Artificial Intelligence Research and Development
M. Villaret et al. (Eds.)

doi:10.3233/FAIA210142

Collection, Processing and Analysis of Heterogeneous Data Coming from Spanish Hospitals in the Context of COVID-19

Marta BARROSO [a,1], Adrián TORMOS [a], Raquel PÉREZ-ARNAL [a], Sergio ALVAREZ-NAPAGAO [a] and Dario GARCIA-GASULLA [a]

[a] *Barcelona Supercomputing Center (BSC), Spain*

Abstract. The COVID-19 pandemic has already caused more than 150,000,000 cases worldwide. In Spain this has lead to a massive and simultaneous saturation of all sanitary regions. Coherently, the quick and consistent understanding of the COVID-19 disease requires of the combined analysis of thousands of medical records generated by dozens of different institutions. In the context of the publicly funded CIBERES-UCI-COVID project, we have gathered, cleaned and preprocessed data from heterogeneous sources – more than 30 hospitals, with different data entry systems – in order to produce a unified database, of more than 6.000 patients, that is used in several clinical studies being carried by different multidisciplinary groups. In this paper, we identify the complexities we encountered, the solutions we applied, and we summarise the statistical and machine learning techniques we have applied for the studies.

Keywords. COVID-19, Machine learning, Data migration, Continuous development and integration (CD/CI), Automated report generation

1. Introduction

The COVID-19 pandemic has been the first humankind has faced as a completely digitised society. Thanks to that, developed countries had the resources to collect information related to COVID-19 efficiently and in real-time from early stages of the pandemic, resulting in the creation of multiple medical databases overnight. Clearly, the variety of information sources complicates the homogenisation and aggregation of data, two necessary steps for any automated analysis, mining or learning to be made on that data.

In May 2020, the CIBERES-UCI-COVID [9] project was awarded, funded by ISCIII. This project has the goal of carrying out an in-depth, retrospective and multicenter analysis on the distribution, correlations and missing values of covid-infected patient data in Spain. Within this project, artificial intelligence (AI) is used for extracting information about the factors involved in mortality, for classifying patients according to certain patterns, and for estimating the time for a group of individuals to experience an event of interest (*e.g.*, reach a critical condition or require mechanical ventilation), among other

[1] Corresponding Author: Marta Barroso, c/Jordi Girona, 31, 08034 Barcelona, Spain; E-mail: marta.barroso@bsc.es

things. To feed all this processes, data is gathered from many different hospitals in Spain, including several specific sources such as Getafe hospitals and the SEMICYUC consortium (which have their own data storage system). For gathering, processing and exploiting such amount of information, the collaboration of experts from interdisciplinary fields is required. For this reason the CIBERES-UCI-COVID consortium is composed by medical doctors, bioinformatics and AI researchers. The authors of this paper have the latter role, and were in charge of most of the automation process, as detailed next.

In this paper we review some of the methodologies used within CIBERES-UCI-COVID for two different purposes. First to collect, aggregate and summarise the available information in an accessible manner. And second to exploit this information through analysis, mining and learning methods for producing novel insights of interest. In detail, the contributions of the paper are the following:

- Describing the process to develop a complete and unified database derived from REDCap platform (a data gathering, form-based system). We also detail the structure and codification of original data and discuss the complexities of current representation in §2.
- For the sake or re-usability and accessibility, we review the technical requirements of the project, identifying key aspects and functionalities that our work must provide to the medical experts and the rest of involved institutions. In addition, for structuring information and data flows, several proposals have been taken into account. The selection criteria and the current proposal are discussed in §3.
- Developing a fast and efficient method to combine different sources into one database. We also detail how this database is populated and kept updated regularly by means of migrations. During this process we homogenise data in order to avoid inconsistency issues. This is explained in depth in §4.
- Automating data pre-processing capable of transforming any dataset within the context of CIBERES-UCI-COVID and generating reports on missing data, outliers, correlations and feature selection analysis. The aspects reported for each pre-processing task are explained in detailed in §5.
- Defining a global configuration for automate pre-processing, report generation process and to generate validated data with the appropriate format for subsequent studies. This is discussed in §6.

2. Data sources

REDCap [5] is a web-based database for medical and biomedical research support created by the REDCap Consortium [4]. Given the familiarity of the medical partners with this technology, the CIBERES-UCI-COVID project uses this database as single source of truth. This database is filled by specialised stuff from the medical side sometimes temporary hired by each hospital, also named data entries. The data is collected and introduced into the platform manually. The considerable volume of patients that Spanish hospitals have received since the onset of the pandemic limits the number of medical professionals available on non-assistencial tasks, which makes data collection a complex task and susceptible to errors.

Inside REDCap, data is structured in entries, such that each entry represents a different patient. An entry consists in a series of forms, which contain organised registers with blank fields to fill. Each blank usually represents a variable that is then stored. These forms do not have a linear structure, and there are fields that depend on other variables. For instance, there are variables that depend on another variable having a certain value in order to appear (*e.g.*, duration of a treatment only appearing if a treatment is applied). The dependency hierarchy of a certain form can be obtained as an HTML file, but it is very complex and the structure it provides is very difficult to navigate. To access the data, REDCap offers an API. REDCap API is limited and does not allow to get more than a 1,000 patients at the same time.

REDCap codifies variables as textual (numerical and textual values, and dates), radio (categorical variables, shown in the forms as a radio button) and checkboxes (categorical values with more than one possible choice at the same time). Radio variables are codified as integers, in which usually “Unknown” or missing values are codified as a numerical value too. For checkbox variables, one binary value is stored per possible choice (*e.g.*, a checkbox variable with 5 choices needs 5 binary values to store). These representations are not efficient in terms of storage, and easily allow datatype-related errors such as a data entry writing letters in what should be a numerical value (*e.g.*, “175 cm” instead of “175”, “37^{o}C” instead of “37”).

Some of the variables may have more than one value at the same time, like bacterial or hemorrhagic complications suffered, or tests received. A patient may have suffered 3 different tests, performed at a different date each and with different results. In these kinds of situations, the fields in the form have dependencies with one another (*e.g.*, the fields for the second complication/test only appear if the first one is filled). In addition, each test or complication is stored in different predefined variables (*e.g.*, complication 1 variable, complication 2 variable, etc.), which is both storage-inefficient and a limited representation, as there is always a maximum amount of tests or complications that can be represented (for instance, only 3 bacteria complications can be defined).

Another case of dependency between variables are laboratory units. As different hospitals may use different units in their measurements, a form is dedicated for storing the units that are used in the measurements of entries. In the form corresponding to lab variables, for a certain variable, one value is stored for each possible lab unit. For instance, if haemoglobin was measured in g/dl or mg/dl, two values would be stored. A lab variable usually has 3 or 4 different unit options, and depending on the choice, one field or another appears in the form. Subsequently, one variable or another is the one that stores the value of the corresponding unit. This is inefficient in terms of space, as for a certain lab variable only a maximum of one variable, usually out of 3 or 4, stores a value.

In this project there is also a need to represent time-dependent data, as for instance blood analysis are performed periodically for a certain patient. As such, the forms of each entry are organised in events. Each event represents a different relevant timestamp (*e.g.*, hospital admission, admission in Intensive Care Unit (ICU), etc.). Each event has an associated date, and two or more of them may happen on the same date. When this happens, only one of the forms on the same date is filled and the others are empty except for a field that indicates that the information of the aforementioned forms is in a previous form.

3. Technical requirements and design

One of our first goals within the CIBERES-UCI-COVID project is to generate a database that could be used for any data analysis or AI purpose. Most of the data of each patient (*e.g.*, comorbidities, symptoms...) is present only once in REDCap, so the most natural way to translate this into a database is to design tables which contain a single data row per patient. There are exceptions such as blood analysis results, which are performed periodically. These need a table with several entries per patient and a time bound. As database model, we implement a relational database using MariaDB[2] which comes with a wide range of safety measures and it is faster and more efficient than MySQL[3]. Having the original data (and source of truth) as an external register like REDCap, it is fundamental to have a way of connecting to its API from our database and retrieve the data periodically (since the data is being added continuously during the project). It must be reliable, avoiding the corruption of data.

From clinical point of view, there is an interest of filtering patients according to different factors such as treatments received, events (*e.g.*, ICU admission and discharge, invasive mechanical ventilation start and end, etc), variables with value (*e.g.*, outcome or gender) and patient features (*e.g.*, hypertensive patients). For continuous variables, outliers are removed in order to avoid problems in statistical analyses. We were provided with a list of laboratory and ventilation variables and their normal ranges (on sick patients). For those observations that have variables outside its range, instead of removing the complete observation we ignore these variables.

In addition, results of analyses need to be presented in a clear and organised way such as reports. To avoid manually generating a large quantity of very similar reports, there is a need for an automatic report generator.

3.1. Integration with other data sources

Several studies about COVID-19 patients started concurrently with CIBERES-UCI-COVID. Due to the need for fast results, these studies worked independently and using uncoordinated data structures. Early in the project was decided to aggregate data from two other such studies into CIBERES-UCI-COVID because of the data volume they could offer, and the consequent benefits to any posterior analysis. Also, we establish to migrate these data into REDCap so that only a unique source of truth is maintained. These two independent studies are the SEMYCIUC consortium and Getafe hospitals, which both use spreadsheets as their main data storage. For each study an independent migration program is created, which maps the variables of the studies with the equivalent REDCap variables.

3.2. Target infrastructure

At this point, we define the dataflow of the project, in agreement with the identified technical requirements. In detail, we seek to have a single source of truth which can always be relied on, control over the data by those in charge of managing the analysis and enabling REDCap data modification with improvements made during analyses.

[2] MariaDB is an community-developed open source database system (https://mariadb.com/docs/)

[3] MySQL is an open-source database system developed by Oracle (https://dev.mysql.com/doc/)

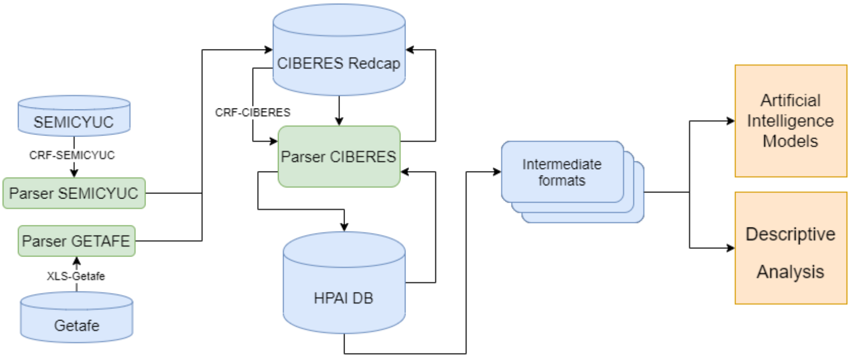

Figure 1. Dataflow implemented in CIBERES-UCI-COVID. As cylinders, data sources. In green, data parsers implemented by the authors.

This results in the scheme shown in Figure 1. The two external sources of data from SEMICYUC consortium and Getafe hospitals each have a specific parser (Parser SEMICYUC and Parser GETAFE), which makes them compatible for integration with REDCap. There is also a parser associated to REDCap's data (Parser CIBERES), which outputs an immutable database (HPAI DB) ready to be exploited by our algorithms.

4. Integration, reports and security

Beyond the technical requirements, we also identify a set of functionalities which are necessary to make a project of this scale feasible. For example, continuous development and integration (CI/CD), which allows us to validate if changes, after the code has been integrated in the app, are stable and correct. Currently this is a desirable feature since it allows to detect and repair failures faster and create workflows across the development, testing, and production environments. This process is automated by a pipeline, which contains all the stages needed to deliver a new version of the application. The pipeline itself can be divided in three sub processes: Code integrity and validation testing (including Unit tests and Integration tests), database migration (including error handling and data recovery) and automated generation of different types of reports.

4.1. Data migration

Data migrations from REDCap to the database are performed periodically, so that the database is kept updated when new registers are added to REDCap. The migration of REDCap data to our database consists of several sequential steps.

The forms that can be filled once per event are checked before the actual migration starts. The migration processes separates forms of the same type by date, and for each group of forms with equal date it looks for the one that is filled and duplicates the information in the rest. This way, our database explicitly contains all information. Because the retrieved data from REDCap is formatted entirely as text strings, the first step is casting each value to the corresponding type in our database. In a few exceptional cases, an additional string processing step is required to remove strange characters (e.g. temperature as "37°C" instead of "37"). When possible, a direct mapping between a REDCap value and a table column is defined and the value is inserted in the column as is. However, some of the columns from the database require additional transformations, like unifying measurements in various units. For these cases, a function that transforms the original value is defined. Both the direct mapping and the needed functions are specified in a configuration JSON file. Due to the number of patients and the amount of data available for each of them, this process has been parallelised.

4.2. Report generation

Report generation is of special interest given their key role in the interaction with the medical participants. In this case, a library for automatically generating reports in different formats has been implemented. The system can generate table (csv files), correlation (matplotlib plots) and text reports (docx files). The latter is used to display results of the different analyses such as missing value and outlier analyses.

The process for generating a report is composed of three main steps. In the first place, the user instantiates a Configuration object. This class contains the minimum variables required to generate reports such as the list of variables and population filters but also contemplates the possibility of adding new fields easily. As a result, the system builds a validated dataframe with the requested variables and the restrictions already applied to it. The conducted analyses are applied to these data and their results are saved in the corresponding report.

Finally, considering the personal nature of all data being handled, special focus has to be put onto security measures. All data is stored in a private isolated – via virtualisation – server placed in the EU (Spain). Only an automated GitLab[4] CI/CD pipeline has access to this server unless emergency access is needed, and in this case only authorised researchers can access via SSH tunnelling using a private key. All private data is stored on a MariaDB database. If data has to be uploaded or downloaded by third parties, we use a MinIO[5] encrypted – at rest and in transport – distributed S3-compatible storage server. All credentials related to the project are stored in a secure Hashicorp vault[6]. Tokens to access this vault are only available to specific code repositories and authorised researchers. By combining repository-based authentication with a token-based vault, we enforce that the storage and processing of data is carried out, in an automated fashion, by code reviewed and pushed by the authorised researchers.

Communication with doctors as well as file sharing are done through GitLab and Rocket.Chat[7] with mandatory two-factor authentication. Role-based permission management is enabled in GitLab, so all users can participate in writing and commenting tasks but only a few users can see or push code. This allows us to restrict, track and audit who, when and why users access data, as well as to carry out a collaborative follow-up and evolution of the sub-studies.

[4] GitLab is a version control application with some DevOps tools https://docs.gitlab.com/

[5] MinIO is an encrypted cloud storage service (https://min.io/)

[6] Vault is a data protection and management tool (https://www.hashicorp.com/products/vault)

[7] Rocket.Chat is a chat service placed on a private server (https://docs.rocket.chat/)

5. Data pre-processing

In this section we review the main pre-processing steps performed, prior to any consequent analysis. Those are the detection and handling of missing values, the definition of outliers, the analysis of correlated data, the creation of new features, and the detection and management of errors.

Missing analysis and imputation One of the most common situations when working with real data is the existence of missing data. These missing values arise due to many reasons such as undefined values, data input errors, irrelevant information, mismatch of variables between databases, etc. In the context of CIBERES-UCI-COVID, where data is introduced by tens of different data entries (each hospital hiring its own), and where hundreds of medical variables are requested for each sample/patient, missing data is frequent and must be addressed thoroughly. Not handling missing data properly can have a negative impact on performance of machine learning models. As the authors of [6] point out, missing values can reduce statistical power and representativeness of samples, introduce bias and reducing drastically the quality of the study. In our case, we obtain a median of 168 (0.0065%) (IQR 12-9466) missing values per variable and 248 (0.0096%) (IQR 209-351) missing values per patient.

Missing values are replaced by estimated values based on other variables using K-Nearest-Neighbour or Multiple Imputations by Chained Equations techniques, using the fancyimpute library[8]. In view of those considerations, CIBERES-UCI-COVID offers a set of techniques that analyses missing data in depth in order to understand its nature and address the issues mentioned before. Depending on the study being carried and the variables being targeted by them, a combination of some or all of these methods can be applied, so we implemented them in a way that they can be activated, deactivated, and composed.

Outliers analysis Before performing any statistical analysis, outliers are to be removed. To identify outliers, medical expertise is of capital importance, as they can define the data ranges that can be considered as feasible. There are variables that must be carefully supervised due to their clinical importance. If outliers are located in large numbers, the project coordination must consider the possibility of contacting the hospital in question to understand and fix the source of outliers. For this reason, an analysis of outliers by variable and by hospital is carried out. Taking into account continuous variables, a total of 6468 (0.22%) outliers are generated with a median of 44 (IQR 26.5-144) outliers per variable.

Correlation analysis To interpret the relationships that may exist between features, a correlation analysis is performed. Understanding these relationships is useful in order to avoid multicollinearity and redundancy issues. Correlation is computed by different means depending on the data types. For pairs of continuous variables, Pearson is used to measure their statistical relation. For pairs of categorical variables, we measure the correlation ratio [2] by computing the relation between the statistical dispersion within individual categories and the dispersion across the whole population or sample. Finally, for pairs of categorical and continuous variables we use Cramer's V [1], which is a mea-

[8] fancyimpute is a python imputation library. We used version 0.5.4, this library was developed by Alex Rubinsteyn and Sergey Feldman (https://github.com/iskandr/fancyimpute)

sure of association based on Pearson's chi-squared statistic between two nominal variables (a value between 0 and 1). Correlations are computed for all pairs of features in a dataframe and reported to the medical team to decide which variables have higher clinical relevance.

Feature engineering After the collection and transformation of REDCap data, we compute derived medical variables and the enrichment of other ones through domain knowledge. It is the case of scoring systems such as APACHE II or SOFA used to measure the critical state of a patient. Most derived variables are computed and saved in the database after migration. There are others that are used only in very particular studies, so it is not necessary for them to be saved permanently. For instance, delta variables, which compare the results for a variable in different events, are computed at execution time. These variables are used to measure the evolution of a variable as the patient progresses through the different hospital stages.

Some variables are enriched with knowledge from medical experts. This includes variables that can change their value when some conditions are fulfilled. Usually, these are variables that are involved in the calculation of other more complex variables. This is the case for example of the Glasgow Coma Scale, if it equals 15 for some medical event then the rest of the events take this value.

Error handling The error handling is basically divided into two tasks: controlling the errors that may occur in our system and those derived from the REDCap data. For the former, as mentioned above, we perform unit and integration testing. This allows us to quickly detect and resolve bugs, refactor and improve the code, reduce complexity and ensure that all code meets quality standards before it is deployed.

In the case of REDCap errors, our task is to inform the centers so that they can be corrected in the platform. Among the errors found we distinguish variables with wrong units, with impossible (the value is very far from the normal range) or inconsistent values such as variables whose value may be incorrect when observed in combination with others, either because they are part of a sequence or that depend on other variables.

The risk that we encounter these types of errors is high when data is entered manually, so we must be careful when processing it before conducting any analysis.

Feature selection In order to be able to determine those variables that have a greater impact predicting the outcome of interest, importance rankings are generated using several feature selection techniques. As conventional methods we find regression (lasso, logistic) and classification algorithms (random forest) and embedded methods (recursive feature elimination). Techniques dedicated to survival analysis have also been incorporated extracted from PySurvival [9] library.

Statistical analysis It is performed to interpret data and discover patterns and trends. Given a set of variables, categorical variables are presented as frequency/percentage of a group from which they were derived, and for continuous variables the median [interquartile range (IQR)] is used. Categorial variables were compared with the use of Chi-square test or Fisher's exact test, while continuous variables were compared with the Student's t test or Mann–Whitney U test. For comparing variables between more than two groups one-way ANOVA and Kruskal tests are used. Missing values for each feature are ignored.

[9] pySurvival is an open source python package for Survival Analysis modeling (https://square.github.io/pysurvival/)

6. Enabling sub-studies

The scale of the CIBERES-UCI-COVID project (both in terms of samples and features) enables lots of research lines. These are called sub-studies, specialised research projects which feed off the project. In order to facilitate and scale sub-studies, we implement a set of enabling methodologies. In particular, a global configuration has been defined in order to avoid code repetition, automate pre-processing, report generation process and to generate validated data with the appropriate format for each sub-study.

The global configuration is a guide for the system to create the validated dataframe that includes the variables and the chosen population by the responsible of each sub-study. Once the dataframe is built, it is returned and its format is adapted to build the report. Among other things, the configuration contains information about the target type of report (*i.e.*, table, document or correlation report), list of variables needed and to be derived, how to split the population and variable ranges for imputation purposes.

The other main consideration for sub-studies is the population upon which it is based. This is defined by the values of certain variables (*e.g.*, age, gender, comorbidities, severity of disease, time of admission *etc.*), by mandatory variables (which cannot be missing) and by exclusion filters.

In addition to implementing the necessary tools to build a dataset to work with, CIBERES-UCI-COVID has developed a framework on which artificial intelligence techniques (classification, clustering, survival analysis techniques) can be easily incorporated. Thus, allowing the use of artificial intelligence to be within the reach of any clinical study under development. As a result, CIBERES-UCI-COVID has enabled the successful execution of several clinical studies such as [3,8,7] and others that are in progress. Currently under development are studies that aim to analyse the influence of intubation and tracheostomy strategies on the evolution of critical patients, risk factors involved in respiratory infection and the influence of metabolic disorders in the progression of the illness among others. Regarding survival, several methodologies have been implemented for predicting survival and analysing risk factors in hospital mortality.

7. Conclusion

In this paper, we describe the methodologies used in CIBERES-UCI-COVID consortium. One that spanned tens of hospitals, hundreds of variables and thousands of patients. These have allowed us to create a database with pre-processed, accessible and updated data from different data sources. With that in place, the project can foster multiple analysis and learning methods, carrying out studies of interest and high impact. In fact, roughly a year after the project started, several relevant research papers [3,8,7] have already been published thanks to this work, and to the data collected from the REDCap platform by means of regular migrations, and from the SEMYCIUC consortium and Getafe hospitals, through specialised parsers.

CIBERES-UCI-COVID also provides a set of pre-processing steps that allow to examine data in depth before conducting any further analysis. Among those steps we find missing analysis and imputation, outliers and correlation analysis in addition to feature engineering to enrich data after cleaning and error handling in a effective way. Results of those analysis are presented as reports through tables (csv files) and documents (docx files) generated in an automated way. Pre-processing and data retrieval are automated as well. A global configuration is used in order to manage which variables and filters we want to report and its format.

To sum up, this work has resulted in a scalable, efficient and flexible tool to generate validated data adapted to an arbitrary number of medical studies while collectively establishing and strictly following a set of good practices in design, implementation and security. Significantly, this work has been done under pandemic conditions, in less than a year, and under the scientific pressure generated within a huge project such as CIBERES-UCI-COVID.

Acknowledgements

The authors are very grateful to CIBERES-UCI-COVID consortium with special mention to Antoni Torres, Anna Motos, Laia Fernández-Barat, Ferran Barbé and Albert Gabarrús for their collaboration and support in the development of this paper.

References

[1] Harald Cramer. *Mathematical methods of statistics.* Princeton University Press, Princeton, 1946.

[2] R.A. Fisher. *Statistical methods for research workers.* Edinburgh Oliver & Boyd, 1925.

[3] Jessica González, Iván D Benítez, Paola Carmona, Sally Santisteve, Aida Monge, Anna Moncusí-Moix, Clara Gort-Paniello, Lucía Pinilla, Amara Carratalá, María Zuil, et al. Pulmonary function and radiologic features in survivors of critical covid-19: A 3-month prospective cohort. *Chest*, 2021.

[4] Paul A Harris, Robert Taylor, Brenda L Minor, Veida Elliott, Michelle Fernandez, Lindsay O'Neal, Laura McLeod, Giovanni Delacqua, Francesco Delacqua, Jacqueline Kirby, et al. The redcap consortium: Building an international community of software platform partners. *Journal of biomedical informatics*, 95:103208, 2019.

[5] Paul A Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, and Jose G Conde. Research electronic data capture (redcap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics*, 42(2):377–381, 2009.

[6] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64:402–6, 05 2013.

[7] Lucía Pinilla, Ivan D Benitez, Jessica González, Gerard Torres, Ferran Barbé, and David de Gonzalo-Calvo. Peripheral blood micrornas and the covid-19 patient: methodological considerations, technical challenges and practice points. *RNA biology*, 18(5):688–695, 2021.

[8] Ana P Tedim, Raquel Almansa, Marta Domínguez-Gil, Milagros González-Rivera, Dariela Micheloud, Pablo Ryan, Raúl Méndez, Natalia Blanca-López, Felipe Pérez-García, Elena Bustamante, et al. Comparison of real-time and droplet digital pcr to detect and quantify sars-cov-2 rna in plasma. *European journal of clinical investigation*, page e13501, 2021.

[9] Antoni Torres, María Arguimbau, Jesús Bermejo-Martín, Raquel Campo, Adrian Ceccato, Laia Fernandez-Barat, Ricard Ferrer, Natalia Jarillo, Jose Ángel Lorente-Balanza, Rosario Menéndez, et al. Ciberesucicovid: A strategic project for a better understanding and clinical management of covid-19 in critical patients. *Archivos de Bronconeumologia*, 2020.