://www.schrodinger.com/products/pymol*
15. Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., and Ferrin, T.E. "UCSF Chimera - A Visualization System for ExploratoryResearch and Analysis." *J. Comput. Chem.* **25**(13):1605-1612 (2004).
16. https://doi.org/10.1016/0263-7855(96)00018-5
17. Jmoldevelopment team. (2016). *Jmol*. Retrieved from http://jmol.sourceforge.net/
18. Fiser, A; Sali, A. (2003). "ModLoop: automatedmodeling of loops in proteinstructures". *Bioinformatics*. **19** (18): 2500–1. doi:10.1093/bioinformatics/btg362. PMID 14668246.
19. *"Whatis FASTA Format?". zhanglab.ccmb.med.umich.edu.* explains the FASTA format
20. PymodTool *: http://schubert.bio.uniroma1.it/pymod/*andModeller*:https://salilab.org/modeller/*
21. Lengauer T, Rarey M, *Computationalmethods for biomolecular docking*, in *Curr. Opin. Struct. Biol.*, vol. 6, n. 3, 1996, p. 402–6, DOI:10.1016/S0959-440X(96)80061-3, PMID 8804827.
22. Alder, B. J.; Wainwright, T. E. (August 1959). "Studies in Molecular Dynamics. I. General Method". *The Journal of ChemicalPhysics*. **31** (2): 459–466. Bibcode:1959JChPh..31..459A. doi:10.1063/1.1730376.
23. Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S. and Olson, A. J. (2009) Autodock4 and AutoDockTools4: automated docking with selectivereceptorflexiblity. *J. ComputationalChemistry* 2009, **16**: 2785-91.

24. Forli, S., Huey, R., Pique, M. E., Sanner, M. F., Goodsell, D. S., &Olson, A. J. (2016). Computationalprotein–ligand docking and virtualdrug screening with the AutoDock suite. Nature protocols, 11(5), 905-919.
25. CyrilDominguez, RolfBoelens and Alexandre M.J.J. Bonvin. HADDOCK: a protein-protein docking approachbased on biochemical and/or biophysical information. *J. Am. Chem. Soc.* **125**, 1731-1737 (2003).
26. *Raveh B *, London N * and Schueler-Furman OSub-angstrom Modeling of ComplexesbetweenFlexiblePeptides and GlobularProteins. ;Proteins, 78 (9): 2029-2040 (2010).*
27. *Landau, L.D.; Akhiezer, A.I.; Lifshitz, A.M. (1967). General Physics; mechanics and molecularphysics (First English ed.).*
28. *J. R. Forshaw; A. G. Smith (2009). Dynamics and Relativity. Wiley. ISBN 978-0-470-01460-8.*
29. Kutzner, et al. *J. Comput. Chem.*, 2015, **36** 1990-2008. DOI: 10.1002/jcc.24030
30. *Sun H, Mumby SJ, Maple JR, Hagler AT (April 1994). "An ab Initio CFF93 All-Atom Force Field for Polycarbonates". Journal of the American Chemical Society. **116** (7): 2978–2987. doi:10.1021/ja00086a030. ISSN 0002-7863.*
31. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz KM Jr, Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW, Kollman PA, *A Second Generation Force Field for the Simulation of Proteins, NucleicAcids, and OrganicMolecules*, in *J. Am. Chem. Soc.*, vol. 117, 1995, p. 5179–5197.
32. Charmm Force Field : http://mackerell.umaryland.edu/charmm_ff.shtml
33. *S. Jo, T. Kim, V.G. Iyer, and W. Im (2008)CHARMM-GUI: A Web-basedGraphical User Interface for CHARMM. J. Comput. Chem. 29: 1859-1865*
34. Frank Jensen (1999). *Introduction to ComputationalChemistry*. England: John Wiley and Sons Ltd
35. SteeredMolecular Dynamics Simulations for StudyingProtein–LigandInteraction in Cyclin-DependentKinase 5J. Chem. Inf. Model. 2014, 54, 2, 470–480Publication Date: January 17, 2014 https://doi.org/10.1021/ci4003574
36. Trezza, A., Iovinelli, D., Santucci, A. *et al.* An integrateddrugrepurposingstrategy for the rapididentification of potential SARS-CoV-2 viralinhibitors. *Sci Rep* **10,** 13866 (2020). https://doi.org/10.1038/s41598-020-70863-9
37. Van Rossum, G., & Drake Jr, F. L. (1995). *Pythonreferencemanual*. CentrumvoorWiskunde en Informatica Amsterdam.
38. Coutsias EA, Seok C, Dill KA (2004). "Using quaternions to calculate RMSD". *J ComputChem*. **25** (15): 1849–1857. doi:10.1002/jcc.20110. PMID 15376254. S2CID 18224579.
39. Frenkel, Daan&Smit, Berend. *Understandingmolecularsimulation: From algorithms to applications*. Academic Press, 196 (2nd Ed.), p. 97
40. *https://manual.gromacs.org/documentation/2018/onlinehelp/gmx-gyrate.html*
41. J. Chem. Phys. **133**, 044114 (2010); https://doi.org/10.1063/1.3462276
42. Coutsias EA, Seok C, Dill KA (2004). "Using quaternions to calculate RMSD". *J ComputChem*. **25** (15): 1849–1857. doi:10.1002/jcc.20110. PMID 15376254. S2CID 18224579
43. Kabsch W (1976). "A solution for the best rotation to relate two sets of vectors". *Acta Crystallographica*. **32** (5): 922–923. Bibcode:1976AcCrA..32..922K. doi:10.1107/S0567739476001873
44. Hurley JR, Cattell RB (1962). "The Procrustes Program: Producingdirectrotation to test a hypothesizedfactorstructure". *Behavioral Science*. **7** (2): 258–262. doi:10.1002/bs.3830070216
45. *Petitjean M (1999). "On the RootMeanSquare quantitative chirality and quantitative symmetrymeasures" (PDF). Journal of Mathematical Physics. **40** (9): 4587–4595. Bibcode:1999JMP....40.4587P. doi:10.1063/1.532988*

46. Petitjean M (2002). "Chiralmixtures" (PDF). *Journal of Mathematical Physics*. **43** (8): 185–192. Bibcode:2002JMP....43.4147P. doi:10.1063/1.1484559
47. Perrot, Pierre (1998). *A to Z of Thermodynamics*. Oxford University Press. ISBN 0-19-856552-6.
48. Genheden S, Ryde U. The MM/PBSA and MM/GBSA methods to estimate ligand-bindingaffinities. *Expert OpinDrugDiscov*. 2015;10(5):449-461. doi:10.1517/17460441.2015.1032936
49. Trezza, A., Iovinelli, D., Santucci, A. *et al.* An integrateddrugrepurposingstrategy for the rapididentification of potential SARS-CoV-2 viralinhibitors. *Sci Rep* **10,** 13866 (2020). https://doi.org/10.1038/s41598-020-70863-9
50. Wishart, D. S. et al. DrugBank 5.0: a major update to the DrugBank database for 2018. NucleicAcids Res. https://doi.org/10.1093/ nar/gkx1037 (2018).
51. Zhang, X. Direct anti-HCV agents. Acta Pharm. Sin. B https://doi.org/10.1016/j.apsb.2015.09.008 (2016)
52. Ma, C. et al. Boceprevir, GC-376, and calpaininhibitors II, XII inhibit SARS-CoV-2 viralreplication by targeting the viralmainprotease. Cell Res. https://doi.org/10.1038/s41422-020-0356-z (2020).
53. Peterson, L. In silicomoleculardynamics docking of drugs to the inhibitoryactive site of SARS-CoV-2 protease and theirpredictedtoxicology and ADME. Chemistry https://doi.org/10.26434/CHEMRXIV.12155523.V1 (2020)
54. *Simona Saponara, Fabio Fusi, Daniele Iovinelli, Amer Ahmed, Alfonso Trezza, Ottavia Spiga, Giampietro Sgaragli, Massimo Valoti,Flavonoids and hERGchannels: Friends or foes?, European Journal of Pharmacology ISSN 0014-2999,https://doi.org/10.1016/j.ejphar.2021.174030.*
55. Kumari, R., Kumar, R., Open Source DrugDiscoveryConsortium, Lynn, A., 2014. g_ mmpbsa–a GROMACS tool for high-throughput MM-PBSA calculations. J. Chem. Inf. Model. 54 https://doi.org/10.1021/ci500020m, 1951-1562
56. Wang, C., Nguyen, P.H., Pham, K., Huynh, D., Le, T.B., Wang, H., Ren, P., Luo, R., 2016. Calculatingprotein-ligandbindingaffinities with MMPBSA: method and erroranalysis. J. Comput. Chem. 37, 2436–2446. https://doi.org/10.1002/jcc.24467.
57. Saxena, P., Hortigon-Vinagre, M.P., Beyl, S., Baburin, I., Andranovits, S., Iqbal, S.M., Costa, A., IJzerman, A.P., Kügler, P., Timin, E., Smith, G.L., Hering, S., 2017. Correlationbetween human ether-a-go-go-related gene channelinhibition and actionpotentialprolongation. Br. J. Pharmacol. 174, 3081–3093. https://doi.org/ 10.1111/bph.13942.
58. Zhang, D.Y., Wang, Y., Lau, C.P., Tse, H.F., Li, G.R., 2008. Both EGFR kinase and Srcrelatedtyrosinekinasesregulate human ether-` a-go-go-related gene potassiumchannels. Cell. Signal. 20, 1815–1821. https://doi.org/10.1016/j. cellsig.2008.06.006.
59. Macdonald, L.C., Kim, R.Y., Kurata, H.T., Fedida, D., 2018. Probing the molecularbasis of hERGdrugblock with unnatural amino acids. Sci. Rep. 8, 289. https://doi.org/ 10.1038/s41598-017-18448-x.
60. *Landrum MJ, Lee JM, Benson M, Brown GR, Chao C, Chitipiralla S, Gu B, HartJ,Hoffman D, Jang W, Karapetyan K, Katz K, Liu C, Maddipatla Z, Malheiro A, McDaniel K,Ovetsky M, Riley G, Zhou G, Holmes JB, Kattman BL, Maglott DR. ClinVar: improvingaccess to variantinterpretations and supportingevidence. NucleicAcids Res. 2018 Jan4; 46 (D1): D1062-D1067. doi: 10.1093 / nar / gkx1153. PMID: 29165669; PMCID:PMC5753237.*

61. *Bongini, Pietro &Niccolai, Neri & Bianchini, Monica. (2019). Glycineinducedformation and druggability score prediction of proteinsurfacepockets. Journal of Bioinformatics and ComputationalBiology. 17. 10.1142/S0219720019500264*
62. *Feldmann, M. et al. Trials of anti-tumournecrosisfactortherapy for COVID-19 are urgentlyneeded. Lancet https://doi.org/10.1016/ S0140-6736(20)30858-8 (2020)*
63. *Hutter, M. C. Te currentlimits in virtual screening and propertyprediction. Future Med. Chem. https://doi.org/10.4155/fmc-2017-0303 (2018)*
64. *Saravanan, K., Kalaiarasi, C. &Kumaradhas, P. Understanding the conformationalfexibility and electrostaticproperties of curcumin in the active site of rhAChE via molecular docking, moleculardynamics, and chargedensityanalysis. J. Biomol. Struct. Dyn. https://doi.org/10.1080/07391102.2016.1264891 (2017)*
65. Hickson, L.J., Langhi Prata, L.G.P., Bobart, S.A., Evans, T.K., Giorgadze, N., Hashmi, S.K., Herrmann, S.M., Jensen, M.D., Jia, Q., Jordan, K.L., Kellogg, T.A., Khosla, S., Koerber, D.M., Lagnado, A.B., Lawson, D.K., LeBrasseur, N.K., Lerman, L.O., McDonald, K.M., McKenzie, T.J., Passos, J.F., Pignolo, R.J., Pirtskhalava, T., Saadiq, I.M., Schaefer, K.K., Textor, S.C., Victorelli, S.G., Volkman, T.L., Xue, A., Wentworth, M.A., WisslerGerdes, E.O., Allison, D.B., Dickinson, S.L., Ejima, K., Atkinson, E.J., Lenburg, M., Zhu, Y., Tchkonia, T., Kirkland, J.L., 2020. Corrigendum to 2019. Senolyticsdecreasesenescentcells in humans: preliminary report from a clinical trial of Dasatinib plus Quercetin in individuals with diabetickidneydisease. EBioMedicine 47, 446–456. https://doi.org/10.1016/j.ebiom.2019.12.004. EBioMedicine, 52, 102595
66. *Bongini P, Niccolai N, Trezza A, Mangiavacchi G, Santucci A, Spiga O, Bianchini M,Gardini S. Structuralbioinformaticsurvey of protein-small moleculeinterfacesdelineatesthe role of glycine in surface pocket formation. IEEE / ACM Trans ComputBiolBioinform. 2020 Oct 23; PP. doi: 10.1109 TCBB.2020.3033384. Epubahead of print. PMID: 33095703*
67. *Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Žídek A, Nelson AWR, Bridgland A, Penedones H, Petersen S, Simonyan K, Crossan S, Kohli P, Jones DT, Silver D, Kavukcuoglu K, Hassabis D. Improvedproteinstructurepredictionusingpotentials from deeplearning. Nature. 2020 Jan; 577 (7792): 706-710. doi: 10.1038 / s41586-019-1923-7. Epub 2020 Jan 15. PMID: 31942072.*
68. *Callaway E. 'Itwillchangeeverything': DeepMind's AI makesgiganticleap in solving proteinstructures. Nature. 2020 Dec; 588 (7837): 203-204. doi: 10.1038 / d41586-020-03348-4. PMID:33257889*
69. *Bongini, P., Bianchini, M., & Scarselli, F. (2021). Molecular generative GraphNeuralNetworks for DrugDiscovery. Neurocomputing. 2021. doi: 10.1016 / j.neucom.2021.04.039*
70. Spiga, O., Cicaloni, V., Fiorini, C. *et al.* Machine learningapplication for development of a data-drivenpredictive model able to investigate quality of life scores in a rare disease. *Orphanet J Rare Dis* **15,** 46 (2020). https://doi.org/10.1186/s13023-020-1305-0
71. *Aaby, Anthony (2004). Introduction to Programming Languages. Archived from the original on 8 November 2012. Retrieved 29 September 2012.*
72. "General Python FAQ — Python 3.9.2 documentation". *docs.python.org*. Archived from the original on 24 October 2012. Retrieved 28 March 2021.

73. *Piotrowski, Przemyslaw (July 2006). "Build a Rapid Web Development Environment for Python Server Pages and Oracle". Oracle Technology Network. Oracle. Archived from the original on 2 April 2019. Retrieved 12 March 2012.*
74. *Batista, Facundo (17 October 2003). "PEP 327 – Decimal Data Type". Python Enhancement Proposals. Python Software Foundation. Archived from the original on 4 June 2020. Retrieved 24 November 2008.*
75. Eby, Phillip J. (7 December 2003). "PEP 333 – Python Web Server Gateway Interface v1.0". *Python Enhancement Proposals*. Python Software Foundation. Archived from the original on 14 June 2020. Retrieved 19 February 2012
76. *Science education with SageMath, Innovative Computing in Science Education, archived from the original on 15 June 2020, retrieved 22 April 2019*
77. "OpenCV: OpenCV-PythonTutorials". *docs.opencv.org*. Archived from the original on 23 September 2020. Retrieved 14 September 2020.
78. *Dean, Jeff; Monga, Rajat; et al. (9 November 2015). "TensorFlow: Large-scale machine learning on heterogeneoussystems" (PDF). TensorFlow.org. Google Research. Archived (PDF) from the original on 20 November 2015. Retrieved 10 November 2015.*
79. *"Python Setup and Usage". Python Software Foundation. Archived from the original on 17 June 2020. Retrieved 10 January2020.*
80. Chapman, Brad; Chang, Jeff (August 2000). "Biopython: Pythontools for computationalbiology". *ACM SIGBIO Newsletter*. **20** (2): 15–19. doi:10.1145/360262.360268. S2CID 9417766.
81. "4.0 New Features and Fixes". *LibreOffice.org*. The Document Foundation. 2013. Archived from the original on 9 February 2014. Retrieved 25 February 2013.
82. SupervisedMolecular Dynamics (SuMD) as a HelpfulTool To Depict GPCR – LigandRecognitionPathway in a Nanosecond Time ScaleDavide Sabbadin and Stefano MoroJournal of Chemical Information and Modeling 2014 54 (2), 372-376DOI: 10.1021 / ci400766b
83. Simeprevir Potently Suppresses SARS-CoV-2 Replication and Synergizes with RemdesivirHo Sing Lo, Kenrie Pui Yan Hui, Hei-Ming Lai, Xu He, Khadija Shahed Khan, Simranjeet Kaur, Junzhe Huang, Zhongqi Li, Anthony KN Chan, Hayley Hei-Yin Cheung, Ka-Chun Ng, John Chi Wang Ho, Yu Wai Chen, Bowen Ma, Peter Man-Hin Cheung, Donghyuk Shin, Kaidao Wang, Meng-Hsuan Lee, Barbara Selisko, Cecilia Eydoux, Jean-Claude Guillemot, Bruno Canard, Kuen-Phon Wu, Po-Huang Liang, Ivan Dikic , Zhong Zuo, Francis KL Chan, David SC Hui, Vincent CT Mok, Kam-Bo Wong, Chris Ka Pun Mok, Ho Ko, Wei Shen Aik, Michael Chi Wai Chan, and Wai-Lung NgACS Central Science 2021 7 (5), 792-802DOI: 10.1021 / acscentsci.0c01186
84. Zhao Y, Marcink TC, Sanganna Gari RR, et al. Transient collagen triple helix binding to a key metalloproteinase in invasion and development. *Structure*. 2015;23(2):257-269. doi:10.1016/j.str.2014.11.021

AKNOWLEDGMENTS

Prof. Ottavia Spiga – University of Siena

Prof. Annalisa Santucci – University of Siena

Prof. Lorenza Trabalzini – University of Siena

Prof. Neri Niccolai – University of Siena

Ph.D. Alfonso Trezza – University of Siena

Ph.D. Francesco Pettini – University of Siena

Ph.D. Anna Visibelli – University of Siena

Ph.D. Vittoria Cicaloni – University of Siena

Ph.D Filippo Prischi – University of Essex