INTRODUCTION TO QUANTITATIVE IMMUNOLOGY

LECTURE NOTES

Andrea Mazzolini

May 19, 2023

Contents

1 Introduction of the introduction to quantitative immunology

1.1 The immune system is a complex system

Everyone knows that the aim of the immune system is to detect, recognize and kill external pathogens. This is a very well defined task, but, at the same time, it is incredibly challenging and complex. There is a potentially infinite variety of pathogens that can be encountered and want to exploit our body, and, moreover, they can evolve very fastly, always trying new strategies to overcome our defenses. If you think for a second about it, this seems an impossible task for our body: we are facing a huge threat with very limited information (our DNA) and a limited energetic budget. Clearly, our DNA cannot contain the prescription of how to fight every possible pathogen that we can encounter, so smarter strategies have to be developed.

In this light, it is not surprising that when one tries to study our immune system it finds an extremely rich and complex machinery, with an incredible number of components that interact with each other. Indeed, we can consider the immune system as the second most complex system after the nervous one. One particular impressive feature of the immune system is that it acts across a huge list of spatial scales, Fig. 1. From the molecular scale, $\sim 10^{-9} - 10^{-8}$ m, e.g. for pathogen binding to lymphocyte receptors or the signaling pathways inside cells in response to external signals, to the scale pathogens that have to be recognized and killed, $\sim 10^{-7} - 10^{-6}$ m, the lymphocytes that communicate and interact, $\sim 10^{-5}$ m, the lymph nodes $\sim 10^{-2}$ m, the lymphatic and circulatory system, $\sim 10^{1}$ m, going until the population scale, $\sim 10^{6}$ m, where collective epidemiological effect can emerge, e.g. herd immunity. Each one of this scale the system is typically linked to all the other scales of the system: for example, a vaccine will boost the presence of receptors that very likely will recognize a given virus (antigen-TCR binding scale). As a consequence this can create herd immunity in a population and lead to the extinction of the virus. A similar reasoning can be done for temporal scales that can go from 10^{-3} s of the time of ligand-receptor interaction to the years of the immune memory against encountered pathogens.

If one goes more into the detailed mechanism of working of the immune system it can realize that it acts using different alternative strategies. We will briefly see some of those, but, in short, they can be grouped into two large categories: the innate response, more ancient, less specific but very fast and the adaptive response, slower but more specific and, in some sense, more powerful. These two categories can further be divided leading to a very complex picture of different layers and components that act in parallel and interact with each other. The complexity of the system and the huge amount of experimental data that new sequencing technologies give us provide a very hard challenge for biologists and scientists. In this context, quantitative sciences can be of great help in developing mathematical and algorithmic tools to better understand the system. This is indeed what happened in the last decades, where fruitful collaborations between biologists and quantitative sciences have become more common. In these lectures we will try to see a few of these successful works, trying to give a glimpse of the potential of collaborations between different disciplines applied to immunology.

1.2 What are these lectures about

The purpose of these classes is to give a biological introduction to the immune system functioning as well as presenting some examples in which quantitative methods have been successfully applied to understand this system. The next section 2 is dedicated to present a few key biological concepts. This is not intended to be a biological course and

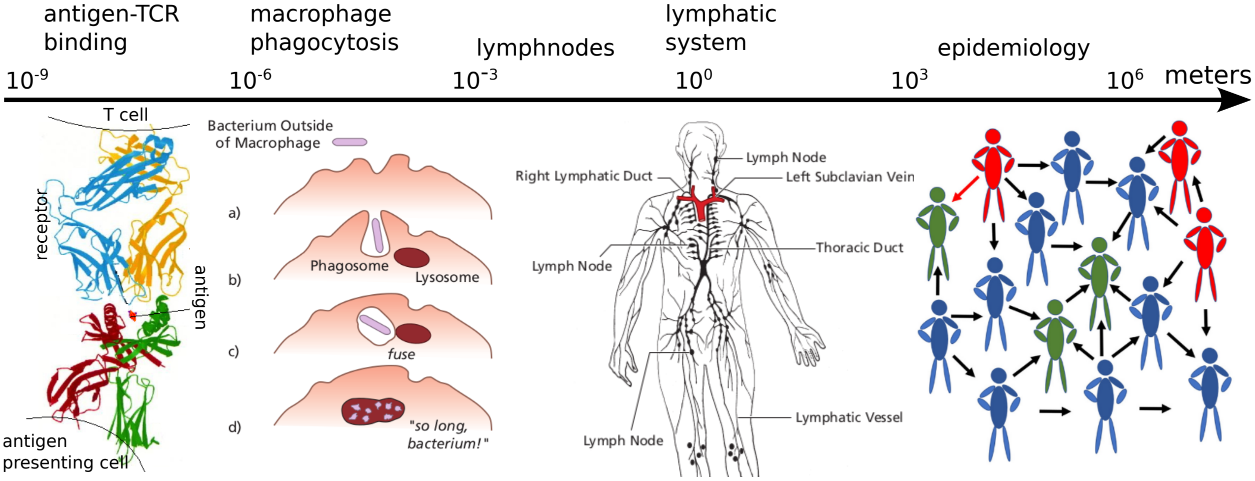


Figure 1: The immune system works across several order of magnitude of spatial scales. The pictures represent: the antigen-receptor binding, the macrophage phagocytosis, the lymphatic system and a network of social contacts through which a virus can spread.

does not provide a comprehensive description of the immune system. The presented biological facts are typically very much simplified. Moreover, some aspects are just mentioned or deliberately ignored, focusing more on the biology connected to the specific examples that we will see later in the lectures.

The core of the course is then discussed in section 3. In particular, we will see *data-driven* approaches to study problems in immunology. In general, the analysis starts from a specific type experimental data which are the main source of information in this field: *high-throughput sequences* of the *lymphocyte receptors*. What is a lymphocyte receptor and how the data are shaped will be explained in the following, at this stage it is sufficient to know that they will give a snapshot of the immune system status of a given person in a given moment. Non-trivial statistical tools are necessary to make sense of those data, which typically provide partial and noisy information about the system, and to use them to understand properties of how the immune system works. Starting from this, the main question that we have in mind is to understand which part of our system is interacting with a disease or a vaccine. This kind of information is crucial not only for the sake of understanding the system, but also for medical purposes, in diagnosis and in designing therapies and vaccines.

2 Biological background

A nice way of meeting all the main players of the immune system is to imagine that some external pathogen tries to enter the human body. The immune system fights this potential threat through multiple stages as roughly schematized in Fig. 2 and discussed in section 2.1. The figure seems already quite complex and full of arrows, but it actually focuses on a few main elements of the system, ignoring a lot of its complexity. You can imagine that the components and the arrows of the figure are just a small subset of the real ones and, moreover, several new players and interactions have probably yet to be discovered. If you want to go a bit deeper in the biology of the immune system (but without a specific biological jargon), this book is a very good reference: [1].

The next sections go more into details of specific properties, which will provide the background for the scientific work discussed in the rest of the lectures. In particular, we will focus on the concept of immune repertoire and how the body is able to generate a huge diversity of our lymphocyte receptors. All these concepts will be used in section 3.

2.1 The main steps to fight an external pathogen

As an introduction to the topic you can see also the video of the popular youtuber *kurzgesagt*, explaining what happens when you cut your finger `https://youtu.be/lXfEK8G8CUI`.

2.1.1 The innate response

Let us start our journey in the immune system by looking at the temporal line of Fig. 2. An external pathogen, that can be a bacteria or a virus wants to enter and invade our body. The first obstacle that it encounters are our physical

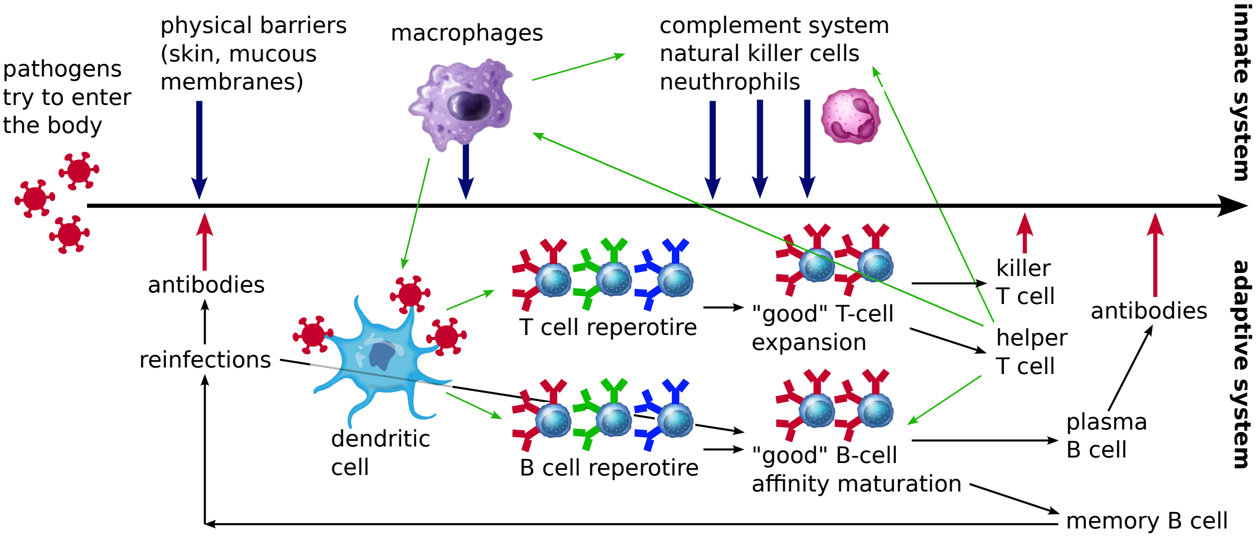


Figure 2: Different protections of the immune system against a pathogen entering the body. The thick black horizontal line is a temporal line that starts from the invasion. Blue arrows represent attacks of the innate immune system, red arrows of the adaptive system. Green arrows are triggering events or communications between different components of the system and are better explained in the text.

barriers. The most obvious that one can think about is the skin, but the mucous membranes that cover our digestive, respiratory and reproductive tracts are also extremely important.

If a pathogen breaks in, typically, the first agents that it finds on his way are the **macrophages**. They are white blood cells that can be present in all tissues and continuously patrol our body. They are attracted by proteins and molecules that they do not recognize as part of the body and, in case they see something unusual, they literally eat the potentially dangerous molecule or microorganism and digest it through enzymes. This process is called *phagocytosis*. In general, they become much more efficient if guided by other agents of our system: helper T-cells and antibodies, which are part of the adaptive immune system and described later. The way macrophage works is quite impressive: they are cells, but, in fact, they behave like autonomous hunters that are able to follow a prey in a complex environment and kill it. What is surprising is that this is a task that requires a sort of strategy (even if simple) that is based on some memory. How can a simple cell implement it without a nervous system? A short answer is that the complex network of chemical reactions within the cell can effectively be at the basis of non-trivial decision making. A lot of research tries to better understand this kind of system and how cells and simple organisms can follow chemical cues, i.e. chemotaxis.

The second role of macrophage, apart from killing, is to release in the body some chemicals that are under the name of **citokynes**. They attract other agents of the immune system and trigger attacks against the pathogen on different levels. We just list a few of them, just to give the idea of the arsenal of our immune system. Together with macrophages, **neutrophils** are the other most important type of cell that can phagocytate microbes. In addition to that, they can also release toxic chemicals that are able to destroy microbes together with our connective tissue. They are very powerful but also very dangerous cells and, luckily, when activated they have a short life of a few days to avoid causing too much damage. A different type of response is provided by the **complement system**. This response involves several different proteins which can be present in high concentrations in the blood. When activated those proteins are very fast in attacking and destroying external pathogens, directly or tagging them for a macrophage. The next players that we mention are **natural killer cells**. We can assign them two main roles: the first is as a factory of cytokines, that will orchestrate the activation and recruitment of other immune system responses. The second role is more active and consists in pushing other cells of our body to commit suicide, technically, to commit *apoptosis*. This happens when some cell of the body is not properly functional or is, for example, infected by a virus that is using it to replicate itself. Those natural killer cells are also useful in the case of cancer, where cells of our own body stop working and start to replicate without control and invade our organs.

All the mechanisms listed so far (upper part of Fig. 2) can be grouped together under the name of **innate immune system**. This system is, in general, present in all the animal kingdom and it has older evolutionary roots (with respect to the adaptive system discussed below). The common features of the innate responses are the rapidity, this is really the first line of defense, and the “unspecificity”, meaning its action is very similar against all the possible types of pathogens that we can encounter. If all this machinery is not enough to clear the infection and the bacteria or virus that entered our body has started to replicate without control, the second, slower but stronger player enters the game: the adaptive immune system.

2.1.2 The adaptive response

The adaptive immune system is evolutionary more recent and only present in vertebrates. In comparison with the innate system its action is slower but it has two big advantages: first, it is much more "specific" (in a sense that will be clearer later) and efficient against the external pathogen and, second, it allows our body to create immune memory of previous infections. This second feature makes future infections of the same pathogen much easier to contrast, providing the mechanism at the basis of vaccinations.

Coming back to Fig. 2, we now focus on the lower part of the image. After the first action of the innate immune system and the subsequent release of chemical signals, a special type of cells are recruited to the site of infection: the **dendritic cells**, which are the most important **antigen-presenting cells**. The task of those cells is to sample the "battlefield" where the infection is going on. That is to say that they collect fragments of other cells that have been destroyed by all the machinery of the innate response. Those fragments contain parts of proteins of the external invaders and are called **antigens**. The dendritic cells sample those antigens and show them on their surface, becoming a sort of showcase of molecular samples of the battlefield. After having collected those samples, the cells travel from the site of infection to lymph-nodes, where the key process of antigen presentation takes place.

Here, one of the main characters of our story comes into the game: the **T-cells**. They are lymphocytes with a unique and special feature: they are characterized by a receptor on their surface, whose shape has a huge variability within the population of T-cells. Its function is to recognize and bind to a specific antigen having a precise molecular structure, with some *tolerance*. The diversity of receptors of the T-cell population is so big that we expect to have at least a good receptor for each possible antigen of external pathogen that can be encountered. This defines the concept of **immune repertoire**, which is central to these lectures and it will be treated more in detail in section 2.2. In the lymph-node, this population of different receptors is exposed to the antigens presented by the dendritic cells. It will happen that a few of them will have a strong *affinity* in binding some of the foreign antigens. The dendritic cell sees the good T-cell attached to its antigen and gives it an activation signal that triggers the subsequent response, which involves a series of cell duplications and, therefore, an exponential expansion of the number of T-cells with the right receptor. This process is also called **clonal expansion**, where with the term "clone" or "clonotype" refers to the family of T-cells having the same receptor. Therefore, one has to imagine that the immune repertoire is composed of a huge amount of different receptors/clonotypes which typically are present in a small number of copies, like a sort of stockpile. Most of the receptors will never be activated, but a bunch of them, depending on the encountered infections, will increase in number and be released in the body. In this sense the response of the adaptive immune system is "specific": differently from the innate system, each pathogen will trigger different clonotypes specialized in fighting them and, therefore, much more effective in destroying them.

After the activation, T-cells can be broadly classified in two big classes: **killer T-cells** and **helper T-cells**. As the name suggests, a killer cell is a front line fighter against pathogens and it works in a similar way to the natural killer cell. It recognizes that a cell of our body is contaminated by the pathogen and send it a signal for its suicide. This can be done because our cells continuously expose on their surface fragments of proteins that are currently expressed, and can potentially contain the targeted antigen of the invader. This mechanism hides a potential problem: most of the exposed proteins are our proteins and, clearly, we do not want the T-cell to recognize those, leading to auto-immune diseases. However, in general, T-cells are extremely good in recognizing self from non-self. This is because, after their creation, there is a "quality check", called **thymic selection**, that tests if the receptors can bind to self-proteins and, in such a case, kill them. Differently from killer T-cells, helper T-cells have the role of orchestrating the action of other parts of the system. They have several different functions, here we mention the ability of enhancing the macrophages and the natural killer cells, making their action more efficient and specific against the pathogen. They also have an important role in the affinity maturation process of B-cells that we will see below.

The final player is the **B-cell**. Similarly to T-cells, those cells are characterized by a large diversity of receptors. However, the process that follows the finding of a good B-cell receptor is very different. This is called **affinity maturation** and its aim is to further enhance the binding affinity of the receptor to the target. This is done through an evolutionary Darwinian process in which the founder B-cell starts to replicate and during the replication several errors, i.e. mutations, are made in the amino-acid chain of the receptor. Then those new cells with mutated receptors will be tested against the antigens and only the cells with better binding affinity will be selected, while the others undergo apoptosis. This process repeats for several rounds and it will increase by several factors the speed and the strength of the receptor-antigen binding. Progressively, B-cells will leave this maturation process and can become of two types: **plasma cells** or **memory cells**. The first type will go to fight the infection as a factory of **antibodies**. An antibody is a small protein having exactly the same shape of the receptor of the mother plasma cell, which, therefore, binds very efficiently to the invader (or to cells infected which expose their antigen on their surface). There are different antibodies that are specialized to different parts of the body and enter in action at different stages of the disease but, in general, they will tag the invader which will be instantly destroyed by all the machinery of the innate immune system. The alternative fate of a B-cell after maturation is to become a memory B-cell. It will not join the fight, but it will stay around for years (plasma B cells will have a much shorter life). They are at the basis of the immune memory and, if the same (or a very similar) pathogen will reinfect the body in the future, the memory cell will recognize it and it will quickly trigger the immune response. Note that, in this case, the body has ready the good receptors against the pathogen and the slow machinery of the antigen presentation, which takes days or weeks, can be passed over. An army of the right antibodies can be immediately released and effectively fight the pathogen, which can be cleared out much more easily.

2.2 The immune repertoire

2.2.1 The immune repertoire is a collection of clonotypes

There are $M \sim 4\ 10^{11}$ T-cells and a similar number of B-cells circulating in our body. As we saw before, what characterizes this kind of cell is the receptor found on the surface of the cell, whose specific sequence of amino-acids allows it to bind to a specific antigen. All the cells having the same receptor are called *clonotype* and within our body there can be several copies of cells belonging to a clonotype, i.e. that have the same receptor. As we saw before, this number can vary in time, for example, in response to a disease, where the T-cell clonotype having the good receptor against the pathogen can increase in number. Let us indicate this number with n_i for the clonotype i and, therefore, we have that

$$\sum_i^U n_i = M.$$

The upper extreme of the summation, U, is the number of unique receptors that are present in our body and represents, somehow, the diversity of our immune repertoire. In general, we can describe the state of our system by the collection of clonotypes that we have and their abundance $\{n_1, n_2, \dots n_U\}$.

Most of the progress in understanding the immune repertoire have been achieved by *sequencing experiments*, which allows us to obtain the sequences contained in a biological sample, typically blood. These kinds of techniques revolutionized the biology of the last decades, especially from the development of *high-throughput methods* that increased the number and the reliability of the sampled sequences by also reducing the price of those experiments. In our case, we can imagine that the outcome of experiment is exactly a sub-sample of clonotypes with their abundance:

$$\{\hat{n}_1, \hat{n}_2, \dots \hat{n}_{\hat{U}}\} \in \{n_1, n_2, \dots n_U\}.$$

More precisely, sequencing experiments provide us tables like the one shown in Fig. 3. This table lists sequences of what is called **CDR3** region. This is not the full sequence of the receptor but the most interesting: it is the most variable region and the one that characterizes which shape the receptor will take once folded and which protein will bind. Therefore, for several purposes we can identify the full receptor with the CDR3.

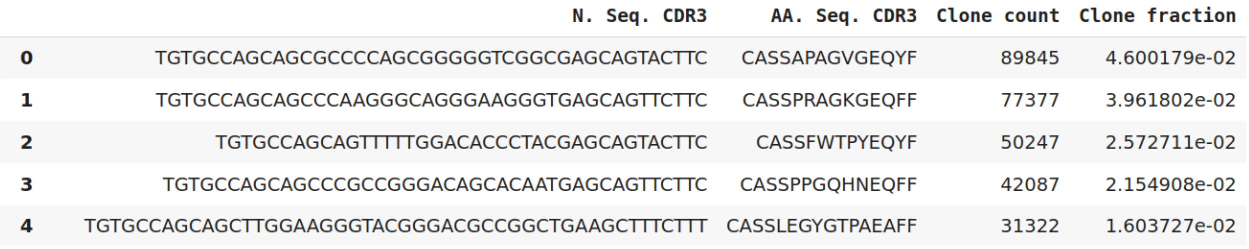
The sub-sample is clearly much smaller than the total system: typically a good experiment can lead to $\hat{M} = \sum_i \hat{n}_i \approx 10^6/10^7$ receptors. Therefore, when inferring properties of the immune system one has to take into account that we are looking at an infinitesimal part of the whole, but, anyway, this provides a lot of information on our repertoire. In general, if we neglect every source of noise of the experiment, we have anyway a sampling noise:

$$P\left(\hat{n}_i|\hat{M}, \frac{n_i}{M}\right) = \text{Binomial}\left(\hat{n}_i|\hat{M}, \frac{n_i}{M}\right) \approx \text{Poisson}\left(\hat{n}_i|\lambda = \hat{M}\frac{n_i}{M}\right). \tag{1}$$

Often, however, the the sequencing experiment introduces biases that leads to a larger noise. A typical way of modeling these effects is to choose a noise distributed as a Negative Binomial distribution.

An important property of the immune system is the fact that the distribution of the number of receptors $\hat{n}_i$ follows a power law distribution with a very robust exponent: $p(\hat{n}_i) \sim \hat{n}_i^{-\alpha}$, where $\alpha \approx 2$. This distribution is studied a lot in immunology [2, 3] since it is believed that its specific shape provides information about the generative process of the repertoire. Another advantage of this property is that it allows us to check the quality of a given dataset: data where the clone-size distribution deviate from this prediction are usually suspicious. In general, power law distributions are typical signatures of complex systems, and huge amount of literature try to investigate the mechanisms at their origin [4, 5, 6].

Trying to make a (probably) forced but useful metaphor, our whole immune system repertoire is like a huge LEGO construction made of basic elementary components, i.e. the clonotypes which play the role of the LEGO bricks. Our



	N. Seq. CDR3	AA. Seq. CDR3	Clone count	Clone fraction
0	TGTGCCAGCAGCGCCCCAGCGGGGGTCGGCGAGCAGTACTTC	CASSAPAGVGEQYF	89845	4.600179e-02
1	TGTGCCAGCAGCCCAAGGGCAGGGAAGGGTGAGCAGTTCTTC	CASSPRAGKGEQFF	77377	3.961802e-02
2	TGTGCCAGCAGTTTTTGGACACCCTACGAGCAGTACTTC	CASSFWTPYEQYF	50247	2.572711e-02
3	TGTGCCAGCAGCCCGCCGGGACAGCACAATGAGCAGTTCTTC	CASSPPGQHNEQFF	42087	2.154908e-02
4	TGTGCCAGCAGCTTGGAAGGGTACGGGACGCCGGCTGAAGCTTTCTTT	CASSLEGYGTPAEAFF	31322	1.603727e-02

Figure 3: First five lines of a typical table containing clonotypes with their nucleotide sequence, amino-acid sequence, count (which is the n_i introduced in the text) and fraction: n_i/M. One can expect to have order 10^5/10^6 total lines.

experiments give us access to a sub-sample of bricks and our duty is to understand something about the how the different brick can be joined together (how the different clonotypes interact) and, hopefully, what was the original building from which they are taken from, (how the immune system behaves and organizes itself). However, an important missing piece of this metaphor is the fact that the immune system is dynamic and responds to the infections that we periodically encounter, adding an additional layer of complexity.

2.2.2 How can we generate the huge diversity of the repertoire?

Here we go back to the variable U defined in the previous section: the number of unique clonotypes in our body. Estimates of this number are difficult to make, due to the fact that, as mentioned before, we have experiments that strongly sub-sample the real repertoire. Moreover, we get mostly the largest clonotypes (which can be extracted with larger probability) and it is very difficult to infer the statistics of the rare ones. Nowadays, this number is considered to be in a window of $U \approx 10^7 - 10^{10}$ [7]. If you think about it for a second, this number is huge. Lymphocytes receptors are proteins and proteins are made by reading the instructions of portions of the DNA that are called genes, which in our DNA are around $2.5\ 10^4$. Even in the most unlikely scenario that most of our genes are dedicated to generating receptors, there is no way of creating such a huge diversity.

To overcome this problem, those lymphocytes undergo a random process called **VDJ recombination** that changes a portion of DNA in a unique way, and is schematized in Fig. 4. note that this is an exception of our cells, that, in general, have exactly the same sequence of DNA. Since this part of DNA is unique and encodes for the proteins that compose the receptor, the receptor will be, in turn, unique and, when folded, will have a shape that will recognize and bind to a unique protein.

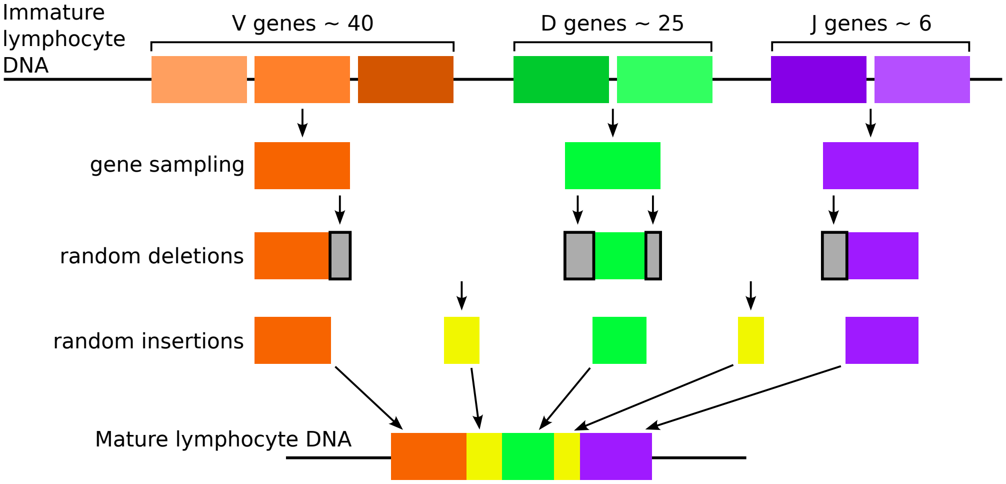


Figure 4: Sketch of the VDJ recombination that leads to a DNA sequence that codes for the lymphocyte receptor.

Let us now go into more details on how the DNA is changed through the VDJ recombination. When lymphocytes are created they are called immature and their DNA will be the same as all our other cells. The portion encoding for receptors will look like the upper part of Fig. 4: There is a sequence of approximately 40 genes of type "V", followed by a sequence of 25 "D" genes and 6 "J" genes. B and T-cells then perform a maturation process, in the thymus for the T and in the bone marrow for the B-cells. During this process this DNA portion is rearranged by cutting, pasting, deleting and inserting new nucleotides. In particular, only one among the V genes, one among the D and one among the J are chosen, and the new DNA fragment will contain a random sample of the three genes. In this way the cell can generate 6000 different DNA sequences, which are a lot but not enough. In fact, during the recombination, other two random processes take place. The first is the deletion of a random number of nucleotides at the right end of the V gene, at the left end of the J gene and at both ends of the D gene. Therefore each copy of the genes in the mature DNA strand will not be exactly equal to the original but a bit shorter. Moreover, in between V and D and D and J two new short sequences of randomly generated nucleotides are inserted. This leads to the huge diversity that we observe in sequencing experiments.

In section 3.1 we will try to model this process with a probabilistic description and fit our model parameters with the real sequences. At the end, one has a very precise quantitative description of the VDJ recombination and, for example, it can give us information on its specific properties, e.g. what is the probability that one among the 40 V genes is chosen? How many deletions of nucleotides can I expect from it? Moreover, the model has a very strong predictive power, and can give access to the probability that a given receptor is generated, even if this receptor has been never seen in data. It turns out that these probabilities are very broadly distributed as our body can produce with very high probability some receptors, which are several orders of magnitude larger than probabilities of other receptors. This kind of information is crucial in understanding which receptors of our system are interacting with a disease as we will see in section 3.2. The problem is that if we see an enrichment of a given receptor for a given disease in different patients, this does not imply that this receptor is involved in the response. Maybe it is present just because it is very easy to produce by the recombination process. Therefore, to infer which receptors are involved in a response, one has to take into account this generation probability.

3 Extracting information from receptor sequences

3.1 Inferring the generation probability of a receptor

The study of the VDJ recombination, section 2.2.2, with quantitative methods is very recent and starts in 2012, [8, 9, 10]. Those series of works have had a good impact among people with a quantitative background working in immunology, but also among biologists and physicians, because of the reliable predicting power of the model.

The basic idea is to describe the VDJ recombination as a stochastic process, which encodes the rules shown in Fig. 4 for recombining the DNA and generating a receptor sequence. The general ideas of what the rules are, e.g. one V gene is selected then a few bases are deleted and so on, are known biologically, but they have not been quantified in a precise way. For example there are around 40 V genes, but the probability of choosing one of these genes in the process is not known. Are they uniformly chosen? Or some genes are more likely to be selected? How many nucleotides are typically deleted from the gene side? All this question can be answered by using a maximum likelihood inference. For more details, the general setting of maximum likelihood is discussed in section 4.1. In short, the probabilistic model is built with some parameters that encode all the different events of the VDJ recombination, among those, the probability of selecting each of the 40 genes and the probability of deleting a given number of nucleotides. All the parameters are contained in the set θ. As discussed in section 2.2, there are experiments that give access to a subset of the receptors present in humans, which we indicate with $\mathbf{x}$. The model allows us to write down a probability for generating all those sequences if in my model I have a set of parameters θ: $P(\mathbf{x}|\theta)$, which is also called the likelihood of the model. The real parameters can be inferred by maximizing $P(\mathbf{x}|\theta)$ over θ. Those parameters θ^* are our best guesses on all the specific events that happens during the VDJ recombination.

To better understand all this procedure we tackle a much simpler invented VDJ recombination discussed in the next section. This allows us to familiarize with the maximum likelihood setting, which is one of the most important frameworks when dealing with inference of data. We also look more in details a specific technique to solve the likelihood maximization, called the expectation-maximization algorithm.

3.1.1 A simpler toy example

The simpler toy example is shown in Fig. 5. We basically neglect the presence of D and J genes, as well as the insertion events. Moreover we consider a set of only four genes. An implementation of this model and its solution is discussed in the notebook "VDJ_exercise.ipynb" in the repository of the course: `https://github.com/amazzoli/Quantitative_immunology.git`.

As previously mentioned, we want to apply the maximum likelihood framework to the VDJ toy model. Let us first start to describe precisely the rules of the model and the associate parameters. The first step of the model chooses one of the four V genes according to the probabilities $(P(V_1), P(V_2), P(V_3), P(V_4))$. These probabilities are also

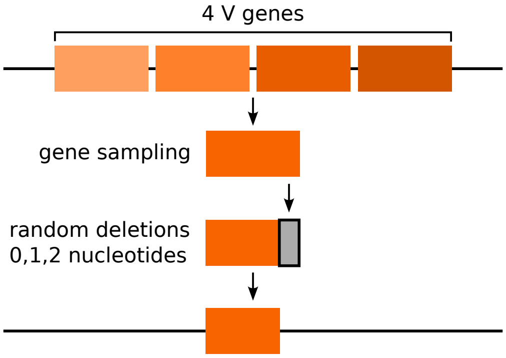


Figure 5: Sketch of the toy model for the VDJ recombination.

unknown parameters of our model. Notice that we actually have only 3 free parameters, since the fourth is fixed by the normalization $\sum_i P(V_i) = 1$. Then, the next and final step is to delete 0, 1, or 2 nucleotides from the right end of the V gene. The deletion probabilities depend on the gene sampled at the previous step, and they are, for each V_i, $(P(0|V_i), P(1|V_i), P(2|V_i))$. Again, these sets are normalized $\sum_k P(k|V_i) = 1$, leading to 2 free parameters for each V gene. Writing the 11 free parameters in a compact way, we have

$$\theta = \{P(V_i)\}_{i=1,2,3} \cup \{P(k|V_i)\}_{i=1,2,3,4;\ k=0,1}. \tag{2}$$

In the python exercise, before doing the inference, we first need to generate an artificial set of data. To this end, we choose a true value for each of these parameters, θ_{true} that our experiment follows. Then we sample N sequences generated in this way from our true model, leading to the dataset $\mathbf{x}$. Then we move to the actual inference and we forget about θ_{true}. Starting from $\mathbf{x}$ we look for the parameters that maximize the likelihood of this dataset.

Therefore, we need to write our model in the language of the maximum liklihood. One difficulty of this setting (which is extremely amplified in the real VDJ model) is that at one sequence x_i can correspond multiple “scenarios” that can originate it. For example, assuming that there are two artificial V genes TCA and $TCAA$, a sequence in the dataset TCA can be generated by the extraction of the V gene TCA with 0 deletions or from $TCAA$ with 1 deletion (see the notebook for more examples). We use the notation s to indicate a specific scenario, which, more precisely, is defined as a pair $s = (V_i, k)$ of the V gene V_i with k deletions. We have that the probability of a scenario is

$$P(s|\theta) = P(V_i, k) = P(V_i)P(k|V_i), \tag{3}$$

where the second equality uses the definition of conditioned probability and allows us to connect the scenario probability with our parameters.

Each scenario s generates a single sequence, but a sequence x can be generated by multiple scenarios. We indicate with $\mathcal{S}(x)$ the set of scenarios that lead to the sequence x. The probability of finding a given sequence in the dataset is then

$$P(x_i|\theta) = \sum_{s\in\mathcal{S}(x_i)} P(s|\theta) = \sum_{(V_i,k)\in\mathcal{S}(x_i)} P(V_i)P(k|V_i). \tag{4}$$

From the equation above, the likelihood can be recovered by assuming that all the N sequences are generated independently

$$P(\mathbf{x}|\theta) = \prod_{i=1}^{N} \sum_{s\in\mathcal{S}(x_i)} P(s|\theta) = \prod_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} P(V_j)P(k|V_j). \tag{5}$$

Through its maximization with respect to θ we can obtain the set of parameters θ^* that better fit the dataset and, therefore, gives us information about the underlying biological process. However, maximizing the function above is not a trivial task. As section 4.2.4 shows and the python netbook implements, we employ an expectation maximization algorithm (whose general definition is in section 4.2).

In Fig. 6 we show the results of the inference. One advantage that we have with respect to the real VDJ recombination is that we know what is the right answer: the θ_{true} we used to generate the data. The left panel shows that the likelihood of the inferred parameters is as good as the likelihood of the true parameters and much larger than a uniform random guess. This is a good news and is saying that the maximization works very well. However if we compare the specific probabilities of generating, for example, the V genes (right plot) we have small discrepancies. There are two factors

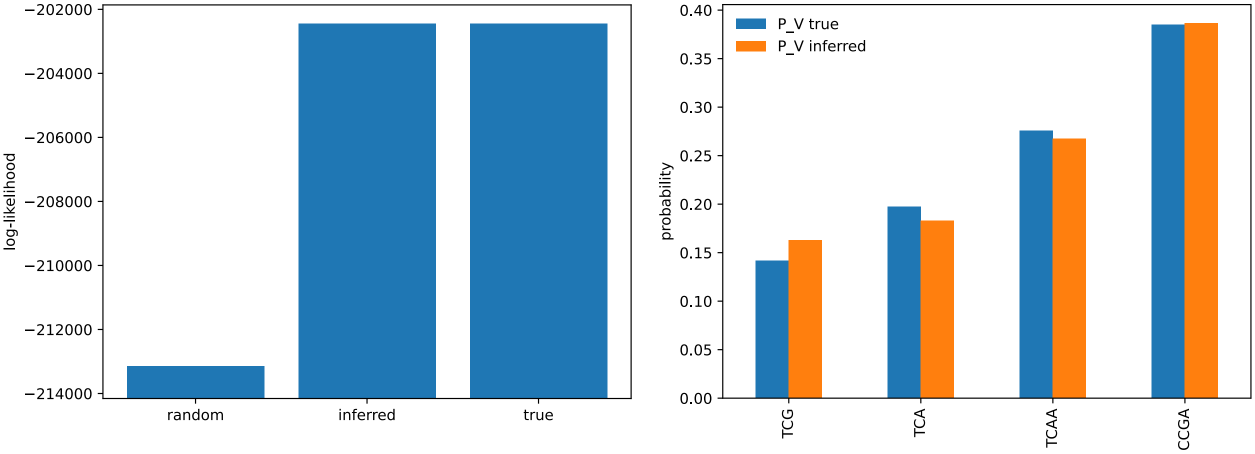


Figure 6: Left panel: log-likelihood for three different sets of parameters. *Random* means uniform probabilities over the V genes and the deletion numbers, *inferred* are the result of the expectation maximization, *true* are the actual parameters use to generated the data. Right panel: comparison between the true and the inferred parameters of generating the four V genes.

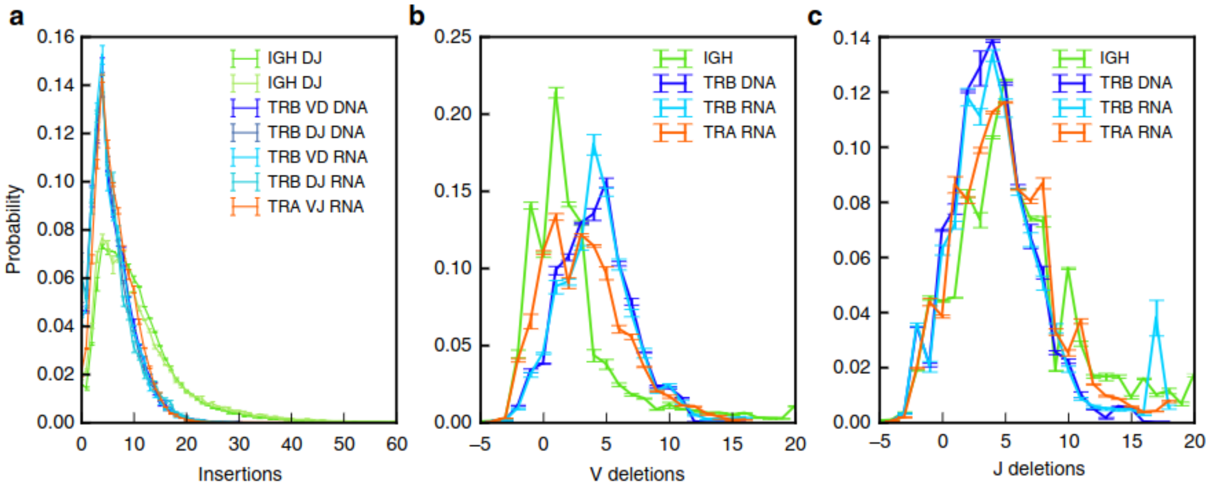


Figure 7: Inferred number of insertions and number of deletions for B and T cells of the real VDJ recombination. The different colors represent different type of receptors and using data for RNA or DNA sequencing Negative deletions correspond to palindromic insertions.

that have be taken into account when using these models. The first is the fact that our estimates are always subject to errors if the number of data-points is not infinite. As we discuss in the appendix 4.1, this can be taken into account using the Fisher information. The second potential problem is that our optimization algorithm is not perfect. We know that our algorithm converges to a maximum of the likelihood, but it is not guaranteed that it is the **global maximum**. Actually, this particular problem suffers from the potential presence of "flat" maxima, i.e. regions of the likelihood that attain the same maximum values for several values of the parameters. In other words, the problem is indeterminate. Both this problems are very relevant in inference problems, but their detailed discussion goes beyond the scope of these lectures.

3.1.2 Generation probabilities in humans

To solve the inference of the real VDJ recombination one basically follows the same steps that we saw for the toy model. However, the problem is much more complex and many more problems appear on the way, especially in the maximization algorithm. Just to have an idea, below we write the counterpart of equation Eq.3, i.e. the probability of

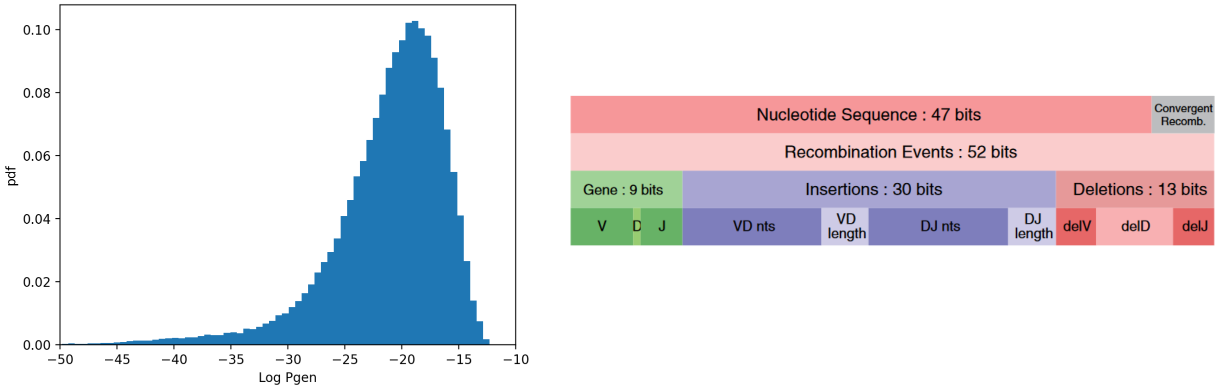


Figure 8: Left panel: distribution of the generation probability of all the sequences in the dataset. Right panel: Shannon diversity of the VDJ recombination.

generating the scenario with all the moves illustrated in Fig. 4.

$$\begin{aligned} P(s|\theta) = &P(V)P(J,V)* && \text{Prob of sampling V, D and J genes} \\ &P(\text{del}|V)P(\text{right del, left del}|D)P(\text{del}|J)* && \text{Deleting probabilities} \\ &P(\text{ins VD})\prod_{i=1}^{\text{ins VD}} P(n_i|n_{i-1})P(\text{ins DJ})\prod_{i=1}^{\text{ins DJ}} P(n_i|n_{i-1}) && \text{Inserting probabilities} \end{aligned} \tag{6}$$

Note that this writing is assuming several hypothesis on the biological process. For example, the V-gene choice is uncorrelated from the D and J choice, which, instead, depends on each other. Another assumption is that insertions and deletions not correlated. By assuming more un-correlated moves the number of free parameters decreases, implying an easier computation. However, all this choices have to be tested *a posteriori*.

The process of Eq. 6 has more than 10^4 parameters to fit, therefore its complexity is order of magnitude larger than our toy model. A big additional difficulty is that in our case we can enumerate all the possible sequences, while in the real case they are practically infinite and this makes the numerical computation much harder.

To finish with, the real model needs to deal with possible sequencing errors that experiments can generate. Those errors can be encoded in the following probability of having a sequence x given a scenario s:

$$P_{err}(x|s,\theta) = (r/3)^{n_{err}}\,(1-r)^{L-n_{err}}, \tag{7}$$

where r is the probability of sequencing error of a single nucleotide, n_{err} is the number of mis-matches between the sequence generated by s and x, L the length of the sequence. As a consequence, the probability of a sequence is not simply a summation over the possible scenarios as in Eq. 4, but becomes:

$$P(x,\theta) = \sum_s P(x,s|\theta) = \sum_s P_{err}(x|s,\theta)P(s|\theta), \tag{8}$$

where $P(s|\theta)$ is given by 6. Despite its complexity the model can still be solved by the expectation maximization algorithm.

The key result of this analysis is that inference starting from receptors of different people leads to almost identical results, showing that this description of the VDJ recombination seems to be universal across people. The model provides all the probabilities of all the possible events encoded by the model and, for example, one can look at the probabilities of inserting or deleting a given number of nucleotides Fig. 7. The precise information of all these rates greatly improve the knowledge of the underlying biological process.

The model allows us to inspect the diversity of generated receptors. The concept of diversity of samples generated from stochastic processes is discussed in the Appendix 4.3. Among the possible different measures, here we consider the Shannon diversity, whose values are shown in Fig. 8, right panel. The total diversity of the process is around 52 bits, which are effectively 47 because several recombination events, i.e. scenarios, lead to the same sequence. We can interpret this number as the bits generated by a uniform distribution having $2^{47} \approx 10^{15}$ outcomes. Among those, the majority is generated by the insertion events.

Probably, the most important result is the prediction of the receptor generation probability. Once the model parameters are learned with a given dataset and the universality of the model is validated, one can use the inferred parameters to compute how much it is probable that one specific receptor can be generated (even if not contained in the studied dataset). This opens the possibility to quickly know how likely it is to create any receptor. We call this probability **generation probability** or P_{gen}. One first important information that one gets after computing P_{gen} of many sequences is the type of plot in Fig. 8. It is important to notice is that this distribution spans a huge domain of orders of magnitude. In a typical sample, one can find extremely rare receptors, with $P_{gen} \sim 10^{-30}$ as well as common ones, with $P_{gen} \sim 10^{-15}$ and the two extremes are separated by a several orders of magnitude. This implies that "no all the clonotypes are equal": the presence of a common receptor can be due just by chance, while a rare one, potentially, is more informative and related to encounters of diseases (that led the receptor to expand). This idea is at the basis of the next paragraph, where we will try to infer receptors that interacted with a disease from sequencing data.

3.2 Inferring expanded clonotypes from a single sample

3.2.1 Basic idea of the algorithm

Let us consider the following situation: we have the receptor repertoire taken in a patient that has been vaccinated against flu a couple of weeks before. The peak of expansion happens after a few weeks from the stimulus, therefore we expect that our sample contains the receptors that interacted with the vaccine. From the list of receptors/clonotypes, which looks like the one in Fig 3, we want to infer which are the ones that have been triggered by the response, and we want to do that just by looking at the table statistics, without any a priori biological information.

This problem is quite non-trivial, since, as we saw before, several clonotypes can be there in large numbers just because they are much easier to produce. The approach that has been employed in a recent work [11] is based on the observation that when the immune system responds to a stimulus, it is not one single clonotype that expand, but they respond in groups and the members of those groups are typically very similar to each other. This is due to the fact that clonotypes with similar sequences have also a similar structure and can bind with similar good affinities on the pathogen. We expect therefore to find in our data clusters of similar receptors that are responding together to the vaccine.

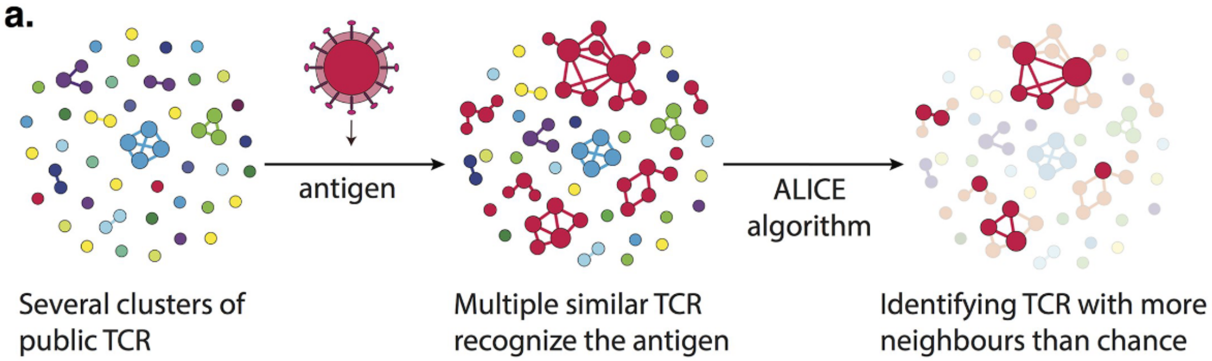

a.
antigen
ALICE algorithm
Several clusters of public TCR
Multiple similar TCR recognize the antigen
Identifying TCR with more neighbours than chance

Figure 9: Sketch of the algorithm that aims to identify clusters of expanded clonotypes in response to an infection. The algorithm is called Antigen-specific Lymphocyte Identification by Clustering of Expanded sequences (ALICE) [11]

To implement this idea, it is useful to represent this system as a network of similarities. Specifically we define a distance between two receptors as a **Hamming distance** between the amino-acid sequences. This is zero if the clonotypes have different lengths, otherwise it is the number of mis-matches between amino-acids. For example, $MTGH$ and $ATFH$ have distance 2, $MTGH$ and $MTGHA$ distance 0 because they have different lengths. The reason for working with the amino-acid sequence and not the nucleotides is that it contains the information about the shape of the receptors, which is what ultimately matters in the binding process. Different nucleotide sequences can have the same translation into amino-acid sequence, and therefore they can be considered functionally equivalent.

To build the similarity network we use the sequences as nodes and we draw a link when two sequences have distance 1. As Fig. 9 shows, one can expect that when an antigen interact with the system, if one receptor responds to the antigen contained in the vaccine (or virus), also other receptors with similar sequences respond. Based on this observation, the key idea of the algorithm is that, after a stimulus like a vaccine, I can look for the receptors whose number of nearest neighbours in this network is "bigger than expected". But what is bigger than expected? To evaluate a statistical significance of this observation, we need to build a "null" model for the number of nearest neighbours, which quantifies how many of them can we find just by chance.

3.2.2 Null model for the number of neighbours

The key object that we want to compute here is the probability that a sequence x has d nearest neighbours, defined as sequences that differ of only 1 amino-acid. This probability is "null" in the sense that does not contain information on specific infections, but just on the neutral generation of receptors, that we know is quantified by $P_{gen}(x)$. The appendix 4.4 provides an explicit derivation of this probability, which is a Poissonian distribution:

$$P(d|x) = \frac{e^{-\lambda}\lambda^d}{d!} \qquad \text{where } \lambda = NQ \sum_{x' \in \mathcal{N}(x)} P_{gen}(x') \tag{9}$$

where N is the number of sequences in the dataset , $\mathcal{N}(x)$ is the set of nearest neighbours of x and Q is a factor correcting for the thymic selection of the receptor. We did not discuss about thymic selection in this lectures which, in short, is a quality check that all the randomly generated receptors have to pass after their creation. This check kills all the receptors that can potentially bind to proteins of the body and trigger auto-immune diseases. To take into account that the sequences we find in experiments have passed this selection, the effective generation probability has to be corrected by a factor that can be considered approximately constant: $Q \approx 9.41$.

3.2.3 Statistical enrichment of the expanded clonotypes

The two notebooks contained in the repository of the course, "ALICE_preprocess.ipynb" and "ALICE_enrichment.ipynb" show in practice the statistical procedure discussed in this chapter. The considered sample is a patient that has been vaccinated against yellow fever in the previous weeks, and the task is to find a small set of clonotypes/receptors that have more nearest neighbours than expected by chance. In particular, "ALICE_preprocess.ipynb" shows how to compute the generation probability P_{gen} using a bio-informatics tool: `https://github.com/statbiophys/OLGA`. Moreover it computes the observed number of neighbours of each sequence $d(x)$ and the expected number of neighbours λ given by Eq. 9.

"ALICE_enrichment.ipynb" performs the statistical computation of associating a p-value to each sequence. The p-value is the probability that the null scenario generates a number of neighbours equal or larger than the observed one:

$$p_v(x) = \sum_{d' > d(x)} P(d'|x).$$

A very small value of this $p_v(x)$ indicates that it is very unlikely to count in the neutral model of receptor generation the observed number of neighbours of data. This points to a receptors that is interacting with the vaccine and, together with it, several other receptors having a similar sequence.

However, we are dealing with a very big list of p-values, one for each receptor, and therefore we have to correct for multiple testing. The method used in the notebook is the Bonferroni correction which says that, given a significance level α, the effective significance that we have to consider is α/m, where m is the number of tests we are doing, i.e. the number of sequences that we are testing. The final step consists in taking the receptors that have $p_v(x) < \alpha/m$. Those are good candidates for interacting with the vaccine and can be a precious information for biologists and biotechnologists to develop new strategies in diagnostics and treatments.

3.3 Inferring expanded clonotypes from temporal samples

3.3.1 Basic idea of the algorithm

We saw in the previous paragraph how to find expanded clonotypes from single snapshots of the immune repertoire. The method was based on the idea that similar clonotypes expand together and, as a consequence, a receptor interacting with an antigen will be followed by several similar receptors undergoing clonal expansion. This implies that an experiment will sample with high probability those expanding clusters and, therefore, expanded clonotype are expected to have more neighbours (similar other clonotypes) in the dataset than expected by chance.

Now we imagine that we have at our disposal two samples taken at different time points, for example, before and after a strong stimulus of the immune system, like a vaccine. In such a case we can expect to see expansions of the interesting clonotypes. In other words, if at t_1 we have a sample $\mathbf{x}_1 = \{n_1^{(1)}, \dots, n_{U_1}^{(1)}\}$ and at t_2, $\mathbf{x}_2 = \{n_1^{(2)}, \dots, n_{U_2}^{(2)}\}$, we want to look for clonotypes that have $n_i^{(2)} > n_i^{(1)}$. This idea, in principle, makes sense, but we have to deal with

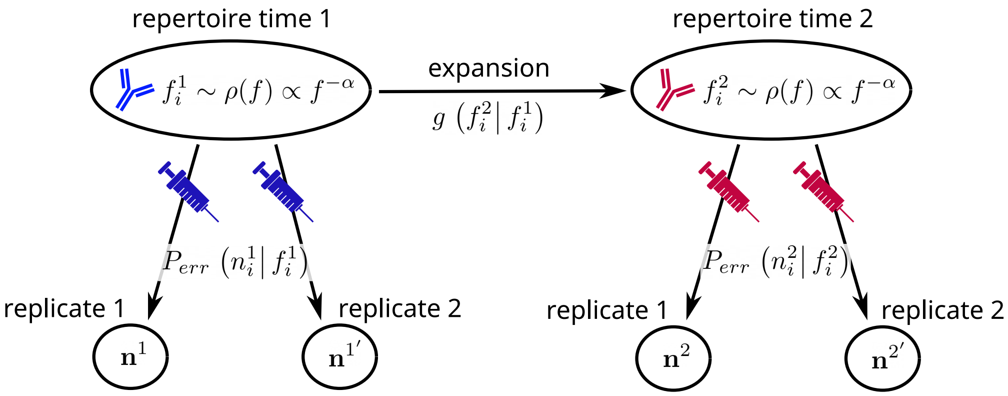

repertoire time 1
$f_i^1 \sim \rho(f) \propto f^{-\alpha}$
expansion
$g\left(f_i^2 | f_i^1\right)$
repertoire time 2
$f_i^2 \sim \rho(f) \propto f^{-\alpha}$
$P_{err}\left(n_i^1 | f_i^1\right)$
$P_{err}\left(n_i^2 | f_i^2\right)$
replicate 1
replicate 2
replicate 1
replicate 2
$\mathbf{n}^1$
$\mathbf{n}^{1'}$
$\mathbf{n}^2$
$\mathbf{n}^{2'}$

Figure 10: Setting for the inference of expanded clonotypes between time points. At each time point two independent experimental replicates are provided. Each experiments provides a set of clonotype counts that is noisy according to $P_{err}(n_f)$, where f is the true frequency of clonotypes that follows a power law distribution. Between time points some clonotypes can expand or contract. This is modeled by $g(f^1|f^2)$.

the noise that repertoire experiments have, like discussed in Sec. 2.2. Changes in numbers from one experiment to another can be caused by this sampling and experimental noise. This problem has been studied in a recent work using a maximum-likelihood approach [12] and a tool for inferring expanding clonotypes based on it, NoisET, has been released [13]. As shown bt Fig. 10, one wants first to estimate the experimental and sampling noise, P_{err}. This can be done running a maximum-likelihood inference using two independent experiments, or replicates, performed in the same conditions. This is the strength but also the weakness of this approach: it allows to rigorously take into account the specific source of noise in the experimental procedure. However, it requires that the experiment is run twice in the same patient at the same time point, and no may datasets are structured this way. The second step consists in running a second Bayesian inference on the clonotypes that can expand (or contract) between two samples at different time points, $g(f^2|f^1)$. This procedure includes the noise learned at the previous step and it is able to provide a list of clonotypes that are significantly expanded (or contracted).

3.3.2 Learning the noise

The key idea is to make a precise estimate of the probability of sampling a given number of clonotypes n, $P(n|\theta)$. This probability can be built by knowing that there is some experimental noise quantified by $P_{err}(n|f, \theta_{noise})$, where f is the true frequency of the clonotype in the repertoire. This noise has been discussed in Sec. 2.2, and it is typically assumed to be a Negative Binomial. In such a case this distribution has an average Mf, where M is the total number of sequences in the experimental sample. The variance of the Negative Binomial is defined by two parameters $\theta_{noise} = \{a, b\}$, $Var(n|f, \theta_{noise}) = Mf + a(Mf)^b$. Notice that for $a = 0$ one recovers the Poissonian distribution where the average is equal to the variance. The second ingredient one needs to compute $P(n|\theta)$ is the probability distribution of the true frequencies, $\rho(f|\theta_f)$. This object is, in principle, unknown, but experiments and some models point to the idea that it is a power law distribution $\rho(f|\theta_f) = cf^{-\alpha}$, for $f_m < f$. Here, the free parameters, $\theta_f = \{\alpha, f_m\}$ are the exponent, expected to be around 2 and the cutoff on the frequency, f_m, below which the distribution is zero. This minimal frequency is necessary for the convergence of the distribution. Note that for simplicity here we dropped the notation with the hat, e.g. $\hat{n}$ and $\hat{M}$ for referring to the samples like in Sec. 2.2. Here all the quantities refer to the samples and not the whole repertoire with the only exception of f. Putting all these ingredients together, the probability of sampling a given number of clonotypes has to be integrated over the frequency variable f:

$$P(n|\theta) = \int_{f_m}^{\infty} df \rho(f|\theta_f) P_{err}(n|f, \theta_{noise}). \tag{10}$$

We want to infer the noise parameters θ_{noise} and the frequency parameters θ_f. To this end, Eq. 10 is not enough because running the maximum likelihood machinery over it leads to an indeterminate form. To infer those parameters one needs two parallel independent experiment in the same condition, that we call replicates. The two experiments lead to two samples $\mathbf{x}$ and $\mathbf{x}'$. The probability that a given clonotype as count n and n' is

$$P(n, n'|\theta) \int_{f_m}^{\infty} df \rho(f|\theta_f) P_{err}(n|f, \theta_{noise}) P_{err}(n'|f, \theta_{noise}). \tag{11}$$

Given the two replicates, one can then write down the likelihood of generating them:

$$P(\mathbf{x}, \mathbf{x}'|\theta) = \prod_i P(n_i, n_i'|\theta),$$

where the index i runs over all the clonotypes present in both the experiments. This quantity actually does take into account all the clonotypes that can have zero counts in both the samples, i.e. $P(0,0|\theta)$ is in general a positive quantity. However, we do not have access to these clonotypes in experiments, therefore the actual probability that we have to maximize is $P(n, n'|n + n' > 0, \theta)$

$$P(\mathbf{x}, \mathbf{x}'|\theta) = \prod_i P(n_i, n_i'|n_i + n_i' > 0, \theta). \tag{12}$$

The maximization of this quantity leads to the best parameters that can generate the samples given the experimental noise and the underlying clonotype frequency distribution.

3.3.3 Inferring the expansion

We assume that the immune system is responding to a stimulus and, as a consequence, changing between the two time points. This change across the whole repertoire is described by a propagator $g(f_i^2|f_i^1)$. To model this process we say that only a fraction γ is responding to the stimulus and, therefore, expanding. All the other clonotypes keep the same frequency. In fact, since some clonotypes are increasing in frequency and the frequencies are normalized, we have a small contracting effect over all the receptors. It is convenient to re-write $f_i^2 = f_i^1 e^{s_i}$, which introduce the variable s that quantify the expansion rate. Our choice of g, written as a function of s, is the following:

$$g(s|\theta_{exp}) = \gamma \delta\left(s - s_0 - \bar{s}\right) + (1-\gamma)\delta\left(s - s_0\right), \tag{13}$$

where $\delta(x)$ are delta functions. $s_0 < 0$ is the global negative contraction of all the clonotypes, while $\bar{s} > 0$ the expansion rate of the fraction γ that is responding. In general, one can choose more refined equations for the propagator, for example, by introducing some noise in the amount of expansion of the responding clonotypes.

As before, one has access only to the experimental samples and the true dynamics of g is hidden in the background. The inference over the parameters θ_{exp} has to be done with one sample for each time point. Actually there are four possible pairs $(\mathbf{n}^1, \mathbf{n}^2)$, $(\mathbf{n}^{1'}, \mathbf{n}^2)$, $(\mathbf{n}^1, \mathbf{n}^{2'})$, $(\mathbf{n}^{1'}, \mathbf{n}^{2'})$ (using the notation of Fig. 10). One can therefore run the inference 4 times and, for example, consider the consensus among the 4 inferences. For a single inference, the object to consider is the probability of observing a clonotype with a count n^1 at time 1 and n^2 at time 2:

$$P(n^1, n^2|\theta) = \int_{f_m}^{\infty} df \rho(f|\theta_f^*) P_{err}(n^1|f, \theta_{noise}^*) \int ds\; g(s|\theta_{exp}) P_{err}(n^2|fe^s, \theta_{noise}^*). \tag{14}$$

Notice that the parameters θ_{noise}^* and θ_f^* are the ones inferred at the previous step.

The maximization of the likelihood $P(\mathbf{n}^1, \mathbf{n}^2) = \prod_i P(n_i^1, n_i^2|n_i^1 + n_i^2 > 0, \theta)$ will lead to the estimate of $\theta_{exp}^* = (\gamma, s_0, \bar{s})$. The statistical analysis to identify the significantly expanded clonotypes has to compare the expansion rate found experimentally s_{obs} and the expansion rate that the our inferred model predict. This statistical test can be done in a "Bayesian" way, because our theory gives access to the following posterior of having a clonotype in our dataset that has expanded of a factor s given the counts n_i^1, n_i^2 observed in the experiment:

$$P(s|n^1, n^2, \theta^*) = g(s|\theta_{exp}^*) \int_{f_m}^{\infty} df \rho(f|\theta_f^*) P_{err}(n^1|f, \theta_{noise}^*) P_{err}(n^2|fe^s, \theta_{noise}^*). \tag{15}$$

Therefore, this object is saying the probability that the clonotype is undergone an expansion (or contraction) of rate s given its counts between two time points. A rate larger than zero implies an expansion, therefore $P(s > 0|n^1, n^2, \theta^*)$ is exactly the probability that the clonotype is expanding. Notice that this posterior is already taking into account the multiple testing across different clonotypes since the model was trained on the whole dataset.

The final output of this procedure leads to this probability for each clonotype in the dataset. By setting a threshold of confidence α on the probability of not expanding, $1 - P(s > 0|n^1, n^2, \theta^*)$, one can get the list of all the receptors potentially interacting with the stimulus.

4 Methods

4.1 The maximum likelihood framework

4.1.1 The general framework

The maximization of the likelihood is a standard framework to infer parameters of a stochastic model from data. Its employ is widespread in disciplines that study complex systems that have to deal with a large amount of data. Here we will see its application to the description of a toy model for the VDJ recombination, but we start with a brief general introduction.

We call $\mathbf{x} = \{x_1, \ldots, x_N\}$ result of a given experiment. For the moment we let x_i to be very general and, in principle, they can be just scalars, encode vectors of numbers or be more complex objects. In our VDJ example they will represent the receptor sequences sampled from a human. The likelihood maximization is based on the fact that one can write a model for the description of the process underlying the experiment that generated $\mathbf{x}$. This model depends on a list of parameters that we write as a vector θ. This parameters are, in general, unknown. To use the maximum-likelihood framework, a necessary condition is to be able to write the probability that our model, parametrized by θ, is able to reproduce the dataset $\mathbf{x}$. This probability is called likelihood:

$$P(\mathbf{x}|\theta) = \prod_{i=1}^{N} P(x_i|\theta). \tag{16}$$

In general we can assume that the experiment generates independent samples x_i, and this allows us to factorize the likelihood as a product over the single objects x_i. Very often, what people actually use is the logarithm of the likelihood, that has the advantage of transform the product over the independent probabilities in a summation. This is called log-likelihood:

$$\mathcal{L}(\mathbf{x}|\theta) = \log P(\mathbf{x}|\theta) = \sum_{i=1}^{N} \log P(x_i|\theta). \tag{17}$$

The objective is to find the parameters of the model that are able to describe the result of the experiment in the best possible way given our model This translate in finding the parameters that maximize the likelihood (or the log-likelihood, since they have the same stationary points):

$$\theta^* = \arg\max_{\theta} \mathcal{L}(\mathbf{x}|\theta) = \arg\max_{\theta} P(\mathbf{x}|\theta). \tag{18}$$

One has to be aware that the value θ^* is actually just an estimate of the "true" set of parameters that generated the data θ_{true}. However, there is an important result, the Cramér-Rao bound, that allows us to take under control this error. This states that, for a sufficiently large N, the estimate θ^* that we get is a Gaussian variable centered at the true value θ_{true} and with a given covariance matrix

$$\theta^* \sim \mathcal{N}\left(\theta_{true}, \frac{\mathcal{I}^{-1}}{N}, \right) = \mathcal{N}\left(\theta_{true}, \sigma_\theta^2, \right), \tag{19}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a multi-variate Gaussian distribution (a Gaussian distribution in more that 1 dimension) and we have defined $\sigma_\theta^2 = \mathcal{I}^{-1}/N$. What is important to notice is that the error on the estimate, i.e. the standard deviations of the Gaussian σ_θ, decreases as a square root on the number of samples, which guarantees a good convergence property. The covariance matrix can be found by inverting the following matrix, called **Fisher information**:

$$\mathcal{I}_{i,j} = -\sum_{i=1}^{N} P(x_i|\theta) \frac{\partial^2 \log P(x_i|\theta)}{\partial\theta_i \partial\theta_j}. \tag{20}$$

In general, when data and the models are very complex the likelihood can depend on many parameters. Therefore, one usually has to do with high dimensional likelihoods that are difficult to evaluate (even numerically) and, in turn, difficult to maximize. But if those technical difficulties can be overcame, this framework has many advantages. It is very general and applicable to a potentially infinite number of cases, making easy to bridge different disciplines and export technical tools between them. It is usually interpretable, since the model underlying the likelihood is a physical model inspired by the system under study. It allows to generalize to new data points generated by other experiments (if one assumes that experiments are reproducible). We will see, for example, that it will allows us to compute the generation probabilities of all the possible lymphocyte receptors and antibodies that can be created, even if they have never been observed.

4.1.2 Model selection

All the framework described above is based on the fact that we have built a model that allows us to write down the likelihood. This model is written by exploiting intuitions and information about the system under study. For example, the model for the VDJ recombination encodes all the different biological steps that occur in the recombination process (gene sampling, deletions and insertions). As physicists, we are aware that models are just tools to interpret and predict real phenomena. In general, we have always to be opened to the possibility that better models can be found, that provide more accurate predictions and easier interpretations of the phenomena. Using the Bayesian-likelihood framework we can better quantify the concept of "better models" using **model selection**.

There is no a unique theory for model selection, but all the different approaches say that a better model has to provide better descriptions of the data, i.e. maximising the likelihood, and, at the same time, be as simple as possible, i.e. minimizing the number of parameters or the model "complexity". This create a trade-off: one would expect that increasing the number of parameters the model would increase the likelihood. However, more complex models have the risk of overfitting the data and they are, in general, less interpretable. Where to choose the sweet-spot of this trade-off can be different for different model selection theories.

Here we just state one of the most popular: the **Bayes Information Criterion**. We call $\mathcal{L}(\mathbf{x}|\mathcal{M},\theta)$ the log-likelihood of a dataset $\mathbf{x}$ of N elements, described by the model $\mathcal{M}$. Each model has a different set of parameters θ, that can be also of different number, $|\theta|$. The criterion states that the model to choose is the one that minimize the following quantity:

$$BIC = |\theta| \log N - 2\mathcal{L}(\mathbf{x}|\mathcal{M},\theta^*), \tag{21}$$

where the first term is the "Occam's razor" term that favors the simplicity of a model and the second is the log-likelihood evaluated at the best parameter estimate. This quantity can be derived from the maximization of $P(\mathbf{x}|\mathcal{M})$ using a Bayesian approach and under the assumption that $N \gg |\theta|$. This second assumption says that deep-learning models are typically excluded from the framework.

4.1.3 Recovering the average and standard deviation estimators from maximum likelihood

Let us assume that we have N measurements of a given physical quantity: $\mathbf{x} = \{x_i \dots, x_N\}$. Our probabilistic model for the process generating the measures is a Gaussian distribution $P(x_i,\theta) = \mathcal{N}(\mu,\sigma^2)$ (as the central limit theorem would suggest if the measurements are independent and with an "identical" process). As an exercise, we can apply the maximum likelihood framework to this problem. The unknown parameters are the average μ and the standard deviation σ, $\theta = \{\mu,\sigma\}$. The log-likelihood is then

$$\log P(\mathbf{x}|\theta) = \sum_{i=0}^{N} \log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]\right) = -N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=0}^{N}(x_i-\mu)^2. \tag{22}$$

If we maximize this function over μ, we can obtain the best estimate for the average of our points. We can do this by finding the stationary points of the equation above, $\partial_\mu \log P(\mathbf{x}|\theta^*) = 0$. This leads to

$$-\frac{1}{2\sigma^2}\sum_{i=0}^{N}\partial_\mu(x_i-\mu)^2|_{\mu^*} = \frac{1}{\sigma^2}\sum_{i=0}^{N}(x_i-\mu^*) = \frac{1}{\sigma^2}\left(\sum_{i=0}^{N}x_i - N\mu^*\right) = 0,$$

therefore

$$\mu^* = \frac{1}{N}\sum_{i=0}^{N}x_i, \tag{23}$$

which, surprise, is nothing else than the usual empirical average used in statistics. If we maximize over σ we get:

$$\sigma^* = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\mu^*-x_i)^2}, \tag{24}$$

which is the classical estimator for the variance. So, in this trivial example, we are seeing that the classical statistical estimator can be derived by the maximum likelihood framework.

As a final exercise, we can try to estimate the error of the average estimator μ^* using the Cramér-Rao bound. For simplicity let us assume that we know the true σ, that simplify the setting and allows us to consider the Gaussian in

Eq. 19 as a simple one dimensional distribution depending only on μ. It is centered in the true value μ_{true} and the we have to compute its standard deviation for having an estimate of the error. To get it, we need the Fisher information:

$$\mathcal{I}_\mu = -\sum_{i=1}^{N} P(x_i|\theta) \frac{\partial^2 \log P(x_i|\theta)}{\partial \mu^2} = \sum_{i=1}^{N} P(x_i|\theta) \frac{1}{\sigma^2} = \frac{1}{\sigma^2},$$

where in the second equality we used the normalization of the $P(x_i|\theta)$. This implies that the standard deviation on the estimate of μ^* is

$$\sigma_\mu = \sqrt{\frac{\mathcal{I}_\mu^{-1}}{N}} = \frac{\sigma}{\sqrt{N}}. \tag{25}$$

Again, we are finding a well-known result of classical statistics which is the standard error of the mean.

4.1.4 Recovering the least-square method

A simple exercise to get more familiar with this tool is to imagine that we have a list of scalars y_i obtained at a given time point t_i which compose our dataset $x_i = (y_i, t_i)$. We assume that those data y_i are generated by a linear function $y = \theta_1 + \theta_2 t_i$ on the top of which we have Gaussian noise. We can encode this in the following probability

$$P(x_i|\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y_i - \theta_1 - \theta_2 t_i)^2}{2\sigma^2}\right], \tag{26}$$

That leads to the average of our observable y as $\langle y_i \rangle = \theta_1 + \theta_2 t_i$, and a standard deviation of σ. If we try to write down the log-likelihood of the model we get

$$\mathcal{L}(\mathbf{x}|\theta) = \sum_{i=1}^{N} -\frac{1}{2}\log(2\pi\sigma) - \frac{(y_i - \theta_1 - \theta_2 t_i)^2}{2\sigma^2} = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{\sum_i (y_i - \theta_1 - \theta_2 t_i)^2}{2\sigma^2}. \tag{27}$$

The parameters that we have to choose are the ones that maximize the object above, but this object attains its maximum when $\sum_i (y_i - \theta_1 - \theta_2 t_i)^2$ (all the other parts of the expression are constant in the parameters θ), which corresponds to the mean square error used in the classical least-square method.

4.1.5 Estimating frequencies from a set of outcomes

Let us consider another very simple (almost-trivial) example that goes more in the direction of our VDJ exercise. We have a set of outcomes, say a specific nucleotide in a specific position of the DNA, $\{A, T, C, G\}$, that can be generated with some probabilities: $p_A, p_T, p_C, p_G = 1 - p_A - p_T - p_C$. These three independent probabilities are our unknown parameters, $\theta = \{p_A, p_T, p_C\}$, and the experiment consists in N observations of those letters: $\mathbf{x} = \{T, A, A, G, \ldots\}$. Effectively, this set can be equivalently described as the counts of each outcome $\mathbf{x} = \{n_A, n_T, n_C, n_G\}$. The probability of generating the observations according to our model is

$$P(\mathbf{x}|\theta) = p_A^{n_A} p_T^{n_T} p_C^{n_C} (1 - p_A - p_T - p_C)^{n_G} \tag{28}$$

We can then compute the maximization of the likelihood (or the log-likelihood) with respect the model parameters. For p_A the stationary point condition, $\partial_{p_A}\mathcal{L}(\mathbf{x}|\theta) = 0$ is

$$n_A(1 - p_A^* - p_T^* - p_C^*) = p_A^* n_G.$$

By computing the same relation with respect the other nucleotides and summing together the four relations, one gets

$$(n_A + n_T + n_C + n_G)(1 - p_A^* - p_T^* - p_C^*) = n_G.$$

This expression can be substituted in the one above, obtaining the best estimate for the probability of an outcome

$$p_A^* = \frac{n_A}{n_A + n_T + n_C + n_G}. \tag{29}$$

This is actually a trivial result: the best we can do for estimating a probability is its frequency, but shows us one more time that the maximum likelihood framework can be applied in a very different context (with respect the previous examples).

4.2 Expectation-Maximization for the maximum likelihood

4.2.1 The general algorithm

The likelihood or our model will be maximized using a classical algorithm called **expectation-maximization**, EM. We start with a general abstract statement of the algorithm, proving that it is indeed leading to the maximization of the likelihood. In the next section we will apply it to our exercise where it will become much easier to interpret.

The algorithm comes in handy when the likelihood depends on latent or hidden variables $\mathbf{z}$: $P(\mathbf{x}, \mathbf{z}|\theta)$. This variables are necessary to generate the data, but they cannot be accessed from the experiment, i.e. $\mathbf{x}$ does not contain information on $\mathbf{z}$. Therefore the likelihood that we want to maximize has to be marginalized over those variables

$$P(\mathbf{x}|\theta) = \sum_{\mathbf{z}} P(\mathbf{x}, \mathbf{z}|\theta). \tag{30}$$

In our VDJ exercise, those hidden variables are the scenarios.

The EM algorithm defines a **pseudo-likelihood** that gives a proxy of how well I am fitting the data with a set of parameters θ' given a fixed other set of parameters θ which are used to evaluate the latent variables:

$$Q(\mathbf{x}|\theta', \theta) = \sum_{i=1}^{N} \sum_{\mathbf{z}} P(\mathbf{z}|x_i, \theta) \log P(x_i, \mathbf{z}|\theta'). \tag{31}$$

It can interpreted as the average the log-likelihood evaluated at θ' over the latent variables evaluated at θ. This object can be a bit scary at first, but it is, in general, easier to maximize that the real likelihood itself. Later we will prove that maximizing Q over θ' is the same as finding a likelihood $\mathcal{L}(\mathbf{x}|\theta')$ larger than $\mathcal{L}(\mathbf{x}|\theta)$.

The EM algorithm starts from an arbitrary choice of initial guesses of the parameters $\theta^{(0)}$. Then one iterates over $t = 1, \ldots$ two steps:

- **Expectation step**: compute the pseudo-likelihood $Q(\mathbf{x}|\theta, \theta^{(t-1)})$.
- **Maximization step**: the new set of parameters $\theta^{(t)}$ are obtained bu maximizing the pseudo-likelihood over θ:
$$\theta^{(t)} = \arg\max_{\theta} Q(\mathbf{x}|\theta, \theta^{(t-1)}) \tag{32}$$

The sequence $\theta^{(t)}$ will eventually reach a fixed point that is a maximum of the likelihood and our good set of parameters.

4.2.2 Proof of the maximization step

One thing we have prove to trust the algorithm is that the maximization step is going towards a maximum of the likelihood. and that the new parameter θ' is indeed better than the old θ. We start to write the log-likelihood of θ'

$$\mathcal{L}(\mathbf{x}|\theta') = \sum_{i=1}^{N} \log P(x_i|\theta') = \sum_{i=1}^{N} \log P(x_i, \mathbf{z}|\theta') - \log P(\mathbf{z}|x_i, \theta').$$

In the second equation we have substituted $P(x_i|\theta')$ using the definition of conditional probability over the latent variables: $P(x_i, \mathbf{z}|\theta') = P(\mathbf{z}|x_i, \theta')P(x_i|\theta')$.

The next step is to sum over $\mathbf{z}$ and multiply by $P(\mathbf{z}|x_i, \theta)$ both sides of the equation.

$$\sum_{\mathbf{z}} P(\mathbf{z}|x_i, \theta)\mathcal{L}(\mathbf{x}|\theta') = \sum_{\mathbf{z}} \sum_{i=1}^{N} P(\mathbf{z}|x_i, \theta) \log P(x_i, \mathbf{z}|\theta') - \log P(\mathbf{z}|x_i, \theta').$$

One can notice that $\mathcal{L}(\mathbf{x}|\theta')$ does not depend on the hidden variables and can be moved outside the summation which, thanks to the normalization sums to 1. This leads to

$$\mathcal{L}(\mathbf{x}|\theta') = \sum_{i=1}^{N} \sum_{\mathbf{z}} P(\mathbf{z}|x_i, \theta) \left(\log P(x_i, \mathbf{z}|\theta') - \log P(\mathbf{z}|x_i, \theta')\right).$$

Looking at the right term, one can recognize the definition of $Q(\mathbf{x}|\theta', \theta)$, Eq. 31.

$$\mathcal{L}(\mathbf{x}|\theta') = Q(\mathbf{x}|\theta', \theta) - \sum_{i=1}^{N} \sum_{\mathbf{z}} P(\mathbf{z}|x_i, \theta) \log P(\mathbf{z}|x_i, \theta').$$

The next step is to consider the likelihood difference between the parameters θ and θ':

$$\begin{aligned}\mathcal{L}(\mathbf{x}|\theta') - \mathcal{L}(\mathbf{x}|\theta) =& Q(\mathbf{x}|\theta',\theta) - Q(\mathbf{x}|\theta,\theta) - \sum_{i=1}^{N}\sum_{\mathbf{z}} P(\mathbf{z}|x_i,\theta)\log P(\mathbf{z}|x_i,\theta') + \sum_{i=1}^{N}\sum_{\mathbf{z}} P(\mathbf{z}|x_i,\theta)\log P(\mathbf{z}|x_i,\theta)\\ & Q(\mathbf{x}|\theta',\theta) - Q(\mathbf{x}|\theta,\theta) + \sum_{i=1}^{N}\sum_{\mathbf{z}} P(\mathbf{z}|x_i,\theta)\log \frac{P(\mathbf{z}|x_i,\theta)}{P(\mathbf{z}|x_i,\theta')}\end{aligned}$$

The term with the summations can be proven to be always positive. Indeed one can recognize the Kullback-Liebler divergence between $P(\mathbf{z}|x_i,\theta)$ and $P(\mathbf{z}|x_i,\theta')$, which is always a positive quantity.

$$\sum_{\mathbf{z}} P(\mathbf{z}|x_i,\theta)\log\frac{P(\mathbf{z}|x_i,\theta)}{P(\mathbf{z}|x_i,\theta')} = D_{KL}(P(\mathbf{z}|x_i,\theta),(\mathbf{z}|x_i,\theta')) \geq 0.$$

This can be, in turn, proven using the *Gibbs inequality*, which uses the fact that the, for $x > 0$, $\log x \leq x - 1$:

$$\sum_i p_i \log\frac{p_i}{q_i} = -\sum_i p_i \log\frac{q_i}{p_i} \geq -\sum_i p_i\left(\frac{q_i}{p_i} - 1\right) = -\sum_i q_i + \sum_i p_i = 1 - 1 = 0.$$

Therefore we have

$$\mathcal{L}(\mathbf{x}|\theta') - \mathcal{L}(\mathbf{x}|\theta) \geq Q(\mathbf{x}|\theta',\theta) - Q(\mathbf{x}|\theta,\theta),$$

which implies that a value of θ' that lead to a positive increment, also imply a positive increment in the log-likelihood. In other words, the maximization step of the EM algorithm leads to a positive increment in the original log-likelihood.

4.2.3 Application to 1-d gaussian mixtures

4.2.4 Expectation-Maximization for the VDJ model

We now try to apply the general EM scheme to our problem. We start with writing down the pseudo-likelihood, Eq. 31 where, as mentioned before, the latent variables are identified with the scenarios that can generate a sequence.

$$Q(\mathbf{x}|\theta',\theta) = \sum_{i=1}^{N}\sum_{s} P(s|x_i,\theta)\log P(x_i,s|\theta'). \tag{33}$$

One first simplification comes from expressing the probability in the logarithm (a joint probability of x_i and s) using the definition of conditional probability: $P(x_i,s|\theta') = P(x_i|s,\theta')P(s|\theta')$. We can also use the Bayes theorem for the probability outside the logarithm, obtaining:

$$Q(\mathbf{x}|\theta',\theta) = \sum_{i=1}^{N}\sum_{s} \frac{P(x_i|s,\theta)P(s|\theta)}{P(x_i|\theta)}\left(\log P(x_i|s,\theta') + \log P(s|\theta')\right). \tag{34}$$

We now turn our attention to $P(x_i|s,\theta)$ outside the logarithm. This is the probability that, given a scenario s, my model generate a sequence x_i. By knowing that a scenario can generate only one probability, this is actually a simple quantity that is 1 if the sequence can be actually generated by s, i.e. $s \in \mathcal{S}(x_i)$, and 0 otherwise. This basically cancels all the terms of the summation for which the scenario is cannot reproduce x_i, and we can restrict the summation terms only to $s \in \mathcal{S}(x_i)$:

$$Q(\mathbf{x}|\theta',\theta) = \sum_{i=1}^{N}\sum_{s\in\mathcal{S}(x_i)} \frac{P(s|\theta)}{P(x_i|\theta)}\left(\log P(x_i|s,\theta') + \log P(s|\theta')\right). \tag{35}$$

Moreover, we can notice that the same probability in the logarithm, $P(x_i|s,\theta')$, is exactly the same object we discussed before, which is 1 (and its logarithm 0) for all the terms of the summation. Therefore it disappears

$$Q(\mathbf{x}|\theta',\theta) = \sum_{i=1}^{N}\sum_{s\in\mathcal{S}(x_i)} \frac{P(s|\theta)}{P(x_i|\theta)}\log P(s|\theta'). \tag{36}$$

We can also explicitly express the scenarios as couples of V gene and number of deletions, $s = (V_i, k)$, as well as the scenario probabilities as $P(s|\theta) = P(V_i)P(k|V_i)$. This leads to

$$Q(\mathbf{x}|\theta',\theta) = \sum_{i=1}^{N}\sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_i)P(k|V_j)}{P(x_i|\theta)}\log P'(V_j)P'(k|V_j). \tag{37}$$

By using Eq. 5 for the probability of generating a sequence $P(x_i|\theta)$, we can write down the pseudo-likelihood as a direct function of all our parameters.

Our objective now is to find what are the values of θ', namely all the $P'(V_i)$ and $P'(k|V_i)$, that maximize the pseudo-likelihood. To do so we can find the stationary points of Q with respect to those parameters. However, this operation has to take into account that we cannot choose every value for those P' since we have to constraint at the same time the fact that they have to be normalized. A constrained optimization is done using the technique of **Lagrangian multipliers**, where the objective function to optimize is extended to be

$$\hat{Q}(\mathbf{x}|\theta', \theta) = Q(\mathbf{x}|\theta', \theta) + \lambda_0 \left(1 - \sum_i P'(V_i)\right) + \sum_i \lambda_i \left(1 - \sum_k P'(k|V_i)\right). \tag{38}$$

Here new arbitrary constants, the Lagrangian multipliers, λ_0 and $\{\lambda_i\}_{i=1,2,3,4}$, are introduced and multiply the normalization constraints for all the probabilities we have.

Let us do the explicit derivative of the Lagrangian function above just for the V gene parameters $P'(V_i)$ and the set it equal to zero to find its maximum. The derivative for the deletion probabilities $P'(k|V_i)$ does not present further difficulties and it is left as exercise. We have

$$\frac{\partial \hat{Q}(\mathbf{x}|\theta', \theta)}{\partial P'(V_i)} = \frac{\partial Q(\mathbf{x}|\theta', \theta)}{\partial P'(V_i)} - \lambda_0 \sum_j \frac{\partial P'(V_j)}{\partial P'(V_i)} = \frac{\partial Q(\mathbf{x}|\theta', \theta)}{\partial P'(V_i)} - \lambda_0 = 0, \tag{39}$$

where the derivative on the Lagrangian constraints is zero for all the terms of deletions, i.e. $\{\lambda_i\}_{i=1,2,3,4}$. The derivative over the constraint λ_0 leads to a Kronecker delta inside the summation which then leads to 1. We now consider just the derivative over the pseudo-likelihood using Eq. 37. The derivative filters across the first term of the summation which depends on the older parameters $P(V_i)$ and acts only on the logarithm part

$$\begin{aligned} \frac{\partial Q(\mathbf{x}|\theta', \theta)}{\partial P'(V_l)} =& \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \frac{1}{\partial P'(V_l)} \log P'(V_j)P'(k|V_j) \\ & \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \frac{P'(k|V_j)\delta_{l,j}}{P'(V_j)P'(k|V_j)} \\ & \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \frac{\delta_{l,j}}{P'(V_j)}\Bigg|_{P'^*(V_j)} = 0. \end{aligned} \tag{40}$$

By substituting this last equation in Eq. 39 and moving $P'(V_j)$ out the summation, we have

$$P'^*(V_l)\lambda_0 = \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \delta_{l,j}. \tag{41}$$

The last step is to fix the Lagrangian constraint with the normalization $\sum_i P'(V_i) = 1$. It is quick to see that by summing the equation above over j has the effect of making the Kronecker delta disappears:

$$\sum_j P'^*(V_l)\lambda_0 = \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)}. \tag{42}$$

From that expression one can also see that summation of the numerator is exactly the sequence probability, Eq. 4, leading to $\lambda_0 = N$. Therefore we finally have

$$\begin{aligned} P'^*(V_l) =& \frac{1}{N} \sum_{i=1}^{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \delta_{l,j} \\ & \sum_x \frac{n_x}{N} \sum_{(V_j,k)\in\mathcal{S}(x_i)} \frac{P(V_j)P(k|V_j)}{P(x_i|\theta)} \delta_{l,j}. \end{aligned} \tag{43}$$

In the last equation we just express the summation over all the sequence in our dataset as a summation over all the unique sequences x, each one with a given number of occurrences n_x. This is done for computational reason since the summation in this way has much less terms. We also need the update equation for the deletion probability which can be obtained with similar calculations:

$$P'^{*}(h|V_i) = \frac{1}{P'^{*}(V_i)} \sum_x \frac{n_x}{N} \sum_{(V_j,l)\in\mathcal{S}(x)} \delta_{i,j}\delta_{h,k} \frac{P((V_i,k)|\theta)}{P(x|\theta)} \tag{44}$$

The expressions above is used in the notebook that solves the likelihood maximization in the VDJ toy example.

4.3 Diversity of stochastic samples

The concept of diversity is central in several complex systems. Probably the most famous case is the ecological diversity in ecosystems. Its quantification is based on the underlying stochastic process that generates the "components" of the system, such as species in ecology or receptors in immunology. We define this process over a set of components $i = 1, 2, \dots, k$ which possibly can be infinite. Each component can be generated with probability p_i. The diversity of this process can be defined by a general quantity called **Réiny entropy**:

$$H^q = \frac{1}{1-q} \log \left(\sum_i p_i^q \right), \tag{45}$$

which is parameterize by a parameter q. In some fields, it is more common to consider the **diversity index** $D^q = \exp H^q$. The parameter interpolates between different possible definitions of diversity. Possibly, the most natural concept is for $q = 0$, which leads to $H^q = \log k$, which is connected directly with the total number of different components in the system. $D^0 = k$ is called **richness**. However, this definition has some problems. First, it would not be well-defined for an infinite set. Second it considers all the components as equivalents. Often, this is not a desirable property, especially when the system is a "small" sample of the process that typically can contain only the most common components.

By increasing the value of q one starts to provide more weight to the more common components and solve the convergence of the entropy in the case of infinite components. For $q = 1$ one can take the limit of Eq. 46 and use de l'Hôpital to obtain the **Shannon entropy**:

$$H^1 = -\sum_i p_i \log(p_i). \tag{46}$$

This is a very common choice for a diversity measure and connects the concept of diversity with the concept of "average surprise" of an outcome that is typical of the Shannon entropy in information theory.

A final well known limit of the Réiny entropy is for $q = 2$, leading to $H^2 = \log(1/\lambda)$, where

$$\lambda = \sum_i p_i^2 \tag{47}$$

is called **Simpson index**. It has a straightforward interpretation as the probability that sampling twice from the distribution leads to two identical components.

4.4 Computation of the probability of the number of similar sequences

We want to compute the probability of a sequence x to have d nearest-neighbours, where a nearest-neighbour is a sequence having the same length of x and different of just one amino-acid. To this end, we have to consider all the potential neighbours of x. If the sequence length is 10 and the possible amino-acids are 20, there are $10 * 19$ possible neighbours, where for each position we can change the amino-acid in 19 possible ways (that are different from the starting one). In mathematical notation, we call $\mathcal{N}(x)$ the set of all possible neighbour sequences of x.

The first object we need is the probability of sampling a sequence x_i in an experiment that has N samples. We call the probability the *existence* probability of x_i, and can be written as

$$p_i^E = 1 - \text{Prob of never sampling } x_i = 1 - (1 - P_{gen}(x_i))^N. \tag{48}$$

We can also say that P_{gen} is very small, also much smaller that N, that leads to the simplification

$$p_i^E = 1 - \exp(N \log(1 - P_{gen}(x_i))) \approx 1 - \exp(-N P_{gen}(x_i)) \approx N P_{gen}(x_i). \tag{49}$$

To compute the probability of a given number of neighbours of a sequence, it is convenient to use the notation of stochastic variables. We introduce the variable Z_i that says if the sequence x_i exists or not in our sample:

$$Z_i = \begin{cases} 1 & \text{with prob. } p_i^E \\ 0 & \text{with prob. } 1 - p_i^E \end{cases} \tag{50}$$

This allows us to define the stochastic variable indicating number of neighbours of x_i, D_i, as the summation of the variables $Z_{i'}$ across all the possible nearest neighbours of x_i

$$D_i = \sum_{i' \in \mathcal{N}(x_i)} Z_{i'}. \tag{51}$$

We now have to look for the probability distribution of this variable. This is actually a quite ugly object, that is the sum of independent but not identically distributed variables, that takes the name of Poisson binomial distribution. There are no many analytical results for this distribution, but we can exploit the approximation of small P_{gen}. To do that, we have to use the generating function of a random variable X: $G_X(t) = \sum_x p(x)\exp(xt)$. This function can be written down and take a relatively easy shape because of the fact of a generating function of summations of independent variables is the product of the generating their functions:

$$\begin{aligned}
G_D(t) &= \sum_d p(d)\exp(dt) \\
&= \sum_{z_{i1}} \sum_{z_{i2}} \dots \sum_{z_{i|\mathcal{N}(x)|}} p(z_{i_1}, z_{i_2}, \dots, z_{i_{|\mathcal{N}(x)|}}) \exp\left(\sum_{i' \in \mathcal{N}(x)} z_i t\right) \\
&= \sum_{z_{i1}} \sum_{z_{i2}} \dots \sum_{z_{i|\mathcal{N}(x)|}} p(z_{i_1}) p(z_{i_2}) \dots p(z_{i_{|\mathcal{N}(x)|}}) \exp\left(\sum_{i' \in \mathcal{N}(x)} z_i t\right) \\
&= \sum_{z_{i1}} p(z_{i1}) e^{z_{i1} t} \sum_{z_{i2}} p(z_{i2}) e^{z_{i2} t} \dots \sum_{z_{i|\mathcal{N}(x)|}} p(z_{i|\mathcal{N}(x)|}) e^{z_{i|\mathcal{N}(x)|} t} \\
&= G_{Z_{i1}}(t) G_{Z_{i2}}(t) \dots G_{Z_{i|\mathcal{N}(x)|}}(t),
\end{aligned}$$

where in the second line we used the independence of the Z_i. By computing the generating function of Z, $G_Z(t) = p_i^E e^t + 1 - p_i^E$, we can write the generating function of the number of neighbours

$$G_{Y_i}(t) = \prod_{i' \in \mathcal{N}(x_i)} 1 - p_{i'}^E (1 - e^t) \approx \prod_{i' \in \mathcal{N}(x_i)} 1 - N P_{gen}(x_{i'})(1 - e^t), \tag{52}$$

where in the second expression we used 49. By assuming $NP_{gen}(x_i)$ small, the product can be expressed at the first order in this small quantity. This can be done by noticing that the generating function above is a product of factors like $1 + \epsilon_i$, where ϵ_i is a small quantity. The product of all these factors gives, at order 0, only the 1 that comes from the multiplication of all the ones of each factor. At order 1, the only way of having terms of order ϵ is to take the ϵ_i of a factor and multiply it for the ones of all the other factors. Therefore $\prod_i 1 - \epsilon_i = 1 + \sum_i \epsilon_i + O(\epsilon^2)$. You can convince yourself by trying a simple example with a few factors. In our case, this gives

$$G_{Y_i}(t) \approx 1 - \sum_{i' \in \mathcal{N}(x_i)} N P_{gen}(x_i')(1 - e^t),$$

The final step is to rewrite the expression always using the approximation $NP_{gen}(x_i) \ll 1$

$$G_{Y_i}(t) \approx 1 - N(1 - e^t) \sum_{i' \in \mathcal{N}(x_i)} P_{gen}(x_i') \approx \exp\left(-N(1 - e^t) \sum_{i' \in \mathcal{N}(x_i)} P_{gen}(x_i')\right) \equiv \exp\left(-\lambda(1 - e^t)\right),$$

where in the last expression we called $\lambda = N \sum_{i' \in \mathcal{N}(x_i)} P_{gen}(x_i')$. The obtained generating function is the same generating function of a Poisson distribution, indeed:

$$\begin{aligned}
G_{Poisson}(t) &= \sum_{n=0}^{\infty} \frac{e^{-\lambda}\lambda^n}{n!} e^{nt} \\
&= e^{-\lambda} \sum_{n=0}^{\infty} \frac{(e^t \lambda)^n}{n!}, \\
&= e^{-\lambda} e^{e^t \lambda} = \exp\left(-\lambda(1 - e^t)\right),
\end{aligned}$$

This allows us to conclude that the number of neighbours of a sequence follows a Poisson distribution:

$$P(d|x) = \frac{e^{-\lambda}\lambda^d}{d!} \qquad \text{where } \lambda = NQ \sum_{x' \in \mathcal{N}(x)} P_{gen}(x') \tag{53}$$

Notice that to obtain Eq. 9 we have to correct the generation probability with the selection factor Q, as discussed in the main text.

References

[1] Lauren M Sompayrac. *How the immune system works*. John Wiley & Sons, 2022.

[2] Peter C de Greef, Theres Oakes, Bram Gerritsen, Mazlina Ismail, James M Heather, Rutger Hermsen, Benjamin Chain, and Rob J de Boer. The naive t-cell receptor repertoire has an extremely broad distribution of clone sizes. *Elife*, 9:e49900, 2020.

[3] Jonathan Desponds, Andreas Mayer, Thierry Mora, and Aleksandra M Walczak. Population dynamics of immune repertoires. *Mathematical, computational and experimental T cell immunology*, pages 203–221, 2021.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[5] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[6] Thierry Mora and William Bialek. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144:268–302, 2011.

[7] Grégoire Altan-Bonnet, Thierry Mora, and Aleksandra M Walczak. Quantitative immunology for physicists. *Physics Reports*, 849:1–83, 2020.

[8] Anand Murugan, Thierry Mora, Aleksandra M Walczak, and Curtis G Callan Jr. Statistical inference of the generation probability of t-cell receptors from sequence repertoires. *Proceedings of the National Academy of Sciences*, 109(40):16161–16166, 2012.

[9] Yuval Elhanati, Zachary Sethna, Quentin Marcou, Curtis G Callan Jr, Thierry Mora, and Aleksandra M Walczak. Inferring processes underlying b-cell repertoire diversity. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1676):20140243, 2015.

[10] Quentin Marcou, Thierry Mora, and Aleksandra M Walczak. High-throughput immune repertoire analysis with igor. *Nature communications*, 9(1):561, 2018.

[11] Mikhail V Pogorelyy, Anastasia A Minervina, Mikhail Shugay, Dmitriy M Chudakov, Yuri B Lebedev, Thierry Mora, and Aleksandra M Walczak. Detecting t cell receptors involved in immune responses from single repertoire snapshots. *PLoS Biology*, 17(6):e3000314, 2019.

[12] Maximilian Puelma Touzel, Aleksandra M Walczak, and Thierry Mora. Inferring the immune response from repertoire sequencing. *PLOS Computational Biology*, 16(4):e1007873, 2020.

[13] Meriem Bensouda Koraichi, Maximilian Puelma Touzel, Andrea Mazzolini, Thierry Mora, and Aleksandra M Walczak. Noiset: Noise learning and expansion detection of t-cell receptors. *The Journal of Physical Chemistry A*, 126(40):7407–7414, 2022.