PROGRESS REPORT

Final Year Project 1

Semester1, 2020/2021

Bachelor of Computer Science

A. Project Information

Student(s)

1. Mohamad Zaki bin Jaafar (1719207)
2. Ahmad Zaidan bin Adnan (1718733)

Project ID

620D

Project Title

Pharmaceutical Drug Discovery Intelligent System

Project Category

Development

Supervisor

Dr. Sharyar Wani

B. Introduction

Project Overview

Drug discovery is basically a system of processes in pharmacology to discover a new prospect of medication, before going to phase of drug's clinical triasl. In this project, an intelligent system is going to be developed after researching on the best algorithm to be used for a set of features in assisting the drug discovery process. After fetching the datasets, we clean, process and analyze the data to get the insights in order to the train the machine learning models that will aid the procedures of drug discovery.

Problem Statement

The diseases that affected the human being are known since the beginning of the human history. In this millennium, the treatment has evolved, our modern society have invented the scientific method to discover and create the pharmaceutical drug for variety of medical purposes. But in the recent development we have seen that there are possibilities that health disasters can happen at anytime and anywhere, like the world-wide pandemic COVID-19. Thus, we need to prepare for the worse and maximize the use of current technology, and one of them is the machine learning. The traditional drug discovery process takes a lot of time to be done, but the clinical trials after that are much longer. Hence, the drug discovery must be done efficiently and effectively as possible so that the success rate of the trials is at its maximum thus will reduce the chance of wasting the great amount of time and resources only to produce the fail drugs.

Project Objectives

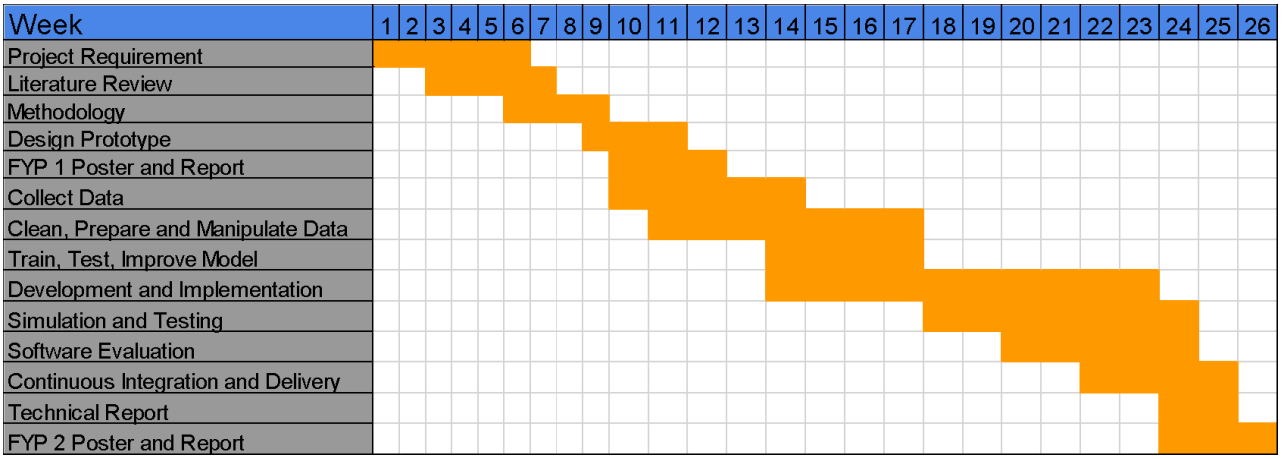
1. To provide an efficient and effective solution to assist the processes in drug discovery

2. To discover the most effective inhibitors for certain target proteins.

Significance of Project

This project can help the one who is involved in the drug discovery research and industry as it automates some steps need to be taken such as determining whether the molecules fulfill the Lipinski's rules or not.

Project Schedule



C. Review of Previous Works

A. Background Review

Introduction

Different sectors today are advancing their machine learning technologies, including the healthcare and the pharmaceutical industry. Machine learning was initially defined as a programme that learns automatically from data to perform a task or make a decision, rather than having the action programmed directly. This definition however is very broad and may include almost every sort of approach driven by data. It could be more helpful to picture machine learning as an algorithm that operates in a continuum between completely human-guided and fully machine-guided data processing (Beam & Kohane, 2018). To find patterns in vast volumes of data, machine-learning algorithms use statistics. And data, here, contains a lot of material, figures, phrases, photos, clicks and more. It can be fed into a machine-learning algorithm if it can be processed digitally. Machine learning is the mechanism that drives all of the services people use today, such as bioinformatics research system like the use of EvalCon and EvalG algorithm in the Pathosystems Resource Integration Centre (PATRIC) for genomes quality estimation (Parrello et al., 2019), privacy-preserving machine learning as a service system like Chiron (Hunt et al., 2018), open-source frameworks like PredictionIO, and cloud-based services offered by Amazon, Google, Microsoft and BigML (Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, 2016). Each platform gathers as much data as possible from various environments and using machine learning to make a highly informed prediction about the possibilities of what could happen. This method, honestly, is very basic: find the pattern, apply the pattern. Yet the world today runs pretty much like that.

In this literature review, the main sector that will be focused on is the drug discovery industry and the machine learning approach that assist the process of drug discovery. In combination, core clinical health Artificial Intelligence (AI) applications could theoretically produce US$150 billion in annual savings for the healthcare economy in the United States alone by 2026, according to Accenture Research (Srivastava & Srinivasan, 2020). McKinsey Global Institute estimated that technology companies invested $20 billion to $30 billion on AI globally in 2016, with 90% of this spent on R&D and deployment, and 10% on AI acquisitions (Bughin, Jacques ;Hazan, Eric; Manyika, James; Woetzel, 2017). Funding, loans, and seed contributions for AI from venture capital and private equity firms have both risen exponentially, although from a small base, to a combined amount of $6 billion to $9 billion (Bughin, Jacques ;Hazan, Eric; Manyika, James; Woetzel, 2017). Almost 60 % of the funding for AI are specifically funded for machine learning, most likely because it is an enabler for so many other innovations and applications as well as investors believe installing new code is simpler and better than rebuilding a computer or another machine that runs the software (Bughin, Jacques ;Hazan, Eric; Manyika, James; Woetzel, 2017). The statistics demonstrate that the possibilities presented by machine learning would be highly leveraged by the healthcare industry. That is why AI firms are interested in diverse practices from diagnosis to therapy to drug production in the treatment process.

Throughout this review, the reader is briefly overviewed on the aspects of machine learning in drug discovery. From the potential of Artificial Intelligence to the challenges of machine learning in the field to applications of this algorithm towards drug discovery, it will all be covered in this literature review.

The Potential of Artificial Intelligence in Drug Discovery

Computational drug discovery strategies help reduce the cost of delivering medicines to the market dramatically. Since AI is mostly related to robots, robots here refer to the medical robots that can also assist researchers in health industry especially drug discovery. In 2017, medical robot sales accounted for a 2.7 % increase in all professional robot service sales, with sales rising 73 % over 2016 (Ahmed & Hossain, 2019). From an estimated $6.46 billion in 2018, the medical robot industry is expected to hit $22.10 billion by 2027, at a CAGR of 14.6 % (Ahmed & Hossain, 2019). On the contrary, Grand View Research estimates that the demand for medical robotic systems will rise at an average compound annual growth rate (CAGR) of 12.6 % from 2014 to 2020 and will be worth $18 billion by 2020, due to the decline in the price of sensors and software (Ahmed & Hossain, 2019). From this overview, it can be seen that the market's valuation is increasingly increasing with artificial intelligence being used in drug discovery. The population of the planet is growing rapidly, and life expectancy is increasing. More individuals need more supply, such as growing demands for healthcare and logistics for life sciences. UPS reports that healthcare logistics is a global industry worth $70 billion, rising at an annual rate of 4% (Lehmacher, 2017). Therefore, a lot of opportunities are offered for many technological advances especially the newly developed machine learning algorithms.

The technical and paradigm change seen in the pharmaceutical industry to machine learning helps researchers to assist the method by using innovative computational algorithms. Since biomedical data is extremely complex, it has become more feasible than ever to use algorithms to develop new drugs. Machine learning can improve a variety of steps in the drug development process such as Lead optimisation, Hit-to-lead and Hit generation (Renaud et al., 2016). This is a broad process but mostly, machine learning assists in the preliminary yet critical steps concerning the nature of the chemical composition of a drug (Chen et al., 2018). Furthermore, machine learning is involved in both fundamental preclinical research and clinical trials, in which a lot of biomedical data is generated, the impact of a drug is studied. Machine learning may promote the discovery of new patterns in such results. There are various kinds of data, including those that are genetic and imaging (Nistor & Hernández-Garcíac, 2018). With machine learning, both of them can be analysed and further used to create novel drug development solutions.

Challenges in Machine Learning for Drug Discovery

One of the principal problems in the field of drug development is ensuring drug safety. It is a difficult challenge to analyse information about the known effects of medications and to predict their side effects. In order to extract useful information from clinical evidence collected in clinical trials, scientists and engineers from academic institutions and pharmaceutical firms have been attempting to use machine learning. For example, a random forest classifier was used in the field of drug safety to predict the effect

of drugs on the foetus (Ekins et al., 2020). Furthermore, a serious clinical pharmacology or toxicology issue that has created enormous medical and societal pressures is the adverse drug reaction (ADR). It is also one of the two key reasons that contribute to inability to develop new medicines. As a solution, an integrated machine learning model and its ADRAlert-gene web service was introduced for rapid drug safety evaluation using a dynamic network of drug-gene-ADR learning (Liu et al., 2020). Therefore, the analysis of this whole data in the light of drug safety is an active field of study.

The most costly stage of the production of drugs is clinical trials. It is important to use the expertise obtained during prior clinical trials in the early stages of the production of medications to reduce their prices. Clinical trials take 15 years on average and $1.5 to $2.0 billion to bring a new drug to the market (Harrer et al., 2019). According to Deloitte Analysis, the cost drivers in clinical trials are patient recruitment, outsourcing costs, site recruitment, clinical trial management system and other technology, site retention, data management and validation, and patient retention (Properzi et al., 2020). In order to reduce the costs of clinical trials, it can be accomplished in two procedures. Firstly, biomedical results from laboratory trials may be analysed and interpreted to forecast the efficacy and side effects of a drug using machine learning. As an example, Trials.ai uses this procedure. To facilitate study design, AI is used to evaluate vast collections of genomic data, journal papers, previous clinical trials and other types of analysis. The system is able to unlock knowledge, draw insights and make suggestions to trial sponsors on how to better plan and refine their trial protocols using its patented codified clinical trials database, as well-designed protocol constraints increase recruiting and retention and reduce the pressure on patients and trial sites by increasing cost and time performance (Properzi et al., 2020).

From another perspective, biopharmaceutical firms will draw on the digitalization of healthcare in the future and remotely control clinical trials by the establishment of a hub-and-spoke command centre. One known example is Novartis. By establishing a central 'control tower' to digitally track and troubleshoot clinical trials taking place around the globe in real time, known as Nerve Live, Novartis is now tackling its research and development expenses. This framework uses ML with predictive algorithms to help Novartis in its clinical trials 12, 18 or even 24 months down the road to detect possible logjams, enabling the company to take steps to avoid delays (Properzi et al., 2020). Secondly, biological data analysis can be assisted by data from clinical trials examined using machine learning. Gottlieb said the US Food and Drug Distribution (FDA) are creating a new regulatory framework aimed at encouraging the use of AI. The FDA has developed Master Clinical Trial Protocols (MAPs) unique to clinical trials, designed to improve reliability and lower the cost of developing clinical trials (Kent, 2018). It is possible to design better preclinical experiments with those two methods built simultaneously to come up with the most promising treatments with the fewest side effects.

Integrating Biomedical Data with Computational Approaches

Through combining biomedical and clinical data with computational models, machine learning may help improve treatment, and can be used to develop drug testing tools and combinatorial therapies. Any clinical data integration computer models and

methods are still under development, but there are also a few very strong examples of effective data integration in biology and medicine.

A factor graph-based probabilistic graphical models (PGM) method, PAthway Recognition Algorithm (Vaske et al., 2010) using Data Aggregation on Genomic Models (PARADIGM), which was proposed to combine copy number and gene expression with curated National Cancer Institute (NCI) pathway information provides patient-specific inference of genetic pathway operations. In order to provide patient-specific genomic inferences on the status of gene functions, complexes and cellular processes, the PARADIGM approach combines various high-throughput genomics data with known signalling pathways. To integrate the different data sets, the cornerstone of the approach uses a factor graph to leverage inference. Using these inferences instead of, or in combination with the initial high-throughput datasets increases the ability to distinguish samples into subtypes that are clinically important. Patient subtypes associated with various survival profiles were revealed by clustering the glioblastoma multiform (GBM) patients based on the PARADIGM-integrated behaviours. Clustering the samples using either expression data or copy-number data, on the other hand, did not disclose any relevant clusters in the dataset. Overall, inferred cellular behaviours by PARADIGM helped to distinguish patients into sub-groups that are clinically important.

For multi-omics data integration, a multiple imputation (MI)-based approach, referred to as MI for multiple factor analysis (MI-MFA), was recently suggested (Voillet et al., 2016). Due to its scalability to a large number of variables with missing values, MI-MFA used hot-deck imputation, which is a non-parametric technique widely used in large surveys. The missing value on a variable is replaced by the observed value from a related sample or donor to conduct hot-deck imputation. The most significant drawback of other model-based approaches (such as joint modelling (JM) and fully conditional specification (FCS)) is overcome by the hot-deck method. A deficiency is that good donor-recipient matches are needed. It is not an easy job to locate such good matches and it is outside the reach of this report (Voillet et al., 2016). A probable shortcoming of the approach is that there could be very few donor findings in the donor pools and therefore a chance of bias in the outcomes of the MFA.

XCMS online (Warth et al., 2018) is a cloud-based predictive pathway analysis data mining tool for metabolomics that allows multi-omics data analysis by combining metabolic pathways with gene and protein data. While metabolite mapping may provide an indication of gene and protein function, a more detailed and validated characterisation of a system under analysis is given by combining these metabolomics findings with proteomics and transcriptomics details. As a coherent and intuitive workflow that harnesses cloud-based multi-omic technologies, XCMS Online can now be used to perform metabolomics-guided systems biology research. Another cloud-based tool for integrative metabolomics research is MetaboAnalyst (Chong et al., 2018). Via knowledge-based network analysis and different ML-based clustering, feature selection and classification algorithms, it implements modules for multi-omics data integration.

Genetic Data Analysis and Personalized Medicine

Many pharmaceutical firms and start-ups rely on the analysis of genomic evidence and precision medicine. Understanding the genetic profile of the patient allows delivering optimal medications and treatment. With machine learning, statistical methods may be built to evaluate genetic data and suggest new therapies. There are only a few examples that impact current clinical practice based on machine learning solutions which carry tremendous potential to personalised medicine and drug discovery. They include the development of novel drug reaction biomarkers and computational methods focused on machine learning used in clinical practice. These methods are used on the basis of genotype analyses to predict the susceptibility to individual drugs and to combinatorial therapies. For example, Geno2pheno (Lengauer & Sing, 2006) is a computational method used in clinical practice to estimate HIV tolerance to an individual drug and to viral genotype-based combinatorial therapies. Another example is gene signature (S3 score) for prognostic prediction in patients with clear renal cell carcinoma (Büttner et al., 2015).

One of the potential methods is to view the genetic code as a one-dimensional image and then to apply a regular algorithm for machine learning (Chen et al., 2018). The data is then searched for patterns and anomalies, as has been done in many other image detection initiatives. In fact, the study of genomics can be used in the same way as it is used in classical paintings when it comes to recognising art style (Lecoutre et al., 2017). In order to achieve over 62 % accuracy on WikiArt results, the researchers successfully applied a deep learning approach. The use of a residual neural network and the importance of retraining were primarily due to this development. The existence or shape of an image to be analysed is meaningless to the algorithm, so the computer is equally successful in analysing a single-dimensional DNA chain or some other form of image data.

As genomic data is typically viewed as a string of letters, Natural Language Processing techniques may also be applied (Chen et al., 2018). The gain of doing so is that it extends the field that can be processed by the algorithm. When unique changes or patterns are being searched, that may be significant, or the pattern to be found consists of a longer sequence of genes. An example of the use of Natural Language Processing techniques is its application in understanding temporal evolution of COVID-19 research (Ebadi et al., 2020). In this study, the researchers concentrated on two separate sources of evidence which are PubMed and ArXiv, and used techniques of machine learning and natural language processing to better understand the landscape and evolution of COVID-19 research over time. Their detailed studies, undertaken at multiple levels of granularity, could allow decision-makers and policymakers to better understand the complexities of COVID-19 research that could help set or adapt strategies.

A significant challenge is to truly unleash the capacity for drug exploration and personalised medicine through machine learning. In order to thoroughly recreate genetic networks on the basis of expression data, time series data may be useful. In time series, expression genetic data and sequencing data should be acquired to create robust predictive models based on machine learning. A novel algorithm, MICRAT (Maximal Information coefficient with Conditional Relative Average entropy and Time-series mutual information) (Yang et al., 2018), was proposed to infer gene regulatory networks (GRNs) from time series gene expression data by taking into account the dependency and time delay of regulator

expresses and their target genes. To represent the underlying relationships between genes, this technique used maximum information coefficients to recreate an undirected graph.

Innovative start-ups, including Cambridge Cancer Genomics (Eisenstein, 2020), use machine learning to interpret data gained from liquid biopsy, a diagnostic technology in which circulating tumour cells or cell-free DNA is extracted from blood samples. The flagship tool of the firm, OncOS, is intended to work with data obtained from either traditional biopsies or less invasive liquid biopsy assays that classify the circulating tumour DNA in the bloodstream. While it is not a fully standardised method for cancer therapy control, owing to its potential to collect genetic data in time series during treatment, it is highly awaited in personalised medicine. To better grasp these data and address the question of why cancer grows, applying machine learning may help scientists devise less toxic therapies.

Building and Getting Insight from Databases and Datasets

To support medical doctors in their daily practise, scientists use online clinical data archives to solve major challenges in hospitals, as medical information can be derived from public repositories. For drug exploration purposes, these libraries may also be used to provide clinical details in the early stages of drug production. In a research (Bustos & Pertusa, 2018), different classifiers which are FastText and Convolutional Neural Networks (CNN) with pre-trained word-embeddings, k-Nearest Neighbors (kNN), and Support Vector Machines (SVM), have been trained, validated, and compared on a corpus of cancer clinical trial protocols from www.clinicaltrials.gov. Short free-text sentences detailing health details including medical history, concomitant treatment, tumour type and characteristics such as molecular profile, cancer therapy and more, are listed by the models as eligible or not eligible conditions for voluntary inclusion in these trials.

Attempts have been made to use deep neural networks to reflect medical information. Due to improved compatibility with data structures generated with similar methods, data mapped with machine learning can also be easier to combine with biomedical data analysed with machine learning.

Also exciting are recent achievements in developing databases for machine learning purposes. For example, a database for extremely scalable queries and analysis of cancer-related data through different data forms and multiple studies was developed by the authors of the paper "Integrative analysis and machine learning on cancer genomics data using the Cancer Systems Biology Database (CancerSysDB)" (Krempel et al., 2018). The CancerSysDB allows extremely versatile analyses of cancer data across various OMICS data forms and clinical data. In medicine and drug discovery, however, there are several questions that are very difficult to address only on the basis of public data, which is missing if better machine learning models and methods are to be created.

It is not only the manner in which data are pre-processed that must be grasped, but also the concepts of the use of various bioinformatics methods and interdisciplinary expertise in biomedicine and where computer science and medicine intersect, if proper databases are to be developed to address specific scientific questions. Also, small volumes of data from online libraries could be best exploited by teams who have this experience and

expertise. In general, machine learning engineers collect data from scientists, physicians, pharmaceutical firms and hospitals, so the number is small. But for outcomes to be obtained, models must be solid.

"Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures" (Özyurt, 2020) was one of the best examples of developing a model of superior power, one that can cope with the lack of sufficient data. And if there were just a few images presented in the dataset, the model was designed to recognise certain features from an aerial photograph. The most common pre-trained architectures such as Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet have been used as extractor features in the present analysis. Close cooperation and shared comprehension of various languages and disciplines is needed to gain profound insight from data. If there is just the occasional consultation, that is rough.

Primary members of teams developing databases, datasets, machine learning algorithms, methods and applications for biomedical data analysis and drug discovery are scientists with a combined biomedical and analytical background. Leading institutions such as ETH Zurich (Zurich, 2018) have also begun educating a new generation of computer-based and mathematical medical scientists and have created a forum and interdisciplinary teams to evaluate clinical and biomedical data. New approaches to data science and deep learning are being developed by researchers at ETH Zurich. They are active in interdisciplinary research centres with other institutions; the Swiss Data Science Centre in Zurich and Lausanne is jointly run by ETH Zurich and EPFL, drawing together the skills of data experts and providing an interdisciplinary forum that improves education and knowledge transfer. For instance, ETH Zurich trains Master's students in data science as part of this national initiative. In clinical practise, the Swiss Tumour Board and ETH Personalised Health Technologies Platform Nexus are collaborating together to introduce individualised, biomarker-based medical decisions. In advancing drug development through machine learning, these are critical foundational moves.

Standard Machine Learning Approaches for Genetics and Genomics

In order to analyse genetic data such as microarray or RNA-seq expression data, standard supervised, semi-supervised and unsupervised machine learning algorithms are applied (Bogdan et al., 2018). In imaging genetics and genomics, machine learning algorithms are starting to be used to forecast or diagnose disease effects, which is perhaps the most direct therapeutic use of such psychiatric approaches. The use of these techniques in genetics and genomics imaging has generally focused on well-characterized candidate genes, but there are also new data-driven analyses. Although in its infancy, a potential future path of discovery that could have major therapeutic implications is considering clinical, neurological and genetic characteristics in tandem for disease prediction. In addition, these algorithms can expose illnesses and stable phenotypes, and the medication modes of action can be further discovered. The researcher must determine which data to include as input to the algorithm to address complex biomedical questions in every implementation of machine learning methods.

There are a range of detailed articles that summarise the use of large-scale genomic data processing and machine learning techniques to address issues of genomic sequencing, such as identifying unique regions in sequences and understanding transcriptomic site positions (Telenti et al., 2018). Among functional implications, it is one of the greatest problems in genomics. Convolutional neural network (CNN) applications have been applied for predicting DNA-binding from sequence (Jones et al., 2017). Such applications are DanQ (a hybrid convolutional and recurrent deep neural network for quantifying the role of DNA strings), Basset (learning the open genome's regulatory code with deep convolutional neural networks), DeepBind and DeeperBind (predicting the sequence specificities of DNA- and RNA-binding proteins), DeepMotif (visualizing the classification of genomic sequences), and convolutional neural network architectures for predicting DNA–protein binding. For this application, machine learning has promise, but evidence from experimental tests or clinical trials can verify the results generated with machine learning algorithms. In genome interpretation and analysis of genetic variants, deep learning algorithms may be useful, a complex task which needs a combination of robust biological data and clinical knowledge.

Due to artificial learning, scientists and engineers have recently taken a step towards a deeper understanding of the human genome (Madhukar et al., 2015). The ability to solve challenging biomedical prediction issues can be greatly enhanced by supervised heterogeneous ensemble approaches. In order to generate more detailed forecasts, ensembles assimilate a wide number and variety of base predictors and aim to match their collective success and diversity. On several broad genomics datasets, two proven heterogeneous ensemble methods were rigorously tested, namely stacking and ensemble selection. These techniques' superior predictive potential, especially aggregated stacking can be attributed to their attention to the following aspects which are better balance of diversity and efficiency. Even, there is a preliminary stage in the application of machine learning algorithms to genomic problems. Genomic and genetic data are multidimensional, after all and probabilistic machine learning algorithms for their analysis still need to be created.

Machine Learning Approaches for Network Analysis of Biomedical Data

Genetic data analysis may be useful in elucidating genetic networks, which will show the mechanism of action of a drug and better explain how diseases operate. This comes under the framework of a new emerging field called network medicine. A pioneer in network medicine, the Barabasi group (Guney et al., 2016) notes that an unsupervised network-based approach helps novel drug-disease interactions to be anticipated, providing substantial possibilities for discovering new drug applications and for detecting possible side effects. The group has observed that in a specific network neighbourhood, the therapeutic impact of medications could be localised. This indicates that many genes in close proximity to genes linked to the function of a disease in the network may be exploited to treat the disease efficiently.

Machine learning analysis of genetic network data may help to find novel drug targets and predict the optimal drug combination. For biological network analysis, there are research papers that describe how to benchmark machine learning. In order to classify cancer-associated gene pairs, a research (Ghanat Bari et al., 2017) illustrates the use of support vector machine (SVM) models combined with machine learning-assisted network

inference (MALANI). These can be used to rebuild cancer networks to classify main cancer genes that would normally go undetected by traditional methods in high-dimensional data space. For other machine learning and feature selection methods, these algorithms should be similarly valid. Another research is the use of node classification in biological network analysis mainly for protein function prediction (Muzio et al., 2020). Predicting the unknown function of a protein based on the functions of its neighbours in a PPI network is a standard role of node classification in biological network analysis. However, non-standard machine learning algorithms are still being developed for study of complex biological networks, and network and machine learning methods require closer integration.

Machine Learning Algorithms in Image Analysis for Drug Discovery

According to Scheeder et al., their study presents how image-based screening of high-throughput experiments in which drug-treated cells may help to elucidate the mechanism of action of a drug (Scheeder et al., 2018). It is mentioned that unsupervised and simplistic methods of statistical inference appear to be in favour of evaluating image data from large-scale profiling studies, but with supervised algorithms, complex biological phenotypes and single-cell experiments may be effectively categorised. Image-based profiling experiments have shown the potential to enhance the pre-clinical production of small molecules at almost every point of the pipeline, from target recognition to activity prediction systems to toxicity profiling. It would continue to expand the methodological portfolio of cellular screens to assist the drug discovery process by growing the throughput and extending more nuanced research methods of image-based phenotypic screens and profiling approaches.

The recently explored use of supervised learning, especially deep neural networks, in image-based profiling, may be a novelty identification mechanism to recognise unexpected phenotypes discovered in the process of drug discovery (Kusumoto & Yuasa, 2019). With deep learning, it is only from its structure that the properties of a molecule can be predicted. The approach involves the use of a convolutional neural network capable of extracting a molecule's shape and then confronting it with the information obtained about the properties. A label-free cell recognition method called silico labelling has been developed by Christiansen et al., which enables nuclei, cell type, and cell state to be recognised from bright field microscopy images without immunolabeling (Christiansen et al., 2018). Hematopoietic stem cells have multipotency and can be differentiated into all sorts of lineages of blood cells. The technique of deep learning will classify with high precision the final hematopoietic lineage of differentiated cells from microscope images.

Novel Machine Learning Algorithms under Way

Quantum machine learning analysis indicates that this technique can be useful for discovering difficult data patterns (Biamonte et al., 2017). Given the complexities of biological and medical data, probabilistic quantum machine learning algorithms are a real opportunity to better understand them. According to Mishra et al., there are three major specialisations of quantum technology today which are quantum computing that uses quantum phenomena for computational tasks, quantum information that uses quantum phenomena for information transfer, retrieval and reception, and quantum cryptography

that uses quantum phenomena to devise superior cryptography techniques (Mishra et al., 2020). Moving to apply quantum computing and quantum machine learning to drug development, ground-breaking pharmaceutical firms or start-ups have based these activities primarily on predicting the structure of new medicines. For example, high-performance cloud computing and modern algorithms from AMGEN Research (Mardirossian et al., 2020) make efficient fully quantum mechanical protein-ligand scoring. In quantum computing for life sciences such as ProteinQure (Cao et al., 2018), substantial venture capital has been spent, and potential studies will further unveil the extent of quantum gain for broader protein folding issues as quantum devices scale up. Finally, genomics and systems biology are two essential fields in which modern machine learning techniques can be implemented on the basis of in-depth study of biomedical data in order to generate fewer dangerous drugs.

Conclusion

Throughout the literature review, its main focus is to review the possibilities and opportunities of machine learning in the area of drug discovery. The review started by introducing the potential of Artificial Intelligence (AI) which are one subset from machine learning. A 2.7 per cent increase in all technical robot operation revenue in 2017 was accounted for by medical robot sales. In the drug production process, machine learning can facilitate a number of steps. By 2027, the medical robot market is projected to cross $22.10 billion, at a 14.6 per cent CAGR. As every field has its own challenges, drug discovery has its own set of challenges that must be solved by the application of machine learning. One of them is clinical trials. It is important to use artificial learning to derive valuable knowledge from clinical data obtained in clinical trials. A global 'control tower' was set up by Novartis to remotely monitor and troubleshoot clinical trials taking place in real time around the world. In the future, biopharmaceutical corporations will build on healthcare digitalization to remotely monitor clinical trials by creating a hub-and-spoke command centre.

In order to apply machine learning methods towards drug discovery, the integration between biomedical data and computational approach is significant since machine learning is mostly relied on the basis of data. Therefore, in the literature review, the reader is demonstrated with a few examples of this type of data integration. The PARADIGM technique blends different results from high-throughput genomics with known signalling pathways. Hot deck imputation, which is a non-parametric technique typically employed in broad surveys, was used by MI-MFA. XCMS online is a data mining platform for metabolomics focused on cloud-based computational pathway analysis. It facilitates data interpretation of multi-omics by integrating metabolic pathways with data on genes and proteins. It can also be used to conduct metabolomics-guided biology experiments on structures. Statistical approaches to test genetic data and propose alternative treatments can be developed using machine learning. The view of the genetic code as a one-dimensional diagram is one of the possible approaches. For example, outside the field of medicine, researchers successfully applied a deep learning approach to the findings of WikiArt, which obtained over 62 per cent accuracy. Another instance, Cambridge Cancer Genomics can view data gained from liquid biopsy using machine learning. To consider the landscape and evolution of COVID-19 study over time, researchers used methods of deep learning and natural language processing.

Database remains as an important factor in a machine learning project. To overcome major problems in hospitals, scientists use public clinical data collections. In the early stages of drug development, these libraries can also be used to provide clinical information. The templates list brief free-text sentences listing health specifics as qualifying or not qualifying criteria for voluntary participation in trials. Machine-learning mapped data can also be easier to integrate with biomedical data. Due to that, researchers at the University of ETH Zurich are exploring new approaches to data science and deep learning. To predict or detect disease symptoms, machine learning algorithms are beginning to be used. These algorithms will reveal diseases and healthy phenotypes, and it is possible to further explore the modes of action of medication. For predicting DNA-binding from sequence, CNN applications were applied. Perhaps this is the most direct clinical application of psychiatric methods like this. While in its infancy, a theoretical future direction of exploration may have important clinical effects.

One machine learning approach is for network analysis of biomedical data. This falls under the framework of network medicine, a new emerging field. An unsupervised, network-based approach aims to predict new drug-disease experiences. The clinical effect of drugs could be clustered in a single network neighbourhood. Another machine learning approach in drug discovery is in image analysis. The ability to improve the pre-clinical development of small molecules has been shown by image-based profiling experiments. Silico labelling helps nuclei, type of cell, and cell status to be identified without immunolabeling from microscopy images. A new modern approach in machine learning is quantum machine learning. Quantum analysis in machine learning reveals that this method can be useful for finding complicated patterns in results. Probabilistic quantum machine learning algorithms are a real opportunity to better understand them, considering the complexity of biological and medical evidence. These efforts have been focused mainly on forecasting the structure of novel drugs by ground-breaking pharmaceutical giants or start-ups. The two basic areas in which advanced machine learning techniques can be applied are genomics and system biology.

As a result from all the reviews taken from many articles and research, a few limitations have been found in the field of machine learning in drug discovery. The expertise of applying machine learning into drug exploration requires sufficient knowledge in both data science and pharmaceutical knowledge. To succeed in this kind of machine learning project, the project must be guided by experts who have known knowledge in sciences and technology. Another limitation is the lack of database for some diseases especially new ones such as COVID-19. It is more approachable to create this project for common diseases. Common diseases usually contain a large amount of database. In addition, one type of machine learning approach must be chosen to apply on certain drug discovery project. This is due to the variety and uniqueness of machine learning approach for certain disease. One disease might require image detection approach. In the other hand, another disease might require CNN approach. Finally, there is no system that can predict the composition of drugs or medications for every type of diseases known in the globe today. Commonly, the systems developed are specific for certain disease or in other words, one disease, one drug, one system. This limitation is mostly related to the different approach needed for every disease, stated earlier in this paragraph.

B. Product Review

Software	Description	Platform	Developer
Anduril	Anduril is an open source, component-based workflow platform developed at the Systems Biology Laboratory, University of Helsinki, for scientific data analysis. Anduril is designed to allow comprehensive, scalable and effective data analysis, especially in biomedical research in the field of high-throughput experiments. Sequencing, gene expression, SNP, ChIP-on-chip, comparative genomic hybridization and exon microarray research, as well as cytometry and cell imaging analysis, are currently components of the workflow framework for many forms of study.	Linux, macOS, Windows	University of Helsinki
BioPHP	BioPHP is an open-source PHP programming set of classes for study of DNA and protein sequences, alignment, database parsing, and other resources for bioinformatics. As an open-source bioinformatics initiative, the Open Bioinformatics Foundation is affiliated with BioPHP.	Cross-platform	Open Bioinformatics Foundation
Galaxy	Galaxy is a platform for scientific workflow, data integration and persistence of data and analysis and publication that seeks to make computational biology available to research scientists who do not have expertise with computer programming or system management. While it was originally designed for research in genomics, it is essentially domain agnostic and is now used as a general workflow management framework for bioinformatics.	Unix-like	Galaxy Community

geWorkbench	geWorkbench (genomics Workbench) is an open source, interactive genomic data analysis software framework. It is an application for the desktop written in the Java programming language. geWorkbench utilises an architectural component. There are more than 70 plug-ins available as of 2016, allowing for gene expression, sequence, and structure data visualisation and analysis. geWorkbench is the MAGNet Bioinformatics Platform, one of the eight National Centers for Biomedical Computing funded by the NIH Roadmap, the National Center for Multi-scale Analysis of Genomic and Cellular Networks (NIH Common Fund). As geWorkbench plugins, many systems and structure biology methods built by MAGNet investigators are available.	Linux, macOS, Windows	Columbia University
InterMine	InterMine is a data warehouse framework that is open source and licenced under LGPL 2.1. InterMine is used to build biological data databases which are accessible through sophisticated web query software. InterMine can be used from a single data collection to build databases or can combine various data sets. Several popular biological formats are supported, and there is a mechanism for the inclusion of other data. InterMine features a user-friendly web interface that can be quickly modified and operates 'out of the box.' InterMine allows various data sources to be conveniently combined into a single data warehouse. It has a sequence ontology-based core data model and supports multiple biological data formats, allowing sysadmins to customise the appropriate organisms or data files. With a web service API, clients in seven different languages, and an XML format to help import custom data, it's simple to expand the data model and add other data. As an active open-source community, InterMine maintains a developer mailing list and extensive developer and user documentation.	Cross-platform	University of Cambridge

LabKey Server	LabKey Server is a software suite available for combining, evaluating, and exchanging biomedical research data with scientists. The platform offers a secure data repository that enables a variety of data sources to query, monitor and collaborate on a web-based basis. On top of the standard platform, complex scientific technologies and workflows can be applied and utilise a pipeline of data processing.	Linux, macOS, Windows	LabKey Software Foundation

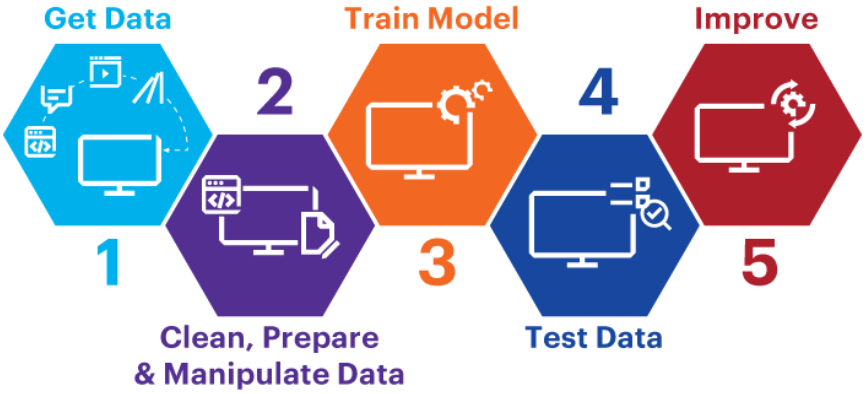
D. Methodology

The methodology in this project is the combination of two methodology. The first one is the machine learning or research methodology and the second one is the software or system development methodology.

A. Research Methodology

For the research part, we will use Python as the language and Google Colab that based on the Jupyter notebook as the IDE. The Python libraries that we use are Pandas, NumPy, SciPy, Matplotlib, Seaborn and RDKit. We get all the packages needed by installing Miniconda. Miniconda is the minimize version of Anaconda which is a distribution of Python for scientific computing, that simplify the package management and deployment. So, we do not need to install manually each package to import the library. Pandas is basically a library for data structures called dataframe and its operations including manipulating the numerical tables and time series. NumPy offers multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. SciPy builds on the NumPy array object and is part of the NumPy stack that contains modules for optimization, linear algebra, integration, interpolation, special functions, signal and image processing and other tasks common in science and engineering. Matplotlib is a plotting library for Python and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI. Seaborn is a data visualization library based on matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. Scikit learn or sklearn is a library that features various classification, regression and clustering algorithms that we will use for building the model. RDKit is a collection of cheminformatics libraries that we specifically need to use in this project.

There are five phases which are getting data, cleaning, preparing and manipulating data, training model, testing data and improving the model.



1. Get Data

We perform the data collection by getting the data from the ChEMBL Database. The ChEMBL Database is a database that contains curated bioactivity data of more than 2 million compounds. It is compiled from more than 76,000 documents, 1.2 million assays and the data spans 13,000 targets and 1,800 cells and 33,000 indications. [Data as of January 5, 2021; ChEMBL version 27].

First and foremost, we need to search the dataset that we want and select the most suitable one. Then we fetch or download the raw data. This step can be executed either directly at the website or manually in the IDE by installing the web service package.

2. Clean, Prepare and Manipulate Data

To clean the data, firstly we need to filter the specific properties from data that we want. Then we need to drop the rows that has missing value of important columns that need to be use later. After that, we delete the the duplicate rows that have the same value for certain columns that cannot be identical which might affect the data analysis later.

For data preparation, we can combine a few important columns into a single dataframe. Then based on the the information we get from the dataset, we can derive a new column to make the data more insightful. After that, we can manipulate and analyze the data based on the framework that we set.

3. Train Model

In this phase, we need to train our model based on the data that we prepared. We can use multiple models from the library such as Sklearn and compare its performance. The model needs to be accurate enough in order to be useful.

4. Test Data

Using the model that we train, we need to test the output data that we get from the model in order to see accuracy and whether it is bias or not.

5. Improve

Lastly, we need to constantly improve the prepared data, the way we see and analyze it, and most importantly the model itself. Improve its accuracy as much as possible and reduce the complexity as much as possible. Or maybe we can consider to change the algorithm.

B. Development Methodology

For the development methodology, we use AI Integrated Software Development Life Cycle (AI-SDLC). This methodology have two parts which are inner and outer cycles.

Inner Cycle

The inner cycle is basically a normal SDLC approach to software development which consist of five phases. Each of these phases can run at the same time but have a different finishing time.

1. Requirement: Collect the requirement of the system by the stakeholders which are our team ourselves. Understand the problem. Create a use case diagram.

2. Design: Design the prototype of the software using Adobe XD

3. Implementation: Coding the backend (machine learning part) and frontend (user interface) and deploy it using Streamlit

4. Testing: Testing the system as a user

5. Evaluation: Evaluate the system for improvement

Outer Cycle

The outer cycle is the first methodology that integrated into the development methodology. The phases also can run at the same with the inner cycle phases.

1. Contextualization: Understand the context of the software by getting, preparing and analyzing the data (from phase 1 and 2 of first methodology)

2. Model Development & Training: Develop and train the model (from phase 3)

3. Simulation and Testing: Test the model (from phase 4)

4. Continuous Integration and Delivery: Deliver, improve the model iteratively (from phase 5 + integration)

E. Progress

A. Research Progress

The current progress of this project is we analyze the bioactivity data of a target protein to get the insight. A drug target is a molecule in the body, usually a protein, that is associated with a particular disease process and that could be addressed by a drug to produce a desired therapeutic effect (D. Cavalla et al., 2017). Therefore it is called target protein. When the drug is consumed, its molecules will come into contact with the protein and induce a modulatory activity in order to activate the protein or to inhibit it. The dataset of a target protein consists of its bioactivity data when it interacts with a list of drug compounds. The compounds that fulfilled the certain criteria will have a chance to be a good drug candidate for the target protein and can be classified as the target protein inhibitor.

1. Get Data

Before we start, we install the Miniconda and RDKit packages that contain the python libraries that we are going to use. Then we install the ChEMBL web service package so that we can retrieve bioactivity data from the ChEMBL Database.

```
! wget https://repo.anaconda.com/miniconda/Miniconda3-py37_4.8.2-Linux-x86_64.sh
! chmod +x Miniconda3-py37_4.8.2-Linux-x86_64.sh
! bash ./Miniconda3-py37_4.8.2-Linux-x86_64.sh -b -f -p /usr/local
! conda install -c rdkit rdkit -y
import sys
sys.path.append('/usr/local/lib/python3.7/site-packages/')
```

```
! pip install chembl_webresource_client
```

Firstly, we import the necessary libraries.

```
import pandas as pd
from chembl_webresource_client.new_client import new_client
```

Then, we decide to search for acetylcholinesterase which is a target protein.

```
target = new_client.target
target_query = target.search('acetylcholinesterase')
targets = pd.DataFrame.from_dict(target_query)
targets
```

```
[3] target = new_client.target
    target_query = target.search('acetylcholinesterase')
    targets = pd.DataFrame.from_dict(target_query)
    targets
```

	cross_references	organism	pref_name	score	species_group_flag	target_chembl_id	target_components	target_type	tax_i
0	[{'xref_id': 'P22303', 'xref_name': None, 'xre...	Homo sapiens	Acetylcholinesterase	27.0	False	CHEMBL220	[{'accession': 'P22303', 'component_descriptio...	SINGLE PROTEIN	960
1	[]	Homo sapiens	Cholinesterases; ACHE & BCHE	27.0	False	CHEMBL2095233	[{'accession': 'P06276', 'component_descriptio...	SELECTIVITY GROUP	960
2	[]	Drosophila melanogaster	Acetylcholinesterase	17.0	False	CHEMBL2242744	[{'accession': 'P07140', 'component_descriptio...	SINGLE PROTEIN	722
3	[]	Nephotettix cincticeps	Ace-orthologous acetylcholinesterase	17.0	False	CHEMBL2366514	[{'accession': 'Q9NJH6', 'component_descriptio...	SINGLE PROTEIN	9440
4	[{'xref_id': 'P04058', 'xref_name': None, 'xre...	Torpedo californica	Acetylcholinesterase	15.0	False	CHEMBL4780	[{'accession': 'P04058', 'component_descriptio...	SINGLE PROTEIN	778
5	[{'xref_id': 'P21836', 'xref_name': None, 'xre...	Mus musculus	Acetylcholinesterase	15.0	False	CHEMBL3198	[{'accession': 'P21836', 'component_descriptio...	SINGLE PROTEIN	1009
6	[{'xref_id': 'P37136', 'xref_name': None, 'xre...	Rattus norvegicus	Acetylcholinesterase	15.0	False	CHEMBL3199	[{'accession': 'P37136', 'component_descriptio...	SINGLE PROTEIN	1011
7	[{'xref_id': 'O42275', 'xref_name': None, 'xre...	Electrophorus electricus	Acetylcholinesterase	15.0	False	CHEMBL4078	[{'accession': 'O42275', 'component_descriptio...	SINGLE PROTEIN	800
8	[{'xref_id': 'P23795', 'xref_name': None, 'xre...	Bos taurus	Acetylcholinesterase	15.0	False	CHEMBL4768	[{'accession': 'P23795', 'component_descriptio...	SINGLE PROTEIN	991
9	[]	Anopheles gambiae	Acetylcholinesterase	15.0	False	CHEMBL2046266	[{'accession': 'Q869C3', 'component_descriptio...	SINGLE PROTEIN	716
10	[]	Leptinotarsa decemlineata	Acetylcholinesterase	15.0	False	CHEMBL2366490	[{'accession': 'Q27677', 'component_descriptio...	SINGLE PROTEIN	753
11	[]	Gallus gallus	Acetylcholinesterase	15.0	False	CHEMBL3227914	[{'accession': 'P36196',	SINGLE	903

Here see that there are 22 search result of acetylcholinesterase of different organism, so we select and retrieve bioactivity data for Human Acetylcholinesterase, which is the first entry. Then, we assign the first entry (index 0) to the selected_target variable and retrieve only bioactivity data for Human Acetylcholinesterase (CHEMBL220) that are reported as IC50. We define a particular a standard type here so that our dataset become more uniform and won't have a mixture of different bioactivity units.

```
selected_target = targets.target_chembl_id[0]
selected_target
activity = new_client.activity
res = activity.filter(target_chembl_id=selected_target).filter(standard
_type="IC50")
df = pd.DataFrame.from_dict(res)
df
```

```
[4] selected_target = targets.target_chembl_id[0]
    selected_target
```

```
'CHEMBL220'
```

```
[5] activity = new_client.activity
    res = activity.filter(target_chembl_id=selected_target).filter(standard_type="IC50")
```

```
[6] df = pd.DataFrame.from_dict(res)
```

```
[7] df
```

	activity_comment	activity_id	activity_properties	assay_chembl_id	assay_description	assay_ty
0	None	33969	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	
1	None	37563	[]	CHEMBL643384	Inhibitory concentration	

```
[7] df
```

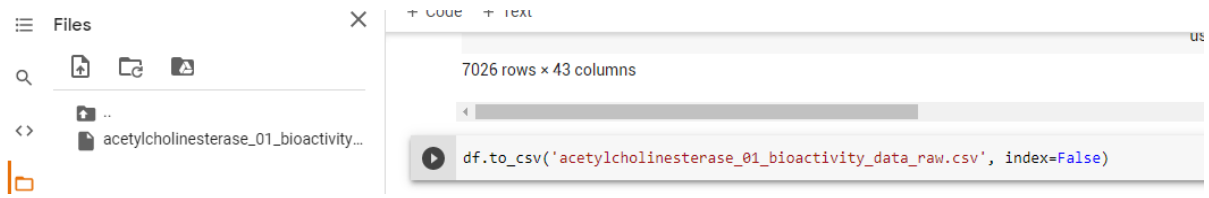
standard_type	standard_units	standard_upper_value	standard_value	target_chembl_id	target_organism	target_pref_name
IC50	nM	None	750.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	100.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	50000.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	300.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	800.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
...	...	...	...	...	...	...
IC50	nM	None	3810.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	3460.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	2780.0	CHEMBL220	Homo sapiens	Acetylcholinesterase
IC50	nM	None	450.0	CHEMBL220	Homo sapiens	Acetylcholinesterase

In this data frame there are 7026 rows and 43 columns of IC50 standard. IC50, which is half-maximal inhibitory concentration, indicates how much drug is needed to inhibit a biological process by half, thus providing a measure of potency of an drug (Aykul et al.,

2016). The standard value above is the value of IC50. The lower the value, the better the potency of the drug.

Now, we save the raw bioactivity data to a CSV file bioactivity_data.csv.

```
df.to_csv('acetylcholinesterase_01_bioactivity_data_raw.csv', index=False)
```



At the left side, we have the data in CSV format that can be downloaded.

2. Clean, Prepare and Manipulate Data

2.1 Data Cleaning

For the cleaning part, we drop any compounds that has missing value for the standard_value and canonical_smiles. Canonical SMILES (Simplified Molecular Input Line Entry Specification) is a line notation that represent the molecular structure which written in canonical format.

```
df2 = df[df.standard_value.notna() & df.canonical_smiles.notna()]
df2
```

```
df2 = df[df.standard_value.notna() & df.canonical_smiles.notna()]
df2
```

	activity_comment	activity_id	activity_properties	assay_chembl_id	assay_description	assay_type	bao_endpoint	bao_format	bao_label	canonical_smiles	data_validity_comment	dat
0	None	33969	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	None	
1	None	37563	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	None	
2	None	37565	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	None	
3	None	38902	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	None	
4	None	41170	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	None	
...	...	...	...	...	...	...	...	...	...	...	...	

Data frame above have 5835 rows.

Then we check the for total unique compound.

```
len(df2.canonical_smiles.unique())
```

```
len(df2.canonical_smiles.unique())
```

```
4695
```

There is 4695 unique rows.

Thus, we drop the replicates rows.

```
df2_nr = df2.drop_duplicates(['canonical_smiles'])
df2_nr
```

```
df2_nr = df2.drop_duplicates(['canonical_smiles'])
df2_nr
```

	activity_comment	activity_id	activity_properties	assay_chembl_id	assay_description	assay_type	bao_endpoint	bao_format	bao_label	canonical_smiles	data
0	None	33969	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	
1	None	37563	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	
2	None	37565	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	
3	None	38902	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	
4	None	41170	[]	CHEMBL643384	Inhibitory concentration against acetylcholine...	B	BAO_0000190	BAO_0000357	single protein format	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	
...	...	...	...	...	...	...	...	...	...	...	...

2.2 Data Preparation

Now we combine the 3 columns (molecule_chembl_id,canonical_smiles,standard_value) into a data frame.

```
selection = ['molecule_chembl_id','canonical_smiles','standard_value']
df3 = df2_nr[selection]
df3
```

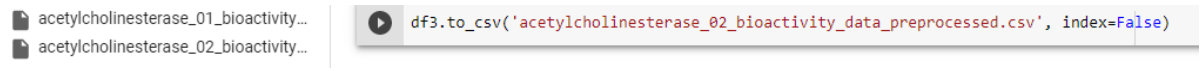
```
selection = ['molecule_chembl_id','canonical_smiles','standard_value']
df3 = df2_nr[selection]
df3
```

	molecule_chembl_id	canonical_smiles	standard_value
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	750.0
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	100.0
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	50000.0
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	300.0
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	800.0
...	...	...	...

4695 rows × 3 columns

```
df3.to_csv('_02_bioactivity_data_preprocessed.csv', index=False)
```

Then we saves the data frame to CSV file. We can see at the left side the second file that are generated.



After that, we label the compounds as either being active, inactive or intermediate. The bioactivity data (standard_value) is the IC50 values in nM (nanomolar) unit. Remember that the IC50 value is better when it is lower. It is better because the compound needs lower volume in order to function. So, we divide the compounds into three bioactivity classes. The compounds that have values of less than 1000 nM will be considered to be active while those greater than 10,000 nM will be considered to be inactive. As for those values in between 1,000 and 10,000 nM will be referred to as intermediate.

```
bioactivity_threshold = []
for i in df3.standard_value:
  if float(i) >= 10000:
    bioactivity_threshold.append("inactive")
  elif float(i) <= 1000:
    bioactivity_threshold.append("active")
  else:
    bioactivity_threshold.append("intermediate")
bioactivity_class = pd.Series(bioactivity_threshold, name='class')
df4 = pd.concat([df3, bioactivity_class], axis=1)
df4
```

```
[13] df3.to_csv('acetylcholinesterase_02_bioactivity_data_preprocessed.csv', index=False)
```

```
[14] bioactivity_threshold = []
     for i in df3.standard_value:
       if float(i) >= 10000:
         bioactivity_threshold.append("inactive")
       elif float(i) <= 1000:
         bioactivity_threshold.append("active")
       else:
         bioactivity_threshold.append("intermediate")
```

```
[15] bioactivity_class = pd.Series(bioactivity_threshold, name='class')
     df4 = pd.concat([df3, bioactivity_class], axis=1)
     df4
```

	molecule_chembl_id	canonical_smiles	standard_value	class
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	750.0	active
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	100.0	active
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	50000.0	inactive
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	300.0	active
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	800.0	active
...	...	...	...	...
4690	CHEMBL4293155	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccccc3Cl)CC...	2440.0	intermediate
4691	CHEMBL4282558	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc(Cl)c3)...	2540.0	intermediate
4692	CHEMBL4281727	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(Br)cc3)...	3810.0	intermediate
4693	CHEMBL4292349	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc([N+](=...	3460.0	intermediate
4694	CHEMBL4278260	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(C#N)cc3...	2780.0	intermediate

4695 rows × 4 columns

Then, we save the dataframe to CSV file

```
df4.to_csv('acetylcholinesterase_03_bioactivity_data_curated.csv', index=False)
```

2.3 Data Manipulation

For data manipulation and analysis, firstly we calculate the Lipinski descriptors. This descriptor come from Christopher Lipinski, who is a scientist at Pfizer that came up with a set of rule-of-thumb for evaluating the druglikeness of compounds. Druglikeness metrics is based on the Absorption, Distribution, Metabolism and Excretion (ADME) that is also known as the pharmacokinetic profile. Lipinski analyzed all orally active FDA-approved drugs in the formulation of what is to be known as the Rule-of-Five or Lipinski's Rule (Lipinski, Lombardo, Dominy, & Feeney, 2001). The Lipinski's Rule stated the following:

- Molecular weight < 500 Dalton
- Octanol-water partition coefficient (LogP) < 5
- Hydrogen bond donors < 5
- Hydrogen bond acceptors < 10

So, the compound properties need to be aligned with these rules in order to be a good candidate of drug.

To start, we import the related libraries.

```
import numpy as np
from rdkit import Chem
from rdkit.Chem import Descriptors, Lipinski
```

Then we create a custom function for the Lipinski descriptors.

```
def lipinski(smiles, verbose=False):

    moldata= []
    for elem in smiles:
        mol=Chem.MolFromSmiles(elem)
        moldata.append(mol)

    baseData= np.arange(1,1)
    i=0
    for mol in moldata:

        desc_MolWt = Descriptors.MolWt(mol)
        desc_MolLogP = Descriptors.MolLogP(mol)
        desc_NumHDonors = Lipinski.NumHDonors(mol)
        desc_NumHAcceptors = Lipinski.NumHAcceptors(mol)

        row = np.array([desc_MolWt,
                        desc_MolLogP,
                        desc_NumHDonors,
                        desc_NumHAcceptors])

        if(i==0):
            baseData=row
        else:
            baseData=np.vstack([baseData, row])
        i=i+1

    columnNames=["MW","LogP","NumHDonors","NumHAcceptors"]
    descriptors = pd.DataFrame(data=baseData,columns=columnNames)

    return descriptors
```

```
import numpy as np
from rdkit import Chem
from rdkit.Chem import Descriptors, Lipinski
```

```
def lipinski(smiles, verbose=False):

    moldata= []
    for elem in smiles:
        mol=Chem.MolFromSmiles(elem)
        moldata.append(mol)

    baseData= np.arange(1,1)
    i=0
    for mol in moldata:

        desc_MolWt = Descriptors.MolWt(mol)
        desc_MolLogP = Descriptors.MolLogP(mol)
        desc_NumHDonors = Lipinski.NumHDonors(mol)
        desc_NumHAcceptors = Lipinski.NumHAcceptors(mol)

        row = np.array([desc_MolWt,
                        desc_MolLogP,
                        desc_NumHDonors,
                        desc_NumHAcceptors])

        if(i==0):
            baseData=row
        else:
            baseData=np.vstack([baseData, row])
        i=i+1

    columnNames=["MW","LogP","NumHDonors","NumHAcceptors"]
    descriptors = pd.DataFrame(data=baseData,columns=columnNames)

    return descriptors
```

Then we run the function and put it into a dataframe name df_lipinski.

```
df_lipinski = lipinski(df4.canonical_smiles)
df_lipinski
```

```
df_lipinski = lipinski(df4.canonical_smiles)
df_lipinski
```

	MW	LogP	NumHDonors	NumHAcceptors
0	312.325	2.80320	0.0	6.0
1	376.913	4.55460	0.0	5.0
2	426.851	5.35740	0.0	5.0
3	404.845	4.70690	0.0	5.0
4	346.334	3.09530	0.0	6.0
...	...	...	...	...
4690	511.150	7.07230	2.0	3.0
4691	511.150	7.07230	2.0	3.0
4692	555.601	7.18140	2.0	3.0
4693	521.702	6.32710	2.0	5.0
4694	501.715	6.29058	2.0	4.0

4695 rows × 4 columns

Then we combined data frame df4 and df_lipinski into a single data frame.

```
df_combined = pd.concat([df,df_lipinski], axis=1)
df_combined
```

```
df_combined = pd.concat([df,df_lipinski], axis=1)
```

```
df_combined
```

	molecule_chembl_id	canonical_smiles	standard_value	class	MW	LogP	NumHDonors	NumHAcceptors
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	750.0	active	312.325	2.80320	0.0	6.0
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	100.0	active	376.913	4.55460	0.0	5.0
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	50000.0	inactive	426.851	5.35740	0.0	5.0
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	300.0	active	404.845	4.70690	0.0	5.0
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	800.0	active	346.334	3.09530	0.0	6.0
...	...	...	...	...	...	...	...	...
4690	CHEMBL4293155	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccccc3Cl)CC...	2440.0	intermediate	511.150	7.07230	2.0	3.0
4691	CHEMBL4282558	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc(Cl)c3)...	2540.0	intermediate	511.150	7.07230	2.0	3.0
4692	CHEMBL4281727	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(Br)cc3)...	3810.0	intermediate	555.601	7.18140	2.0	3.0
4693	CHEMBL4292349	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc([N+](=...	3460.0	intermediate	521.702	6.32710	2.0	5.0
4694	CHEMBL4278260	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(C#N)cc3...	2780.0	intermediate	501.715	6.29058	2.0	4.0

4695 rows × 8 columns

Now, to allow IC50 data to be more uniformly distributed, we convert IC50 to pIC50 which is negative logarithmic scale of IC50, calculated as -log10(IC50). The custom function of pIC50() will accept a data frame as input and will:

- Take the IC50 values from the standard_value column and converts it from nM to M by multiplying the value by 10−9
- Take the molar value and apply -log10
- Delete the standard_value column and create a new pIC50 column

```
def pIC50(input):
    pIC50 = []

    for i in input['standard_value_norm']:
        molar = i*(10**-9)
        pIC50.append(-np.log10(molar))

    input['pIC50'] = pIC50
    x = input.drop('standard_value_norm', 1)

    return x
```

When the values of IC50 are greater than 100,000,000, the resulting pIC50 value will be negative. It will make the interpretation difficult.

```
df_combined.standard_value.describe()
```

```
count    4.695000e+03
mean     3.210931e+12
std      1.189580e+14
min      6.000000e-03
25%      1.250000e+02
50%      2.246000e+03
75%      1.753000e+04
max      5.888437e+15
Name: standard_value, dtype: float64
```

```
-np.log10( (10**-9)* 100000000 )
```

```
1.0
```

```
-np.log10( (10**-9)* 10000000000 )
```

```
-1.0
```

```
-np.log10( (10**-9)* 5.888437e+15 )
```

```
-6.770000032926942
```

Here we can see that the maximum value of IC50 (standard_value) become negative after converting to pIC50.

In order to prevent that, the IC50 values that are greater than 100,000,000 will be fixed at 100,000,000. So we create a function to solve this.

```
def norm_value(input):
    norm = []

    for i in input['standard_value']:
        if i > 100000000:
          i = 100000000
        norm.append(i)

    input['standard_value_norm'] = norm
    x = input.drop('standard_value', 1)

    return x
```

Then we apply the norm_value() function so that the values in the standard_value column is normalized.

```
df_norm = norm_value(df_combined)
df_norm
```

```
df_norm = norm_value(df_combined)
df_norm
```

	molecule_chembl_id	canonical_smiles	class	MW	LogP	NumHDonors	NumHAcceptors	standard_value_norm
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	active	312.325	2.80320	0.0	6.0	750.0
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	active	376.913	4.55460	0.0	5.0	100.0
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	inactive	426.851	5.35740	0.0	5.0	50000.0
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	active	404.845	4.70690	0.0	5.0	300.0
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	active	346.334	3.09530	0.0	6.0	800.0
...	...	...	...	...	...	...	...	...
4690	CHEMBL4293155	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccccc3Cl)CC...	intermediate	511.150	7.07230	2.0	3.0	2440.0
4691	CHEMBL4282558	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc(Cl)c3)...	intermediate	511.150	7.07230	2.0	3.0	2540.0
4692	CHEMBL4281727	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(Br)cc3)...	intermediate	555.601	7.18140	2.0	3.0	3810.0
4693	CHEMBL4292349	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc([N+](=...	intermediate	521.702	6.32710	2.0	5.0	3460.0
4694	CHEMBL4278260	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(C#N)cc3...	intermediate	501.715	6.29058	2.0	4.0	2780.0

4695 rows × 8 columns

```
df_norm.standard_value_norm.describe()
```

```
count    4.695000e+03
mean     3.276494e+05
std      4.718369e+06
min      6.000000e-03
25%      1.250000e+02
50%      2.246000e+03
75%      1.753000e+04
max      1.000000e+08
Name: standard_value_norm, dtype: float64
```

Here we see that the maximum standard value of IC50 become 100,000,000.

Then we implement the pIC50 function to the df_norm and store it into a new data frame

```
df_final = pIC50(df_norm)
df_final
```

```
df_final = pIC50(df_norm)
df_final
```

	molecule_chembl_id	canonical_smiles	class	MW	LogP	NumHDonors	NumHAcceptors	pIC50
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	active	312.325	2.80320	0.0	6.0	6.124939
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	active	376.913	4.55460	0.0	5.0	7.000000
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	inactive	426.851	5.35740	0.0	5.0	4.301030
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	active	404.845	4.70690	0.0	5.0	6.522879
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	active	346.334	3.09530	0.0	6.0	6.096910
...	...	...	...	...	...	...	...	...
4690	CHEMBL4293155	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccccc3Cl)CC...	intermediate	511.150	7.07230	2.0	3.0	5.612610
4691	CHEMBL4282558	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc(Cl)c3)...	intermediate	511.150	7.07230	2.0	3.0	5.595166
4692	CHEMBL4281727	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(Br)cc3)...	intermediate	555.601	7.18140	2.0	3.0	5.419075
4693	CHEMBL4292349	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3cccc([N+](=...	intermediate	521.702	6.32710	2.0	5.0	5.460924
4694	CHEMBL4278260	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(C#N)cc3...	intermediate	501.715	6.29058	2.0	4.0	5.555955

4695 rows × 8 columns

Now we can see that the standard_value column (IC50) has been replaced by pIC50. After that we want to see the range of pIC50 values.

```
df_final.pIC50.describe()
```

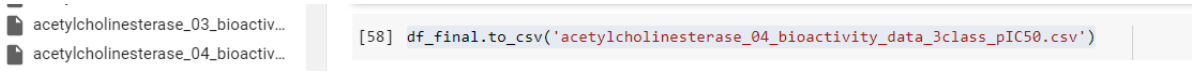
```
df_final.pIC50.describe()
```

```
count    4695.000000
mean        5.820864
std         1.552813
min         1.000000
25%         4.756219
50%         5.648590
75%         6.903090
max        11.221849
Name: pIC50, dtype: float64
```

The minimum value of pIC50 is 1.0000 and maximum 11.2218. This fix the problem of IC50 unit that have ununiform distributed values and have a very large range.

Then we save df_final into csv file.

```
df_final.to_csv('acetylcholinesterase_04_bioactivity_data_3class_pIC50.csv')
```



Next we want to do simple comparison between two bioactivity classes, therefore we create a new data frame by removing the intermediate class from our data set.

```
df_2class = df_final[df_final['class'] != 'intermediate']
df_2class
```

```
df_2class = df_final[df_final['class'] != 'intermediate']
df_2class
```

	molecule_chembl_id	canonical_smiles	class	MW	LogP	NumHDonors	NumHAcceptors	pIC50
0	CHEMBL133897	CCOc1nn(-c2cccc(OCc3ccccc3)c2)c(=O)o1	active	312.325	2.8032	0.0	6.0	6.124939
1	CHEMBL336398	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC1CC1	active	376.913	4.5546	0.0	5.0	7.000000
2	CHEMBL131588	CN(C(=O)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F)c1ccccc1	inactive	426.851	5.3574	0.0	5.0	4.301030
3	CHEMBL130628	O=C(N1CCCCC1)n1nc(-c2ccc(Cl)cc2)nc1SCC(F)(F)F	active	404.845	4.7069	0.0	5.0	6.522879
4	CHEMBL130478	CSc1nc(-c2ccc(OC(F)(F)F)cc2)nn1C(=O)N(C)C	active	346.334	3.0953	0.0	6.0	6.096910
...	...	...	...	...	...	...	...	...
4675	CHEMBL4284261	CCN(C)Cc1cc(N)ccc1O.Cl.Cl	inactive	180.251	1.4261	2.0	3.0	3.015428
4676	CHEMBL4276921	CN(C)Cc1cc(N)ccc1O.Cl.Cl	inactive	166.224	1.0360	2.0	3.0	2.813467
4677	CHEMBL4292574	CNCc1cc(N)ccc1O.Cl.Cl	inactive	152.197	0.6938	3.0	3.0	3.476904
4685	CHEMBL4292766	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccccc3F)CC2...	active	494.695	6.5580	2.0	3.0	6.124939
4687	CHEMBL4284475	CC(C)(C)c1cc(/C=C/C(=O)NCCC2CCN(Cc3ccc(F)cc3)C...	active	494.695	6.5580	2.0	3.0	6.008774

3549 rows × 8 columns

Then we save it.

```
df_2class.to_csv('acetylcholinesterase_05_bioactivity_data_2class_pIC50.csv')
```

Now we will do the chemical space analysis using the Lipinski descriptors. Chemical space is a term in cheminformatics that refer to space spanned by the molecules and chemical compounds based on a given set of construction principles and boundary conditions to see the general overview of potential pharmacologically active molecules (Rudling et al., 2017).

To start, we import the related libraries.

```
import seaborn as sns
sns.set(style='ticks')
import matplotlib.pyplot as plt
```

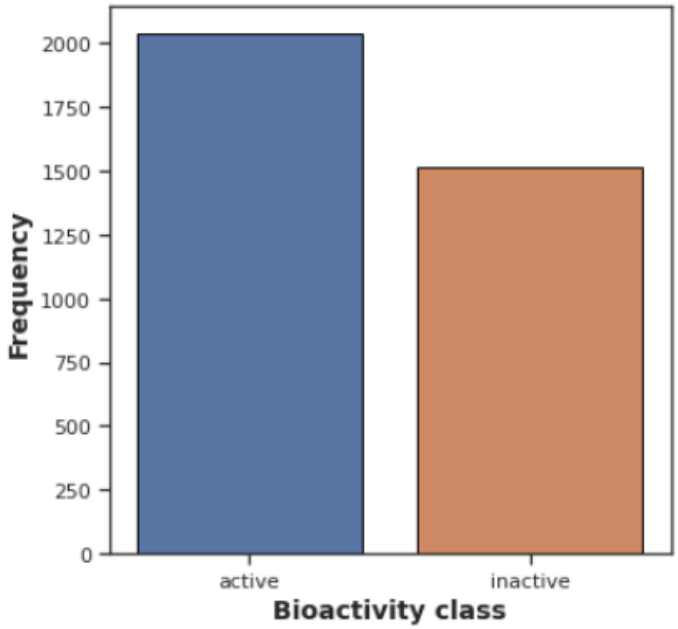
Firstly, we create a simple frequency plot of the 2 bioactivity classes. Then we save it into pdf file.

```
plt.figure(figsize=(5.5, 5.5))

sns.countplot(x='class', data=df_2class, edgecolor='black')

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('Frequency', fontsize=14, fontweight='bold')

plt.savefig('plot_bioactivity_class.pdf')
```



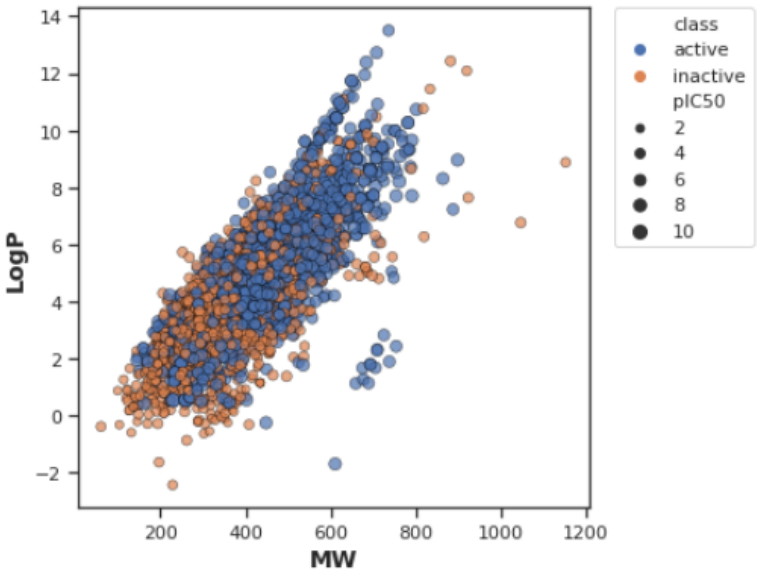
Here we see that the active molecules that react with acetylcholinesterase are more frequent than the inactive ones.

Secondly, we create a scatter plot of MW (Molecular Weight) versus LogP (solubility of molecule)

```
plt.figure(figsize=(5.5, 5.5))

sns.scatterplot(x='MW', y='LogP', data=df_2class, hue='class', size='pI
C50', edgecolor='black', alpha=0.7)

plt.xlabel('MW', fontsize=14, fontweight='bold')
plt.ylabel('LogP', fontsize=14, fontweight='bold')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
plt.savefig('plot_MW_vs_LogP.pdf')
```



From the scatter plot MW vs LogP above, it can be seen that the 2 bioactivity classes span almost similar chemical spaces, but the active class is a bit higher in term of LogP(solubility).

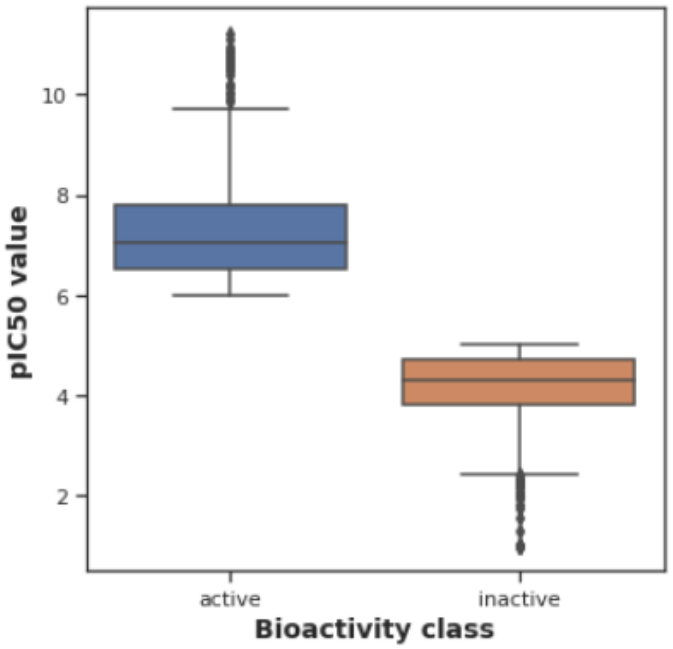
After that, we create a box plot to see the relationship of bioactivity class and pIC50 value.

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'class', y = 'pIC50', data = df_2class)

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('pIC50 value', fontsize=14, fontweight='bold')

plt.savefig('plot_ic50.pdf')
```



The box plot show the distribution of active and inactive class and it is expected as we use threshold to define active and inactive. We can simply say that the molecules that have pIC50 value above 6 are active and below 6 are inactive.

Now we apply statistical analysis Mann-Whitney U Test in order to look at the difference between the two bioactivity classes. We use this to test the statistical significance of the difference whether they are different or not. So first we define the function.

```
def mannwhitney(descriptor, verbose=False):

  from numpy.random import seed
  from numpy.random import randn
  from scipy.stats import mannwhitneyu

  seed(1)

  selection = [descriptor, 'class']
  df = df_2class[selection]
  active = df[df['class'] == 'active']
  active = active[descriptor]
```

```
  selection = [descriptor, 'class']
  df = df_2class[selection]
  inactive = df[df['class'] == 'inactive']
  inactive = inactive[descriptor]

  stat, p = mannwhitneyu(active, inactive)

  alpha = 0.05
  if p > alpha:
    interpretation = 'Same distribution (fail to reject H0)'
  else:
    interpretation = 'Different distribution (reject H0)'

  results = pd.DataFrame({'Descriptor':descriptor,
                          'Statistics':stat,
                          'p':p,
                          'alpha':alpha,
                          'Interpretation':interpretation}, index=[0])
  filename = 'mannwhitneyu_' + descriptor + '.csv'
  results.to_csv(filename)

  return results
```

Then, we apply this function to pIC50

```
mannwhitney('pIC50')
```

```
mannwhitney('pIC50')
```

	Descriptor	Statistics	p	alpha	Interpretation
0	pIC50	0.0	0.0	0.05	Different distribution (reject H0)

The test above compares the active class and the inactive class to see whether there is a statistical significance for the pIC50 variable. So based on this analysis the p-value is 0.0, therefore it reject the null hypothesis as it have different distribution of active and inactive.

After that we create the box plot and apply statistical analysis (Mann-Whitney U Test) for the other four Lipinski's descriptors as well.

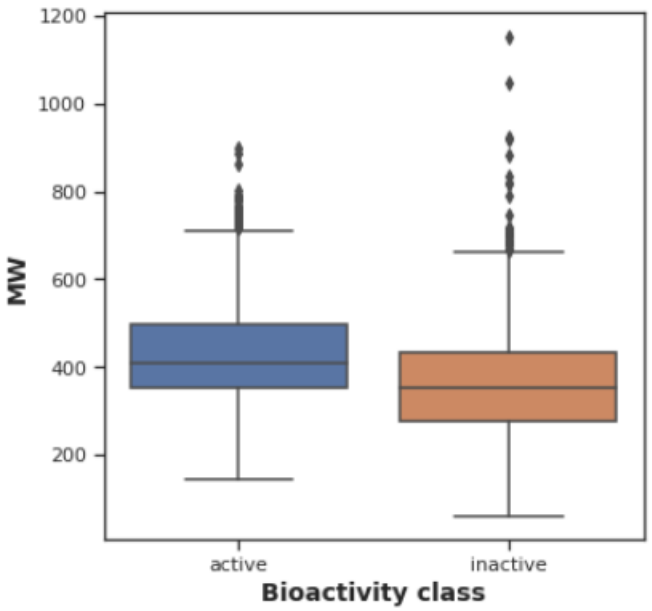
MW (Molecular Weight)

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'class', y = 'MW', data = df_2class)
```

```
plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('MW', fontsize=14, fontweight='bold')

plt.savefig('plot_MW.pdf')
```



We can see that the overall active class have a slightly higher MW than the inactive class.

```
mannwhitney('MW')
```

```
mannwhitney('MW')
```

	Descriptor	Statistics	p	alpha	Interpretation
0	MW	1058779.0	2.072256e-57	0.05	Different distribution (reject H0)

The p-value is extremely low, therefore it rejects the null hypothesis and we can say that it have a different distribution.

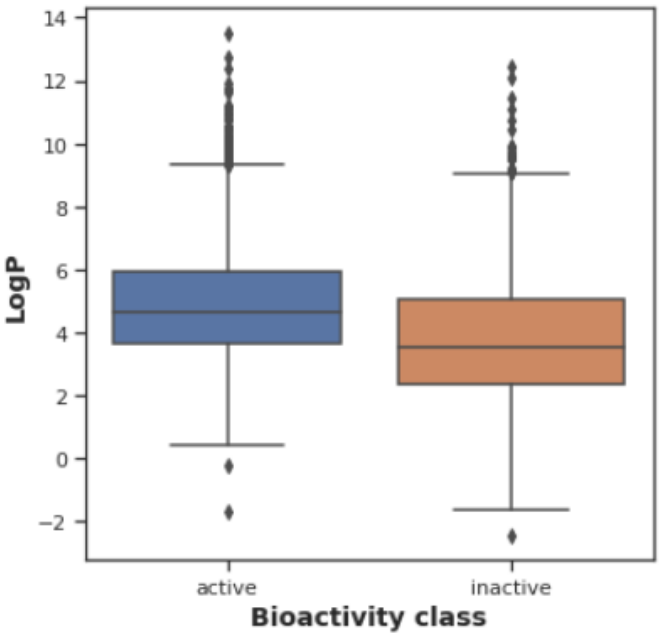
LogP (Octanol-water Partition Coefficient/Molecule Solubility)

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'class', y = 'LogP', data = df_2class)

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('LogP', fontsize=14, fontweight='bold')
```

```
plt.savefig('plot_LogP.pdf')
```



From the box plot, we see that active class have a bit higher LogP value than the inactive class.

```
mannwhitney('LogP')
```

```
mannwhitney('LogP')
```

	Descriptor	Statistics	p	alpha	Interpretation
0	LogP	1041900.0	2.318667e-61	0.05	Different distribution (reject H0)

The p-value of LogP is also very low, therefore we reject the null hypothesis.

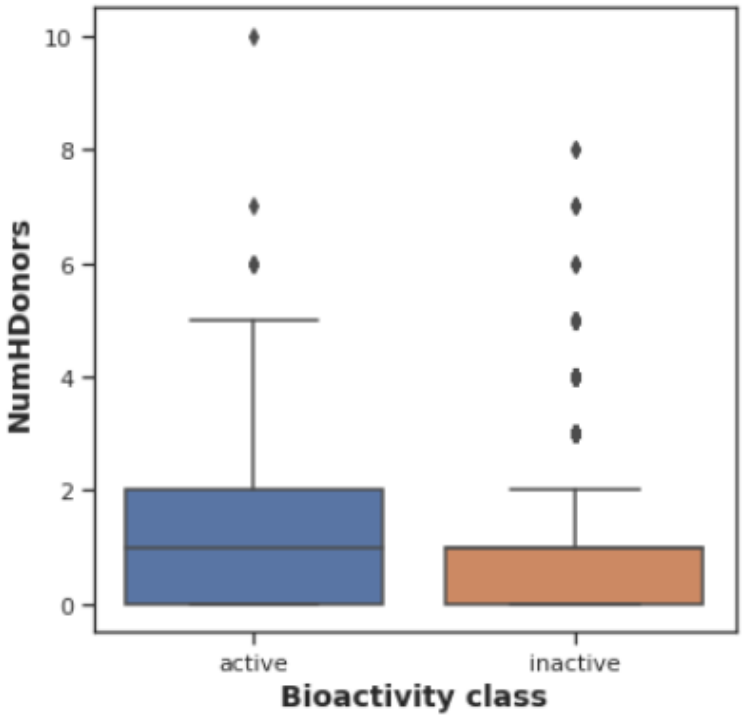
NumHDonors (Number of Hydrogen Bond Donors)

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'class', y = 'NumHDonors', data = df_2class)

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('NumHDonors', fontsize=14, fontweight='bold')
```

```
plt.savefig('plot_NumHDonors.pdf')
```



The NumHDonors of active class is between 0 to 2 while inactive class only 0 to 1.

```
mannwhitney('NumHDonors')
```

```
mannwhitney('NumHDonors')
```

	Descriptor	Statistics	p	alpha	Interpretation
0	NumHDonors	1361005.5	2.520096e-10	0.05	Different distribution (reject H0)

For NumHDonors, it rejects the H0 as p-value is very low.

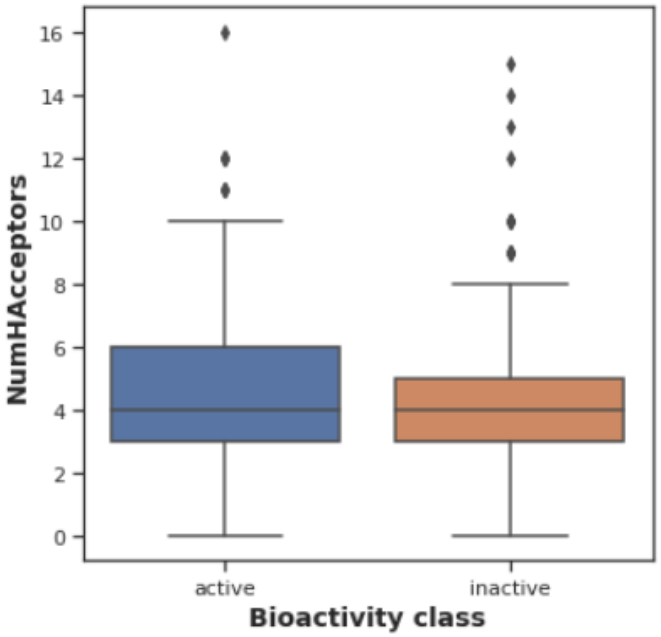
NumHAcceptors (Number of Hydrogen Bond Acceptors)

```
plt.figure(figsize=(5.5, 5.5))

sns.boxplot(x = 'class', y = 'NumHAcceptors', data = df_2class)

plt.xlabel('Bioactivity class', fontsize=14, fontweight='bold')
plt.ylabel('NumHAcceptors', fontsize=14, fontweight='bold')

plt.savefig('plot_NumHAcceptors.pdf')
```



The active class have range of NumHAcceptors near 3 to 6 while for inactive class between 3 to 5.

```
mannwhitney('NumHAcceptors')
```

```
mannwhitney('NumHAcceptors')
```

	Descriptor	Statistics	p	alpha	Interpretation
0	NumHAcceptors	1407572.5	0.000004	0.05	Different distribution (reject H0)

The p-value for NumHAcceptors is still relatively low, so it rejects the null hyphothesis.

In conclusion, all of the 4 Lipinski's descriptors (MW, LogP, NumHDonors, NumHAcceptors) exhibited statistically significant difference between the actives and inactives.

Finally, we zip all the outputs for easier download

```
! zip -r results.zip . -i *.csv *.pdf
```

```
! zip -r results.zip . -i *.csv *.pdf
```

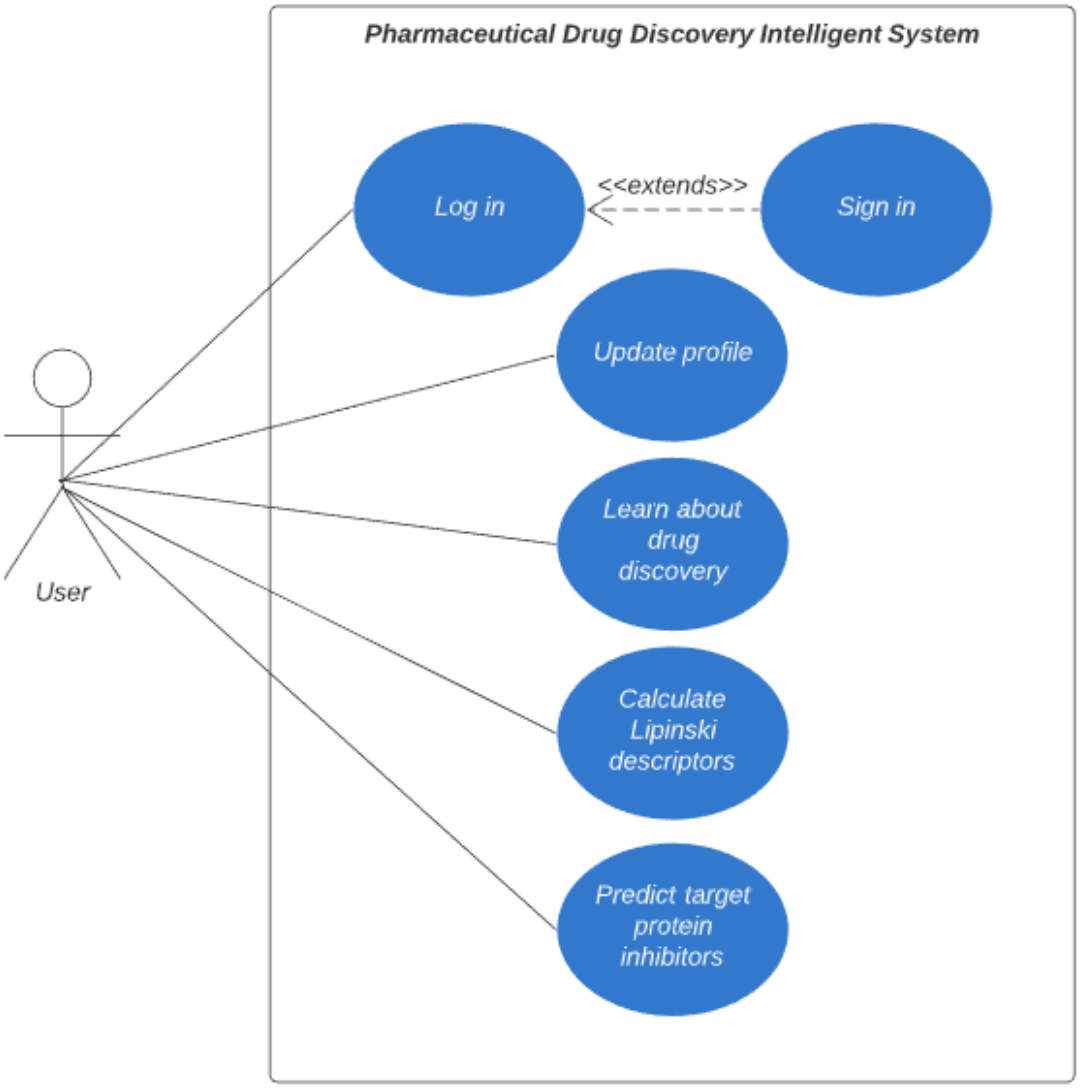
```
  adding: plot_MW_vs_LogP.pdf (deflated 2%)
  adding: plot_MW.pdf (deflated 38%)
  adding: mannwhitneyu_pIC50.csv (deflated 14%)
  adding: plot_bioactivity_class.pdf (deflated 39%)
  adding: mannwhitneyu_MW.csv (deflated 10%)
  adding: acetylcholinesterase_04_bioactivity_data_3class_pIC50.csv (deflated 76%)
  adding: plot_NumHAcceptors.pdf (deflated 38%)
  adding: mannwhitneyu_NumHDonors.csv (deflated 11%)
  adding: acetylcholinesterase_01_bioactivity_data_raw.csv (deflated 90%)
  adding: plot_NumHDonors.pdf (deflated 38%)
  adding: acetylcholinesterase_03_bioactivity_data_curated.csv (deflated 82%)
  adding: acetylcholinesterase_02_bioactivity_data_preprocessed.csv (deflated 81%)
  adding: mannwhitneyu_LogP.csv (deflated 7%)
  adding: plot_LogP.pdf (deflated 37%)
  adding: plot_ic50.pdf (deflated 37%)
  adding: acetylcholinesterase_05_bioactivity_data_2class_pIC50.csv (deflated 76%)
  adding: mannwhitneyu_NumHAcceptors.csv (deflated 11%)
```

To summarize, in these steps above, we manage to do the data preparation and analysis to get the general overview of the bioactivity classes of the selected target protein by using the Lipinski descriptor. For the future works, we will further analyze the unique molecular fingerprint of the active class in order to see the pattern of the inhibitors. Then, we will build a regression model to predict the target protein inhibitors using the random forest algorithm. On the other hand, we will try to use other algorithm to compare their accuracy. And if the model has high accuracy then we can look at the inhibitors molecular structure pattern, thus we know the accurate category of inhibitors for specific target protein to be the potential drug candidate.

B. Development Progress

1. Requirement

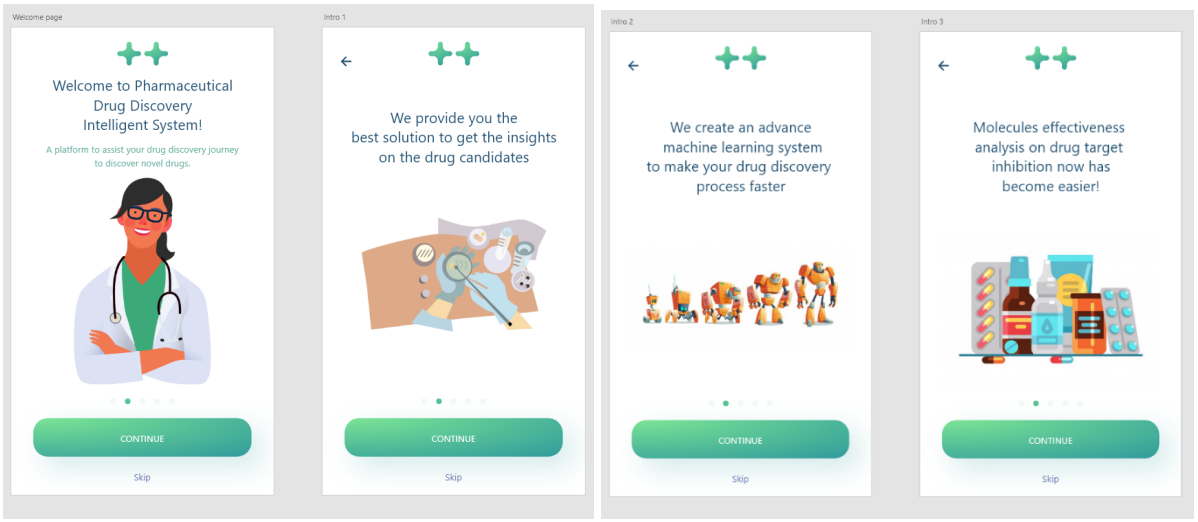
Use case diagram



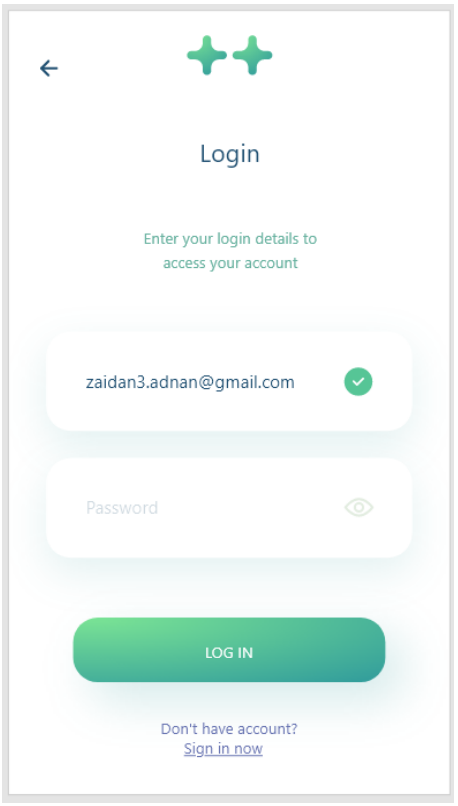
In this use case diagram, the user is the only actor. The use cases are log in with sign in as extend relationship. Then the main use cases are calculate Lipinski descriptors and predict target protein inhibitors and the other use case are update profile and learn about drug discovery.

2. Design

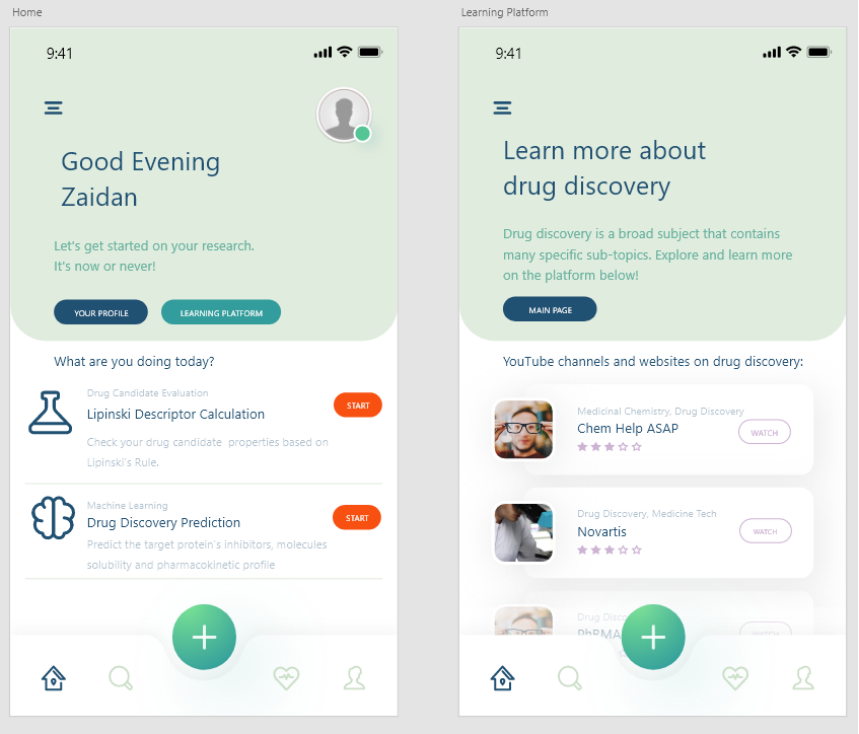
Prototype



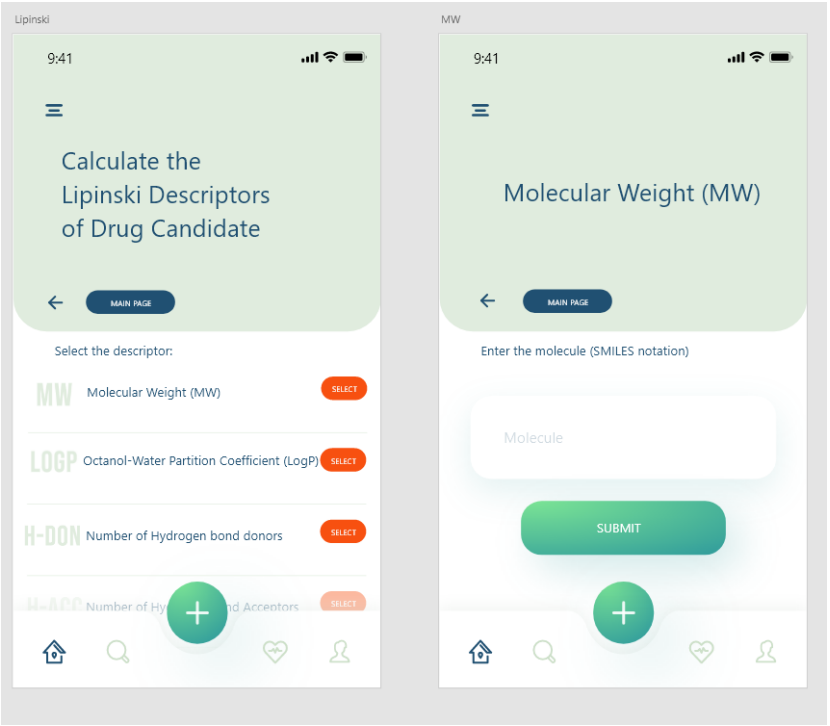
Firstly, when the users open the application, they will see we a welcoming page of the app. The users can either choose to continue or skip the app introduction by clicking button at the bottom.



To enter the app, the users need to log in by entering the email and password. If they do not have account yet, they can click the sign in hyperlink at the bottom.



After the successful login, it will direct to main page. At the centre there are button for users to update their profile and button to go to learn more about drug discovery. At the bottom are the main functions of the app which are calculation of Lipinski descriptor and prediction on target proteins inhibitor.



In the function of Lipinski descriptors calculation, the users can choose one of the descriptors to be calculate. After the user select, the users need to enter the molecule in SMILES notation to get the output. Then the result will pop up and inform whether the molecules satisfied the Lipinski rule or not.

References

Ahmed, S., & Hossain, M. F. (2019). The impact of robotics in the growth and economic development. The Business and Management Review, 10(5), 204–215.

Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. JAMA - Journal of the American Medical Association, 319(13), 1317–1318. https://doi.org/10.1001/jama.2017.18391

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195–202. https://doi.org/10.1038/nature23474

Bogdan, R., Salmeron, B. J., Carey, C. E., Agrawal, A., Calhoun, V. D., Garavan, H., Hariri, A. R., Heinz, A., Hill, M. N., & Holmes, A. (2018). Imaging genetics and genomics in psychiatry: a critical review of progress and potential. Biological Psychiatry, 82(3), 165–175. https://doi.org/10.1016/j.biopsych.2016.12.030.Imaging

Bughin, Jacques ;Hazan, Eric; Manyika, James; Woetzel, J. (2017). Artificial Intelligence – The next digital frontier? McKinsey Global Institute Discussion Paper, 2017(4), 14–17. www.mckinsey.com/mgi.

Bustos, A., & Pertusa, A. (2018). Learning eligibility in cancer clinical trials using deep neural networks. Applied Sciences (Switzerland), 8(7), 1–19. https://doi.org/10.3390/app8071206

Büttner, F., Winter, S., Rausch, S., Reustle, A., Kruck, S., Junker, K., Stenzl, A., Agaimy, A., Hartmann, A., Bedke, J., Schwab, M., & Schaeffeler, E. (2015). Survival Prediction of Clear Cell Renal Cell Carcinoma Based on Gene Expression Similarity to the Proximal Tubule of the Nephron. European Urology, 68(6), 1016–1020. https://doi.org/10.1016/j.eururo.2015.05.045

Cao, Y., Romero, J., & Aspuru-Guzik, A. (2018). Potential of quantum computing for drug discovery. IBM Journal of Research and Development, 62(6), 1–20. https://doi.org/10.1147/JRD.2018.2888987

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. Drug Discovery Today, 23(6), 1241–1250. https://doi.org/10.1016/j.drudis.2018.01.039

Chong, J., Soufan, O., Li, C., Caraus, I., Li, S., Bourque, G., Wishart, D. S., & Xia, J. (2018). MetaboAnalyst 4.0: Towards more transparent and integrative metabolomics analysis. Nucleic Acids Research, 46(W1), W486–W494. https://doi.org/10.1093/nar/gky310

Christiansen, E. M., Yang, S. J., Ando, D. M., Javaherian, A., Skibinski, G., Lipnick, S., Mount, E., O'Neil, A., Shah, K., Lee, A. K., Goyal, P., Fedus, W., Poplin, R., Esteva, A., Berndl, M., Rubin, L. L., Nelson, P., & Finkbeiner, S. (2018). In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images. Cell, 173(3), 792-803.e19. https://doi.org/10.1016/j.cell.2018.03.040

Ebadi, A., Xi, P., Tremblay, S., Spencer, B., Pall, R., & Wong, A. (2020). Understanding the temporal evolution of COVID-19 research through machine learning and natural language processing. Scientometrics, 0123456789. https://doi.org/10.1007/s11192-020-03744-7

Eisenstein, M. (2020). AI Brings Precision to Cancer Immunotherapy. Genetic Engineering and Biotechnology News, 40(S1), S10–S13. https://doi.org/10.1089/gen.40.S1.04

Ekins, S., Puhl, A. C., Zorn, K. M., Lane, T. R., Russo, D. P., Klein, J. J., Hickey, A. J., & Clark, A. M. (2020). Exploiting machine learning for end-to-end drug discovery and development. Nature Materials, 18(5), 435–441. https://doi.org/10.1038/s41563-019-0338-z.Exploiting

Ghanat Bari, M., Ung, C. Y., Zhang, C., Zhu, S., & Li, H. (2017). Machine Learning-Assisted Network Inference Approach to Identify a New Class of Genes that Coordinate the Functionality of Cancer Networks. Scientific Reports, 7(1), 1–13. https://doi.org/10.1038/s41598-017-07481-5

Guney, E., Menche, J., Vidal, M., & Barábasi, A. L. (2016). Network-based in silico drug efficacy screening. Nature Communications, 7(May 2015), 1–13. https://doi.org/10.1038/ncomms10331

Harrer, S., Shah, P., Antony, B., & Hu, J. (2019). Artificial Intelligence for Clinical Trial Design. Trends in Pharmacological Sciences, 40(8), 577–591. https://doi.org/10.1016/j.tips.2019.05.005

Hunt, T., Song, C., Shokri, R., Shmatikov, V., & Witchel, E. (2018). Chiron: Privacy-preserving machine learning as a service. ArXiv.

Jones, W., Alasoo, K., Fishman, D., & Parts, L. (2017). Computational biology: Deep learning. Emerging Topics in Life Sciences, 1(3), 257–274. https://doi.org/10.1042/ETLS20160025

Krempel, R., Kulkarni, P., Yim, A., Lang, U., Habermann, B., & Frommolt, P. (2018). Integrative analysis and machine learning on cancer genomics data using the Cancer Systems Biology Database (CancerSysDB). BMC Bioinformatics, 19(1), 1–10. https://doi.org/10.1186/s12859-018-2157-7

Kusumoto, D., & Yuasa, S. (2019). The application of convolutional neural network to stem cell biology. Inflammation and Regeneration, 39(1), 1–7. https://doi.org/10.1186/s41232-019-0103-3

Lecoutre, A., Negrevergne, B., & Yger, F. (2017). Recognizing Art Style Automatically in painting with deep learning. Journal of Machine Learning Research, 77(2016), 327–342.

Lehmacher, W. (2017). Global Dynamics and Key Trends. https://doi.org/10.1007/978-3-319-51115-3_3

Lengauer, T., & Sing, T. (2006). Bioinformatics-assisted anti-HIV therapy. Nature Reviews Microbiology, 4(10), 790–797. https://doi.org/10.1038/nrmicro1477

Liu, K., Ding, R. F., Xu, H., Qin, Y. M., He, Q. S., Du, F., Zhang, Y., Yao, L. X., You, P., Xiang, Y. P., & Ji, Z. L. (2020). Broad-Spectrum Profiling of Drug Safety via Learning Complex Network. Clinical Pharmacology and Therapeutics, 107(6), 1373–1382. https://doi.org/10.1002/cpt.1750

Madhukar, N. S., Elemento, O., & Pandey, G. (2015). Prediction of genetic interactions using machine learning and network properties. Frontiers in Bioengineering and Biotechnology, 3(OCT), 1–12. https://doi.org/10.3389/fbioe.2015.00172

Mardirossian, N., Wang, Y., Pearlman, D. A., Chan, G. K. L., & Shiozaki, T. (2020). Novel algorithms and high-performance cloud computing enable efficient fully quantum mechanical protein-ligand scoring. ArXiv.

Mishra, N., Kapil, M., Rakesh, H., Anand, A., Mishra, N., Warke, A., Sarkar, S., Dutta, S., Gupta, S., Prasad Dash, A., Gharat, R., Chatterjee, Y., Roy, S., Raj, S., Kumar Jain, V., Bagaria, S., Chaudhary, S., Singh, V., Maji, R., … Panigrahi, P. K. (2021). Quantum Machine Learning: A Review and Current Status. In Advances in Intelligent Systems and Computing (Vol. 1175, Issue October 2020). Springer Singapore. https://doi.org/10.1007/978-981-15-5619-7_8

Muzio, G., O'Bray, L., & Borgwardt, K. (2020). Biological network analysis with deep learning. Briefings in Bioinformatics, 00(April), 1–17. https://doi.org/10.1093/bib/bbaa257

Nistor, N., & Hernández-Garcíac, Á. (2018). What types of data are used in learning analytics? An overview of six cases. Computers in Human Behavior, 89, 335–338. https://doi.org/10.1016/j.chb.2018.07.038

Özyurt, F. (2020). Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. Journal of Supercomputing, 76(11), 8413–8431. https://doi.org/10.1007/s11227-019-03106-y

Parrello, B., Butler, R., Chlenski, P., Olson, R., Overbeek, J., Pusch, G. D., Vonstein, V., & Overbeek, R. (2019). A machine learning-based service for estimating quality of genomes using PATRIC. BMC Bioinformatics, 20(1), 1–9. https://doi.org/10.1186/s12859-019-3068-y

Properzi, F., Taylor, K., Cruz, M., Ronte, H., & Haughey, J. (2020). Intelligent clinical trials. 34.

Renaud, J. P., Chung, C. W., Danielson, U. H., Egner, U., Hennig, M., Hubbard, R. E., & Nar, H. (2016). Biophysics in drug discovery: Impact, challenges and opportunities. Nature Reviews Drug Discovery, 15(10), 679–698. https://doi.org/10.1038/nrd.2016.123

Scheeder, C., Heigwer, F., & Boutros, M. (2018). Machine learning and image-based profiling in drug discovery. Current Opinion in Systems Biology, 10, 43–52. https://doi.org/10.1016/j.coisb.2018.05.004

Srivastava, S. K., & Srinivasan, S. (2020). Intelligent Automation-Led Transformation of Clinical Data Management: A New Solution for a Smarter Biopharma Industry. 1–5. https://doi.org/10.1109/hydcon48903.2020.9242800

Telenti, A., Lippert, C., Chang, P. C., & DePristo, M. (2018). Deep learning of genomic variation and regulatory network data. Human Molecular Genetics, 27(R1), R63–R71. https://doi.org/10.1093/hmg/ddy115

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction apis. {USENIX} Security, 25(16), 601–618.

Vaske, C. J., Benz, S. C., Sanborn, J. Z., Earl, D., Szeto, C., Zhu, J., Haussler, D., & Stuart, J. M. (2010). Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. Bioinformatics, 26(12), 237–245. https://doi.org/10.1093/bioinformatics/btq182

Voillet, V., Besse, P., Liaubet, L., San Cristobal, M., & González, I. (2016). Handling missing rows in multi-omics data integration: Multiple imputation in multiple factor analysis framework. BMC Bioinformatics, 17(1), 1–16. https://doi.org/10.1186/s12859-016-1273-5

Warth, B., Siuzdak, G., Spectrometry, M., Jolla, L., & Diego, S. (2018). Data processing, multi-omic pathway mapping, and metabolite activity analysis using XCMS Online. 13(4), 633–651. https://doi.org/10.1038/nprot.2017.151.Data

Yang, B., Xu, Y., Maxwell, A., Koh, W., Gong, P., & Zhang, C. (2018). MICRAT: A novel algorithm for inferring gene regulatory networks using time series gene expression data. BMC Systems Biology, 12(Suppl 7). https://doi.org/10.1186/s12918-018-0635-1

Zurich, E. T. H. (2018). ETH Zurich Annual Report 2017. ETH Zurich, December, 2–2. https://www.rtda.gov.rw/fileadmin/templates/publications/RWANDA_Annual_Report_2018-2019_SHARING.pdf,

Cervera, A., Rantanen, V., Ovaska, K., Laakso, M., Nuñez-Fontarnau, J., Alkodsi, A., ... & Hautaniemi, S. (2019). Anduril 2: upgraded large-scale data integration framework. Bioinformatics, 35(19), 3815-3817.

de Alvarenga Mudadu, M., & Zerlotini, A. (2020). Machado: open source genomics data integration framework. bioRxiv.

Grüning, B. A., Fallmann, J., Yusuf, D., Will, S., Erxleben, A., Eggenhofer, F., ... & Backofen, R. (2017). The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic acids research, 45(W1), W560-W566.

Kargar, F., Savardashtaki, A., Mortazavi, M., Mahani, M. T., Amani, A. M., Ghasemi, Y., & Nezafat, N. (2020). In Silico Study of 1, 4 Alpha Glucan Branching Enzyme and Substrate Docking Studies. Current Proteomics, 17(1), 40-50.

Kyritsis, K. A., Wang, B., Sullivan, J., Lyne, R., & Micklem, G. (2019). InterMineR: an R package for InterMine databases. Bioinformatics, 35(17), 3206-3207.

Wyner, Z., Dublin, S., Chambers, C., Deval, S., Herzig-Marx, C., Rao, S., ... & Martin, D. (2020). The FDA MyStudies app: a reusable platform for distributed clinical trials and real-world evidence studies. JAMIA Open.

Aykul, S., & Martinez-Hackert, E. (2016). Determination of half-maximal inhibitory concentration using biosensor-based protein interaction analysis. Analytical biochemistry, 508, 97–103. https://doi.org/10.1016/j.ab.2016.06.025

Lipinski, C. A., Lombardo, F., Dominy, B. W., & Feeney, P. J. (2001). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings 1PII of original article: S0169-409X(96)00423-1. The article was originally published in Advanced Drug Delivery Reviews 23 (1997) 3–25. 1. Advanced Drug Delivery Reviews, 46(1–3), 3–26. https://doi.org/10.1016/s0169-409x(00)00129-0

Rudling, A., Gustafsson, R., Almlöf, I., Homan, E., Scobie, M., Warpman Berglund, U., … Carlsson, J. (2017). Fragment-Based Discovery and Optimization of Enzyme Inhibitors by Docking of Commercial Chemical Space. Journal of Medicinal Chemistry, 60(19), 8160–8169. https://doi.org/10.1021/acs.jmedchem.7b01006