Mining virus genomes for host predictive signals

Francesca Young

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

Institute of Infection, Immunity and Inflammation
College of Medical, Veterinary and Life Sciences
University of Glasgow

January 2022

Abstract

The total dependence of a virus on its host for its survival leads to a fundamental entanglement with its host's cellular machinery. This drives a coevolutionary relationship that leaves an imprint of the host in viral genomes. The aim of this thesis was to develop machine learning approaches to identify and exploit these host predictive signals. We present methods that use these signals both to build classifiers that can assign putative information to virus genomes and to locate the discriminative features on viral proteins thereby identifying regions that are important in the host relationship. The first step aimed to identify discriminative features that capture the different aspects of the virus host relationship. We generated a range of feature sets from alternative representations of the viral genomes that each aimed to exploit the different levels of biological information present. We used a supervised machine learning approach to compare a range of feature sets for their ability to predict host taxonomic information. Next, we opened these "black box" classifiers and to extract the discriminative information learnt by the model to identify regions of a viral protein that are associated with their host relationship. We used the 'local' nature of some of the predictive feature sets to transform an amino acid sequence into host signals. Finally, we developed a multi-view generative mixture model, MVC, to tease apart the complex signals that are embedded in viral genomes via different evolutionary processes. This Bayesian approach uses the clustering of the data defined by labels of interest to guide the features associated with those labels into the "relevant view". The MVC model is able to identify features associated with weak effect in the data.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to thank my PhD supervisors David Robertson and Simon Rogers for their guidance and feedback throughout my PhD.

I am extremely grateful to the MRC and the MRC-University of Glasgow Centre for Virus Research (CVR) for giving me the opportunity to pursue a PhD.

I would also like to thank all members of the Robertson lab for being there: Joe, Haiting, Sej, Vandana, Spyros and Kieran. In particular I would like to remember Ben Stamp, who along with Joe and Anna made starting out on the PhD journey together fun.

A special thank you to Alex Pencheva for buddying me through the last two years via WhatsApp for the machine learning chat and the PhD rants.

Thank you to my family. Without the support and encouragement from Simon, my Simon as opposed to supervisor Simon, this would just not have happened. Thank you to, Rachael, Calum and Iain, whose faith in my ability to do this has been amazing throughout, but particularly in lockdown when they all returned home making it bearable. I am eternally grateful to my Mum and Dad for instilling a love of science and a sense of curiosity that has guided me back to research.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Francesca Young
January 2022

Chapter 1

Introduction

Viruses are inexorably entangled with their cellular hosts. The current COVID-19 pandemic has bought into sharp focus the threat that emerging viruses pose to society globally and has highlighted the need for increased surveillance for potential pathogenic viruses. In contrast, research over the last two decades has led to the discovery that viruses are a fundamental part of all earth's ecosystems and to be major drivers of the diversification of life on earth. Recent advances in viral metagenomics has led to an huge growth in viral discovery, increasing our understanding of the ubiquity and diversity of viruses and the ecosystems functions they perform within microbial communities to support life on earth. Knowing which host a virus infects is vital to our understanding of virus-host relationships and their role in microbiome ecology.

Fast, cheap sequencing technology has resulted to exponential growth in the number publicly held viral genomes. This is currently being illustrated by the number of SARS-CoV-2 genomes that have been sequenced. In November 2021 this passed the five million mark and by the beginning of January 2022 there were over six and half million SARS-CoV-2 genomes deposited in the GISAID database. There has been a similar growth in phage sequences from metagenomics experiments in which all DNA or RNA material in an environmental sample is sequenced. The IMG/VR viral resource database is the largest collection of microbial viral genomes containing over 2 million uncultivated viral genomes (UViGs) in December 2021 (Roux et al. 2021; Paez-Espino et al. 2019). Unfortunately, the nature of metagenomics means that the majority, $> 99\%$, of these viruses are not annotated with any information about their host. The absence of reliable high-throughput experimental methods to link a virus to its host means there is a need for computational

methods for the assignment of putative host taxonomic information.

As obligate intracellular parasites, viruses are intrinsically linked to their host being completely dependent on them for their survival. A virus must enter its host cells and divert the host's machinery to the replication and assembly of new viral particles. Over the long term, this antagonistic relationship links their evolutionary processes leaving an imprint of this shared history on both of their genomes. Despite the presence of this host imprint, it remains challenging to exploit the wealth of information contained within the vast numbers of viral genomes. This information has potential not only be able to predict host information about a virus but also to further our understanding of virus-host relationships and the evolutionary processes involved.

The aim of this thesis was to develop machine learning approaches to mine viral genomes for information about their host relationship. The first step was to use supervised machine learning to identify a range of features generated from viral genomes that are predictive of host taxonomic information. I then developed an approach to use these host predictive features to identify sites in viral proteins that are associated with host specificity. Interpreting the signals from the classification models is difficult because different evolutionary processes embed a complex mixture of signals within the virus genomes resulting in the host signal being a weak effect. In order to tease apart the complex signals and identify those associated with their host I developed MVC, a multi-view clustering method that uses a generative approach to find multiple views of the clustering structure in the data.

1.1 Thesis outline

The remainder of this thesis is structured as follows:

Chapter 2 The next chapter aims to introduce the world of viruses and gives enough background to follow the rational behind the approaches used in the rest of the thesis. In it I discuss their importance to life on earth and how they are both fundamentally different to, yet intrinsically link to cellular life. I then describe the nature of virus host relationships that leads imprinting of a host signal in virus genomes.

Chapter 3. I lay out the background to machine learning that underlies the approaches I have taken in this thesis. First, I cover the supervised classification and kernels that I used to tackle virus host prediction. I then discuss the concept of machine learning

interpretability which motivated Chapter 5. Finally, I introduce probabilistic modelling and show how generative modelling can be used to construct and sample the Dirichlet multinomial mixture models that are the basis of the MVC model introduced in Chapter6.

Chapter 4. I investigate the idea that different transformations of the genetic code more fully capture the different biological information present in the virus genomes. I demonstrate that feature sets generated from different representations of viral genomes are all predictive of host taxonomic information for viruses from all Baltimore classes and all host kingdoms. I showed that combining the different feature sets not only has the potential to improve accuracy but also enables us to tune a classifier for specificity versus sensitivity. This chapter is my first author paper:

- Young, Francesca, Simon Rogers, and David L. Robertson. "Predicting host taxonomic information from viral genomes: A comparison of feature representations". In: PLOS Computational Biology 16.5, 2020".

My contribution to this paper: I contributed to the design of the study with my co-authors, I wrote all the code and carried out the analysis and I wrote the manuscript.

Chapter 5. I explore the idea of interpreting the host predictive classifiers developed in Chapter 4 as signals on viral genomes. I show how "local features" can be used to identify regions in viral sequences that are associated with the virus-host relationship. I demonstrate this method by applying it to the SARS-CoV2 receptor binding site to identify potential sites that are important to host specificity.

Chapter 6. I present MVC, an exploratory multi-view clustering method to tease apart the complex mixture of signals contained within viral genomes. I describe the MVC model and Gibbs sampling scheme. I use synthetic data to demonstrate that it identifies covariates associated with weak effects in high dimensional noisy data. Finally, I report on the initial results from applying MVC to two virus datasets.

Chapter 7. This presents a summary of the main results of this thesis and its contribution. I also highlight some questions raised by this work and potential routes for future research.

Code availability: All code written for the work within this thesis is available on GitHub.

Chapter 4 : https://github.com/youngfran/virus_host_predict,

Chapter 5 : https://github.com/youngfran/iVirusML,

Chapter 6 : https://github.com/youngfran/MVC.

Chapter 2

Virus background

"The very essence of the virus is its fundamental entanglement with the genetic and metabolic machinery of the host."

Joshua Lederberg (American Nobel laureate, 1993)

This background chapter discusses the importance of viruses to life on earth and how they are both fundamentally different, yet intrinsically linked to cellular life. Then I discuss two parallel hypotheses of how the interwoven evolutionary histories of a virus with its host leads to a specific host-species "footprint" being present in virus genomes. The first is about what a virus must "do" in order to "hijack" its host in order to successfully replicate. The second, is about how the coevolutionary process result in the virus mimicking its host, both to evade the host's defences and to commandeer its host's resources, giving rise to a host specific signal.

2.1 Viruses are an integral part of life on earth.

Viruses are best known as disease causing pathogens but they are in fact an integral and essential part of life on earth. The importance of viruses as pathogens and their impacts to human, animal and plant health has been known since 1898 (Beijerinck 1898). More recently, through advances in viral metagenomics, it has been established that viruses have a profound influence on all of the earth's ecosystems, from the oceans where they

affect the biogeochemical cycling to our own microbiomes where they impact our health. (Falkowski et al. 2008; Suttle 2007; Turnbaugh et al. 2007).

Viruses are symbionts with their host and this relationship is dynamic and variable (Virgin 2014). It is now known that the majority of viruses are commensal and have no detrimental effect on their host (Roossinck and Bazán 2017). There are also many examples of mutualistic relationships with their host, for example virus infections that can help with drought tolerance in plants (Xu et al. 2008). In fact we only exist as a result of symbiogenesis with viruses, the integration of beneficial viral genes into the host genome. For example, the expression of the endogenous retrovirus gene, syncytin, allows the formation of the placenta in mammals that was co-opted from ancient retroviruses that infected our ancestors more than 130 million years ago (Lavialle et al. 2013).

Advances in metagenomics, whereby all genetic material within an environmental sample are sequenced together, has led to the discovery that viruses are the most abundant and ubiquitous biological entities on earth. They have been found in all environments from the deepest trenches in the oceans to the stratosphere. Multiple virus species have been found to infect every cellular species sampled in all three domains of life, Bacteria, Archaea and Eukaryotes. The number of virus particles circulating in the environment outnumber cellular life by an order of magnitude. The largest numbers are seen in phage, that is viruses that infect bacteria, with estimates of greater than 10^{31} phage particles on the planet (Edwards and Rohwer 2005). These numbers are at an astronomical scale: there are more virus particles on earth than there are stars in the universe; in a litre of coastal seawater there are more viral particles than there are humans on earth.

2.2 Viruses are fundamentally different to cellular life.

Viruses are entirely dependent on a cellular host for their survival. At the most basic level, a virus consists of genetic material packaged in a protein coat. In this particle state, termed a virion, viruses are passive agents that cannot actively do anything to influence their own survival until they encounter a suitable host cell. Unlike cellular life which reproduces independently by undergoing binary division, viruses subvert their infected host's intra-cellular "machinery" to manufacture and assemble their constituent parts into fully formed viruses. To do this they must enter an appropriate host cell and interact with the cellular systems. This intra-cellular state is the active "living" form of the virus,

when the chimeric virocell is given over to the expression and replication of new viruses (Forterre 2011).

Viruses do not code for genes for independent replication or energy production and are unable to actively respond to their environment. They can be seen as a state on the boundary of supra-molecular complexes and simple biological entities. This leads to the misleading question: Are viruses alive? There are different arguments as to whether viruses should be considered living entities, and it comes down to the definition of life. It could be argued that the virus in the virocell state, as a replicating evolving system, is a living entity. If a living organism is defined as the unit element of a continuous lineage with an individual evolutionary history, then the virus fits this definition (Koonin and Starokadomskyy 2016). Given how inextricably linked viruses are in the tree of life, maybe we need to redefine life, (Harris and Hill 2021)?

2.2.1 Replication strategy

All of cellular life uses double stranded DNA to store their genetic information. The ability of DNA to store, copy and code information is implicit in its double stranded complementary structure. The flow of this information from genome to a functional product follows the 'Central Dogma' of molecular biology, that is that DNA is transcribed by DNA Polymerase into messenger RNA (mRNA), which is then translated by the ribosomes into a protein. In contrast, viruses exploit all possible strategies to use nucleic acids as their genetic store. Viruses can use single or double-stranded forms of either DNA or RNA. These different strategies are captured by the Baltimore classification system that assigns all viruses into one of the seven classes (Baltimore 1971). For example, SARS-CoV-2 is a positive-sense single-stranded RNA virus, Baltimore class IV, whereas the majority of phage are double-stranded DNA viruses, Baltimore class I.

All viruses must make mRNA to use their host's ribosome to synthesise viral proteins. Their Baltimore class determines which genes they need to supply via their own genome to complete replication and protein synthesis.

Class I: dsDNA viruses can use their hosts systems for DNA polymerase for replication and transcription. **Class II**: ssDNA viruses will first need to make double-stranded DNA using the host's DNA polymerase before replication and protein synthesis.
Classes III,IV,V: The RNA viruses. Double-stranded (dsRNA), positive single-stranded

(+ssRNA) negative single stranded (-ssRNA) RNA viruses all need their own RNA dependent RNA polymerase (RdRP) gene to replicate their RNA genomes. dsRNA and -ssRNA viruses also need the RdRP to transcribe their genetic RNA into mRNA before translation by the host ribosomes.
Classes VI,VII: The reverse-transcribing viruses. Unlike the RNA viruses that do not go through a DNA phase, the reverse transcribing viruses need to use a virus reverse transcriptase gene, RT, either at the beginning (RNA-RT) or end (DNA-RT) of their replication cycle.

Viruses are polyphyletic, that is the different virus groups have arisen independently in evolution meaning that there is no universal common ancestor and hence no shared genes present in all viruses. Cellular life by contrast is monophyletic, with a universal common ancestor and genes shared across all cellular species. These genes can be used to construct the 'Tree of Life' or more accurately the 'Network of life', due to widespread gene transfer among lineages' (Puigbò et al. 2009).

The lack of shared genes and genetic diversity has made constructing a phylogeny of the deeper evolutionary relationships between viruses near impossible. Virus discovery is still happening at a high rate and uncultured viruses discovered through viral metagenomics are now included in the virus taxonomy by the ICTV, the committee responsible for assigning viral taxonomy, (Simmonds et al. 2017).

2.2.2 The origin of viruses

There is evidence that viruses have been infecting life for over 450Ma (Gifford and Tristem 2003) and ancient viruses have been identified in the fossils of diverse animals (Meyerson and Sawyer 2011). There are three alternative theories as to their origin: **The virus-first hypothesis**, in which viruses were the first biological entities in the primordial "soup" from the first self-replicating genetic elements and therefore predate the original cells, (Koonin et al. 2015); **The reduction hypothesis** suggests that viruses evolved from from early cells, that became parasitic and then lost all but their essential genes, the giant viruses such as the Mimivirus support this route,(Raoult and Forterre 2008); **The escape hypothesis** postulates that viruses came from genetic elements that gained the ability to move between cells. In fact, there is growing evidence that all three theories might be correct for different virus groups.

There are two ways that viruses acquire the ability to infect a host such as humans: cospeciation and cross-species transmission. Given that viruses have been evolving with life since long before the evolution of mammals, many human viruses will infect us because they have undergone co-speciation with our non-human ancestors; any viruses present in the ancestral host at a speciation event will then diverge along with their host into distinct new virus species.

The majority of human viruses have been introduced through cross-species transmission, with two thirds of human viruses are known to be zoonotic, that is they are also capable of infecting other vertebrates, (Woolhouse et al. 2012). Luckily, it is very rare for these spill-over events to achieve onward human to human transmission, and even rarer for them to become established, such as SARS-CoV-2. On-going cross-species transmission overwrites this cospeciation signal as the related forms are displaced. Viruses frequently switch to closely related hosts but rarely to more distant hosts resulting in a mirroring of their trees most evident at deeper levels of phylogeny(Geoghegan et al. 2017).

This is even more evident in phage-bacteria coevolution where most phage are highly specific and can only infect one strain of a bacterial species. Only a very few phage have a broad host range in that they are able to infect multiple distinct hosts within one genus, (Jonge et al. 2018).

2.2.3 Virus Diversity

The smallest known virus, porcine circovirus, PCV, is 17 nm in diameter with a genome of 1760 base-pairs,(bp), containing just 2 genes. The largest is the giant Pandoravirus with a diameter of over 1000nm with a genome of more than 2Mbp containing 2500 genes. Considering the limited gene repertoire available to viruses they display widely diverse morphology. To reduce the genetic load of the structural proteins on their compact genomes, viruses use highly symmetrical structures formed out of identical self-assembling subunits. The main structures seen are simple helical rods or icosahedral polygons although more complex structures are seen by combining both filamentous and polygonal shapes in bacteriophage.

The genetic diversity seen within viruses is much greater than within the whole of cellular life (Nasir and Caetano-Anollés 2015). This diversity is a result of three fundamental properties of viruses: high mutation rates; high numbers of progeny, each infected cell can

produce hundreds to thousands of imperfect copies of the single infecting virus; and fast replication cycle, in the order of hours, for example HIV's cycle is 6 hours.

Despite this potential for seemingly unlimited diversity, the virus is constrained by its need to maintain structural integrity of its molecules and by its dependence on its host. By studying ancient DNA (Meyerson and Sawyer 2011) and endogenous retroviral elements (EVEs) (Gifford and Tristem 2003) where viruses have become integrated into their host's genome, a time dependent rate of virus evolution is observed. That is, high rates of mutations are observed when measured over short timescales, whereas when measured over longer timescales, thousands or even millions of years, the mutation rates are several orders of magnitude slower, closer to those of their hosts (Simmonds et al. 2019). This means despite this diversity we are still able to identify common signals such as sequence homology and conserved gene order to detect more distant relationships in viral genomes, (Aiewsakun and Simmonds 2018).

2.3 The virus-host relationships leads to host specificity

To begin to understand how a virus can manipulate its host systems to turn its cells into "kamikaze virus factories" we first need to understand what the virus needs to achieve - its lifecycle - and what it needs to manipulate - the host cellular systems.

2.3.1 Virus life cycle

Although virus life cycles differ greatly between virus species, they are all totally dependent on their host to complete six stages that are essential for virus replication: attachment, cell entry, uncoating, replication, assembly and release. First the virus fusion protein must recognise and attach to a specific receptor on their host cellular surface. This recognition is key to determining the host specificity because a virus can only infect a cell expressing the right receptor. Followed by cell entry and uncoating to release the viral nucleic acid into the cell. The viral replication leading to the formation of new viral proteins and genomes by a mechanism that is dependent on its Baltimore class as described above. New viral particles are assembled and finally exit from the cell.

A virus must achieve its life cycle while evading the host defence systems. Due to the pressure to control viral infection, prokaryotes, archaea and eukaryotes have all evolved

elaborate and diverse defence systems. The importance of this defensive role is illustrated by the high proportion of the cellular genomes that are given over to defence. Eukaryote hosts have developed a large number of defensive mechanisms including both the innate and adaptive immune systems. The defensive mechanisms seen in phage include: restriction modification, CRISPR/Cas and abortive infection, premature cell death.

2.3.2 The host's cellular systems

The host's cellular systems are orchestrated by a complex interacting system of molecules, including proteins, RNA, lipids and metabolites. The different pathways that are responsible for all the different cellular processes are accomplished by thousands of proteins and associated interactions, each with a specific function. These complex and dynamic relationships can be represented as networks of protein-protein interactions (PIN). This system is coordinated by the regulatory and signalling network that controls all of the cellular processes.

Although the PIN is not strictly scale-free , it is a network that grows by adding new nodes to an existing node, with some preference to attach to more highly connected nodes (Barabási 2009). The emergent properties of these networks is that they tend to be modular systems that are very robust, so that failure of one node will not break the system. Many of the sub-networks/modules within the system are conserved through deep evolutionary time as a consequence of the evolution of functions via interacting molecules.

This molecular interaction network is mediated by domain-domain and domain-motif interactions in the context of protein structures. Domains are protein modules that are discrete structural and evolutionary units, each performing a different sub function within a protein. This modularity enables evolutionary "flexibility" and rapid innovation by the recombination of modules leading to new functions, (Apic and Russell 2010). Increasingly we are learning of the importance of domain-motif interactions involving Short liner motis (SLiMs), (Van Roey et al. 2014). These are short sub-sequences usually 3-15 amino acids long, typically with just a few sites that are critically important to function. Amino acid changes at each position of a motif will have a different effect on binding affinity, with amino acid substitutions at some positions 'breaking' the function. SLiMs are associated with intrinsically disordered regions of a protein. These are regions that do not form a stable three dimensional structure under normal physiological conditions but do on contact with a binding partner. SLiMs are associated with many crucial cellular processes. They

are at the core of the signalling and regulation system which is mediated via weak transitory interactions by means of high specificity and low-affinity interfaces. The disorder and structural flexibility results in binding promiscuity allowing a region to bind multiple partners enabling disordered regions to perform multiple vital functions giving rise to hub proteins. The small size and lack of structural constraints allow more evolability, ideal for 'ex-nihilo' evolution and functional innovation for both hosts and viruses, (Davey et al. 2015).

2.3.3 Virus host molecular interactions

In order to "hijack" its host, the virus must interact with the host's cellular systems to "re-wire" the cells for virus production. The virus must take control with the limited resources contained within their compact genomes. This requires viral proteins to both leverage the modular nature of their host's system and be multifunctional. Viruses are known to preferentially target the signalling system via the host's hub proteins, thereby causing a cascade of downstream alterations in the host network (King et al. 2018; Oyeyemi et al. 2015). Viruses achieve this through hundreds of specific protein-protein interactions mediated through both domain-domain and domain-motif interactions, (DDI and DMI), (Brito and Pinney 2017; Franzosa and Xia 2011; Garamszegi et al. 2013).

The human adenovirus E1A protein is a good example of how viral proteins combine structured domains with overlapping motifs in disordered regions to simultaneously interact with many proteins in different pathways thereby affecting many downstream cellular functions,(King et al. 2018). E1A acts as a viral hub protein with 32 primary and 2,207 secondary binding partners to control over both viral and cellular gene expression.

2.4 Virus-host coevolution imprints a host signal in virus genomes

In this section I discuss the different evolutionary processes are reflected in the genomes and how the antagonistic virus-host relationship results in the virus mimicking endogenous host molecular interfaces though convergent evolution.

2.4.1 Virus evolution

The same evolutionary processes that are responsible for the diversity seen in life on earth also apply to viruses as replicating systems. Evolution is the process by which descendants inherit genetic modifications from their ancestors resulting in changes in the traits or phenotype of a population. It is an inevitable consequence of random mutations causing genetic variation. The evolutionary processes of selection, genetic drift and gene flow then act to change the frequency of genotypes in a population over time. The changes in the traits of a virus can be observed at different levels: changes in the genome result in different genotypes or alleles; which in turn lead to changes in the molecular phenotype resulting in different structures and functionality; which lead to changes in the observable phenotype, such as virulence or the host they infect.

Random mutations occur as a result of copying mistakes during the process of duplicating genetic material. These mutations result in a differential in the fitness of the descendants, that is the relative ability of a genotype or allele to survive and propagate. Fitness is context dependent so that a genotype that confers fitness in one environment may be neutral or unfit in another. For a virus its fitness is defined in the context of its host and also its tropism, that is its ability to infect a particular cell (cellular tropism), tissue (tissue tropism) or host species (host tropism).

Selection is a natural consequence of the differential in fitness. Mutations that have a beneficial effect on fitness lead to an increase in numbers of that allele in a population resulting in positive or adaptive selection. Mutations that cause a decrease in numbers of progeny resulting in the removal of the deleterious allele is termed negative or purifying selection. Many mutations are neutral and have no effect on fitness such as synonymous mutations where mutations of a single nucleotide can leave the amino acid sequence unchanged. Adaptive evolution is an episode of positive selection through which specific traits that increase fitness become fixed in a population. Viruses have a huge capacity for adaptive evolution due to their intrinsic potential for high genetic variability caused by error-prone replication, huge numbers of progeny and fast replication cycle.

Another evolutionary mechanism that is particularly important in viruses is genetic drift which is the random sampling error of a population causing random changes in allele frequency and a big loss of genetic variation. Bottlenecks occur when a random environmental event causes a dramatic reduction in a population size. Whereas the founder effect occurs when a non-representative sub-population is isolated from the original population.

Drift is an important part of virus evolution occurring at different stages of its infection cycle, (Zwart and Elena 2015). It occurs within a host during infection when each new cell is only infected by a small sample of viruses; It occurs during transmission when a small sample transmits to a new host, increased viral load on infection is known to induce more severe disease implying higher virulence; It will also occur when a virus switches tropism, whether that is the specific type of cell or host it infects.

The balance of the different effects of these processes is dependent on the size of the population. Selection is a directional non-random process resulting in adaptive mutations being fixed in a population and is most evident in large populations. Drift is a random non-directional process that fixes neutral or even slightly deleterious mutations in a population. Its effects are greatest when the sampled sub-population is smallest. Neutral theory states that drift accounts for the majority of genetic diversity observed in diverging populations making it challenging to find signals of positive or negative selection in virus genomes against this background of neutral selection.

These different processes are reflected in the phylogenetic tree that represents the relationship between different virus genomes, see Figure 2.1. Convergent evolution is the development of analogous adaptations in species that do not share a common ancestor, termed homoplasy (the red nodes). Divergent evolution is the opposite and occurs when descendants diversify from their common ancestor over evolutionary time due to accumulating mutations. Molecular features that confer essential functionality will be conserved by negative selection resulting in homologous features in the sequences (navy nodes) but these may be confounded by homologous features caused by drift.

2.4.2 Virus-host coevolution

The antagonistic nature of the virus-host relationship results in an 'arms race' that drives the process of coevolution. This is the reciprocal evolutionary changes occurring between interacting species. The constant battle to maintain fitness is known as the 'Red Queen Hypothesis' from the quote in Lewis Carroll's -Through the Looking-Glass - in which the Red Queen tells Alice "Now, here, you see, it takes all the running you can do, to keep in the same place. ". This aptly describes the relentless rounds of recurrent adaptation and counter-adaptation that occur between a virus and its host (McLaughlin and Malik 2017).

This long term evolutionary pressure drives biological diversity (Hembry et al. 2014) by

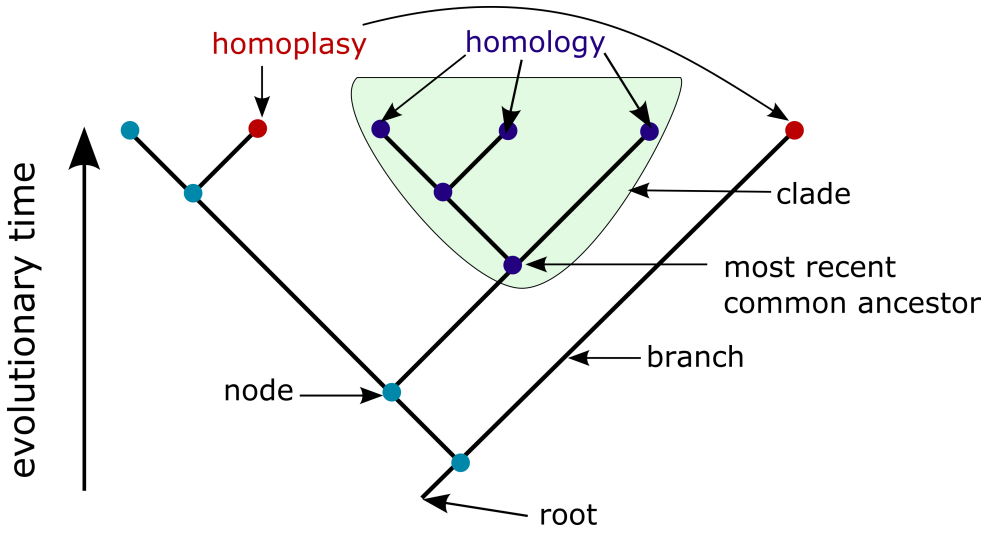


Figure 2.1: A phylogenetic tree, showing how divergent and convergent evolution gives rise to homologous and homoplastic features.

increasing the rates of evolution in both the virus and their host in the absence of environmental or other biotic selection pressures (Koskella and Brockhurst 2014). Its effects can be seen at genome, molecular and population levels, for example studies have found reciprocal changes at interaction sites (Lovell and Robertson 2010).

2.4.3 Mimicry

Mimicry is a consequence of coevolution and it is seen in interacting species throughout biology. Pathogens adopt a characteristic of their host at the molecular level to gain a benefit over its host. This strategy of molecular mimicry is adopted by all pathogens to take advantage of host cell functions to guarantee their replication and dissemination (Via et al. 2015; Davey et al. 2011). It occurs through convergent evolution with many examples of viral, prokaryotic and eukaryotic pathogens mimicking the same host motif, such as the integrin binding RGD motif (Chemes et al. 2015).

Viruses use molecular mimicry to disrupt a range of cellular functions including the host defences by modulating the cell's immunity (Guven-Maiorov et al. 2016) or to subvert the cellular systems to building new viruses (Elde and Malik 2009). Molecular mimicry is found in both the nucleotide and protein sequences of viruses.

2.4.3.1 Nucleotide mimicry

Di-nucleotide bias The di-nucleotide bias of some viruses tends to adapt over time to mimic that of their host (Greenbaum et al. 2008; Atkinson et al. 2014; Simmonds et al. 2013). This is believed to be evasion mechanism to avoid being recognised as foreign nucleic acid and hence not triggering the hosts anti-viral defences. For example, the antiviral protein ZAP is known to selectively bind to GC-rich sequences. The methylation of the cytosine in a CpG di-nucleotide in DNA leads to a reduced occurrence of the CpG in vertebrate DNA. Whereas this methylation does not occur in RNA. This leads to the idea that the low CpG bias in RNA viruses of vertebrates has come about through the selective removal of synonymous CG pairs over time, (Goff 2017).

Codon bias Viruses codon are also under selective pressure to optimise their use of the host resources and to adapt to the translation biases of their hosts. Codon bias which is not uniform across species are the differences in frequency of the use of the synonymous codons. Again, this results in viruses tending to adopt the codon biases of their hosts (Carbone 2008; Jenkins and Holmes 2003).

2.4.3.2 Protein mimicry

Viruses manipulate their host's PIN by mimicking existing endogenous domain-domain and domain-motif interfaces. (Franzosa and Xia 2011; Guven-Maiorov et al. 2016).

Domain motif interactions,(DMI) Viruses use mimicry of their host's SLiMs extensively in many different pathways, (Davey et al. 2011; Hagai et al. 2014; Via et al. 2015). High mutation rates combined with high levels of disordered regions mean that viruses can 'easily' converge to mimic their host SLiMs (Chemes et al. 2015). This can be perfect mimicry to co-opt the host pathway of imperfect mimicry in which viruses 'improve' interactions to usurp host protein partners to re-purpose a pathway. The advantage of closer mimicry for the virus is that it makes it more difficult for the host to evolve in the next step of the arms race without breaking a host function.

Domain domain interactions, (DDI) Viruses are known to target domain-domain interfaces using structural mimicry, (Brito and Pinney 2017; Garamszegi et al. 2013).

Domain mimics usually evolve through divergence after HGT and very rarely through convergence, (Elde and Malik 2009). It is difficult to find evidence of domains that have evolved through convergence due to lack of sequence similarity. In cellular life, domain architecture is driven by evolutionary descent and convergence accounts for less than 4%, forming through chance rather than functional necessity (Gough 2005).

2.5 Host predictive signals

My objective at the start of the PhD was to use a machine learning approach to identify features generated from viral genomes that can be used to predict the host. These features are measurable attributes or variables of a sequence that encapsulate any discriminative signal. At this point (2017), existing machine learning approaches to predicting virus host interactions tended to focus on short nucleotide features. The virus-host molecular interactions and coevolutionary processes will be reflected in different representations of virus genomes. Based on the premise that each representation will capture information about different types of interactions, we tested from four different 'levels' of viral genome representation: nucleotide sequences, amino acid sequences and domain repertoire.

We used an additional representation in which an amino acid sequence is transformed into a sequence of properties. The aim is to group amino-acids into bins based on their physio-chemical properties so that conservative substitutions do not affect the transformed sequence. A conservative substitution is the replacement of an amino acid by one with similar properties that are unlikely to cause dysfunction in the protein. One of the difficulties is that amino acids have multiple overlapping properties such as size, polarity and hydrophobicity which will group the amino acids in alternative ways. Additionally, the properties of amino acids are dependent on their environment and will change depending on their position within a protein structure. There is no definitive grouping that can capture all of this complexity.

It is hoped that by using this more flexible representation that is robust to conservative substitutions that it is able to capture more of the functional signal. Another advantage of using different representations is the different rates at which they diverge from very rapid nucleotide evolutionary changes to long lasting domains. To use virus genomes in machine learning we need to transform a sequence into a numerical vector. Throughout this thesis we have used kmer composition to encode all sequence data. Kmer features are

based on the counts of each unique kmer or subsequence of length k in a sequence. The domain repertoire of each virus was represented as the presence-absence of each unique domain observed within the a dataset.

Chapter 3

Background to machine learning

In this Chapter I lay out the background to machine learning that underlies the approaches I have taken in this thesis. First, I cover classification, support vector machines and kernels that I used to tackle virus host prediction in Chapter 4 . In Section 3.1 I discuss the concept of machine learning interpretability which motivated Chapter 5. In Section 3.3 I introduce probabilistic modelling and show how generative modelling can be used to construct and sample the Dirichlet multinomial mixture models that are the basis of the MVC model introduced in Chapter 6.

Machine learning is a set of methods that enable computers to use an algorithm to learn from data. These algorithms can automatically detect patterns in data without the need for explicit assumptions about the data or any underlying process behind its generation. This is particularly useful in computational biology where the mechanism responsible for the phenotype of interest is often unknown or poorly defined. These patterns can then be used to make predictions on future data or to extract knowledge contained within the data. (Murphy 2012) For instance, if we train machine learning model to classify patients on whether they have cancer or not, we can then use the model to: predict if a new patient has cancer; learn the biomarkers that are discriminate between patients with or without cancer; this may lead to insights into the mechanisms causing cancer.

Machine learning models are able to discover weak patterns in complex and noisy data without requiring prior knowledge of the specific mechanisms responsible for the label of interest (Bishop 2006). The advent of the massive and multiple dimensional biological datasets generated by high-throughput technologies, such as genomics, proteomics, metabolomics and epigenomics has meant that machine learning is becoming an integral

step in biomedical research. It has been successfully applied to a wide range of biomedical problems (Libbrecht and Noble 2015; Leung et al. 2016). In Chapter 4 we will discuss machine learning approaches to predicting virus host interactions and Coclet and Roux (2021) provide a more up to date review. Elsewhere in virology machine learning has been applied to the classification of viral genomes (Remita et al. 2017; Yu et al. 2013), predicting viral genes(Silva et al. 2017) and predicting virus-host protein-protein interactions (Barman et al. 2014; Liu et al. 2018).

3.1 Classification

Classification is a supervised machine learning method that learns the function that maps a set of input variables or features $\boldsymbol{x}$, to a set of discrete output variables or labels, $y \in \{0, 1\}$ or $\{0, 1, 2, 3, ...\}$, that is the function $y = f(\boldsymbol{x})$. The aim of classification is to classify labelled data into different classes based on the labels, for example, the host of a virus. Once a classifier has been trained and tested with existing labelled data it can be used to predict the class of new data.

3.1.1 Terminology

Dataset: This is a matrix that is made up of N objects each with D features that are used along with the associated labels **y** to train and test a classifier. These are the input variables and are often presented in a table. The dataset is split into the **training set** and **test set** before any data pre-processing or training takes place.

Object, Individual: These are the rows in our dataset, each one is generally represented by a D-dimensional numerical vector and its label. In our case this is the viruses, either referred to by species name, tax-id or accession number.

Label: The label or class of the data is the categorical target variable that the classifier learns to predict, in our case this is the taxa of the host of the virus, this can be at different taxonomic ranks, for example species or order. Other terms used are output or response variable, usually theses are a set of categorical or nominal variables.

Features: The first and most important challenge in developing a classifier is to identify discriminative features of the objects that can separate the data into the required classes. This is the subject of chapter 4. Other terms used to are features, attributes or covariates, in Chapter 6 we use covariate to refer to data-agnostic variables, in other words, at that point, we are not interested in where these variables came from.

Models: There are many alternative supervised machine learning models from logistic regression to more complex methods such as gradient boosting machines. These all use different algorithms to try and optimise classification. We will focus here on describing support vector machines (SVM) which we use in both chapter 4 and 5.

Evaluation: Finally, you need to evaluate the performance of your classifier to predict the class of unseen data, (not used in training), which we discuss in section 3.1.7

3.1.2 Support vector machines

Support vector machines (SVM) has been used in Chapters 4 and 5, it was chosen due to its generally high performance, and the fact that (in the linear case) we are able to extract some biological interpretation of the classification function. SVM is a binary classifier where the output of the model is the assignment of an object to either the positive or negative class. Given a set of N training objects $\boldsymbol{x_1}, ..., \boldsymbol{x_N}$, where each object is represented by a vector of dimension D. Each object is assigned to a class described by the label $y_n \in \{-1, 1\}$. The SVM model learns the decision boundary that best separates the two classes of objects using a linear function of the form $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$. This function assigns a label +1 to the objects $\boldsymbol{x}$ with $f(\boldsymbol{x}) \geq 0$, the positive class and a label -1 to the objects $\boldsymbol{x}$ with $f(\boldsymbol{x}) < 0$, the negative class. The task for a machine learning model is to learn the function f from the set of objects $\boldsymbol{x_1}, ..., \boldsymbol{x_N}$ and their associated labels $\boldsymbol{y}$. For each candidate function, we can check how many objects are correctly classified. Many models aim to minimise the number of misclassifications, but this often does not define a unique solution. SVM takes a different approach that it aims to maximise the confidence of the classifications by maximising margin between the decision boundary and the nearest points on both sides. see Figure 3.1

This decision function is given by

$$y_{new} = \text{sign}(\boldsymbol{w}^T \boldsymbol{x_{new}} + b)$$

The parameters $\boldsymbol{w}$ and b are chosen to maximise the boundary margins by minimising the objective function

$$\underset{\boldsymbol{w}}{\text{argmin}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{subject to } \; y_n(\boldsymbol{w}^T \boldsymbol{x_n} + b) - 1 \geq 0, \text{ for all n.}$$

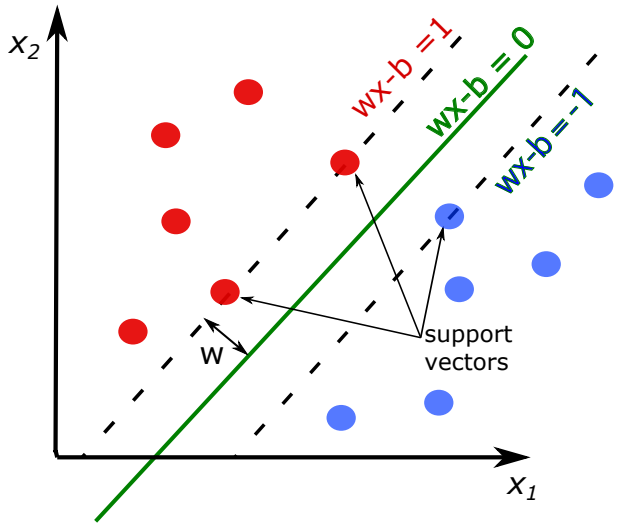


Figure 3.1: Representation of the SVM decision boundary in 2.D, the green line. The margin that separates the data into two classes is shown by the dotted lines that pass through the support vectors. The positive class represented by the red dots and the negative class represented by blue dots.

This is a convex constrained quadratic optimization problem that can be solved numerically. In solving this problem, we find that only a few objects are used to find the optimal support boundary. These points are known as the **support vectors** and are the set of points closest to the maximum margin decision boundary.

This can be reformulated as the "dual form" where the objective function is

$$\underset{\alpha}{\text{argmax}} \; \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n,m=1}^{N} \alpha_m \alpha_n y_m y_n \boldsymbol{x}_m^T \boldsymbol{x}_n \tag{3.1}$$

$$\text{subject to } \alpha_n \geq 0, \sum_{n=1}^{N} \alpha_n y_n = 0$$

This has the advantage that if the number of training examples is smaller than the number of features the number of free parameters is greatly reduced, with $\boldsymbol{w}$ removed from optimisation and bounded by the number of support vectors.

Soft margins. The concept of a soft margin is introduced to allow the algorithm to tolerate a few objects to be on the wrong side of the decision boundary. A new parameter C is introduced which controls the amount of tolerance for objects to either sit within the margin or on the wrong side of the boundary. This is done by balancing the trade-off between maximising the margin and minimising the number of misclassified objects when optimising the decision boundary.

$$\underset{\boldsymbol{w}}{\text{argmin}} \qquad \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{n} \xi_n$$

Where ξ_n represents the possibility that some points lie on the wrong side of the boundary. This can still be framed as the problem of maximising equation 3.1 above but with the added constraint of an upper bound of C on α_n such that $0 \leq \alpha_n \leq C$, for all n. So far, we have been dealing with data that can be separated by a linear boundary but with some added flexibility for noisy data that allows some training points to occur on the wrong side of the boundary.

3.1.3 Kernels

Next, we consider data that is not linearly separable in the original, or input space. The goal is to apply a function $\phi(\boldsymbol{x_n})$ to map the data from the input space into a feature space where the classes are linearly separable. We can then fit the decision boundary, a hyperplane in this feature space, Figure 3.2.

The kernel trick provides a way of finding the linear decision boundary in feature space without having to explicitly transform the data. Kernel methods are based on an alternative way of representing data. Instead an $N \times M$ matrix representing each of the

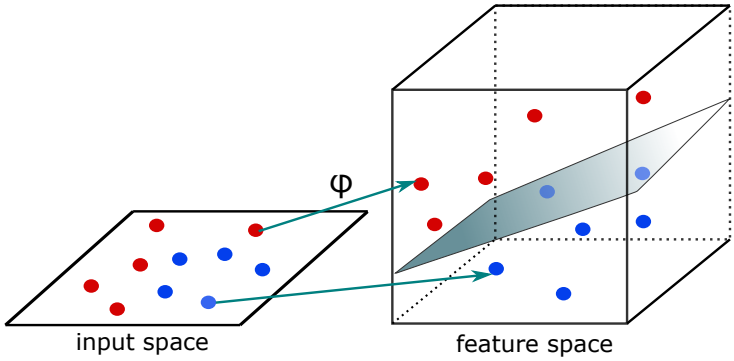


Figure 3.2: Non-linear data is mapped from input space to a feature space allowing the classes to be linearly separable.

N individuals as an M vector of real numbers, the data is represented by an $N \times N$ kernel matrix. Each element is the measure of pairwise similarity between individuals $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ using a kernel function, such that $k_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_j})$. A kernel function takes vectors in input space and returns the dot product of the vectors in feature space without explicitly transforming or representing the data in feature space.

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle \phi(\boldsymbol{x_i}), \phi(\boldsymbol{x_j}) \rangle$$

where $\langle u, v \rangle$ represents the dot product of two points u,v.

The power of using the dual formulation is that equation 3.1 only includes the dot product of the transformed vectors so that the term $\boldsymbol{x}_m^T \boldsymbol{x}_n$ becomes $\langle \phi(\boldsymbol{x_m}), \phi(\boldsymbol{x_m}) \rangle$ which can be replaced by any kernel function. This means we can find the decision boundary in feature space without having to transform the data.

There are many kernel functions each equivalent to the dot product after some transformation such as the Gaussian or polynomial kernels. Although these can perform better than a linear kernel, the relationship between the features and the decision boundary $\boldsymbol{w}$ is no longer easily accessible. In Chapter 5 Where we aim to interpret a host predictive model on a viral sequence, we opt to use a linear kernel specifically because we can access $\boldsymbol{w}$ directly from the model. Kernel functions can also be defined to arbitrary objects for which you can define some notion of similarity, meaning that SVM can classify objects such as strings, trees or graphs without the need to represent individuals as a numerical vector of features.

3.1.4 Kernel combination

Data fusion is the process of integrating heterogeneous data from multiple sources with the expectation is that the fused data is more discriminative than the individual sources. Different approaches of data integration have been applied to successful combine data from multiple technologies across biology and medicine, (Zitnik et al. 2019). Early and late integration strategies can be used for any classification model, see Figure 3.3. Early or feature integration involves concatenating the vectors from the different datasets for each object before training the classifier in the usual way. Late or classifier integration involves training the classifiers on the individual datasets and then combining their resulting scores, these are known as ensemble methods. The third strategy of intermediate or kernel integration is only applicable to kernel methods. It involves computing the kernel for the different datasets and combining these kernels before training a classifier. As shown above, any kernel can be used in the SVM dual equation to transform the data into a linear classification problem. The property of kernel functions that enables intermediate integration is that individual kernels can be combined in any linear combination to produce another valid kernel (Lanckriet et al. 2004). We can combine N kernels using

$$K = \sum_{i=1}^{N} w_i k_i \qquad \text{where} \quad \sum_{i}^{N} w_i = 1$$

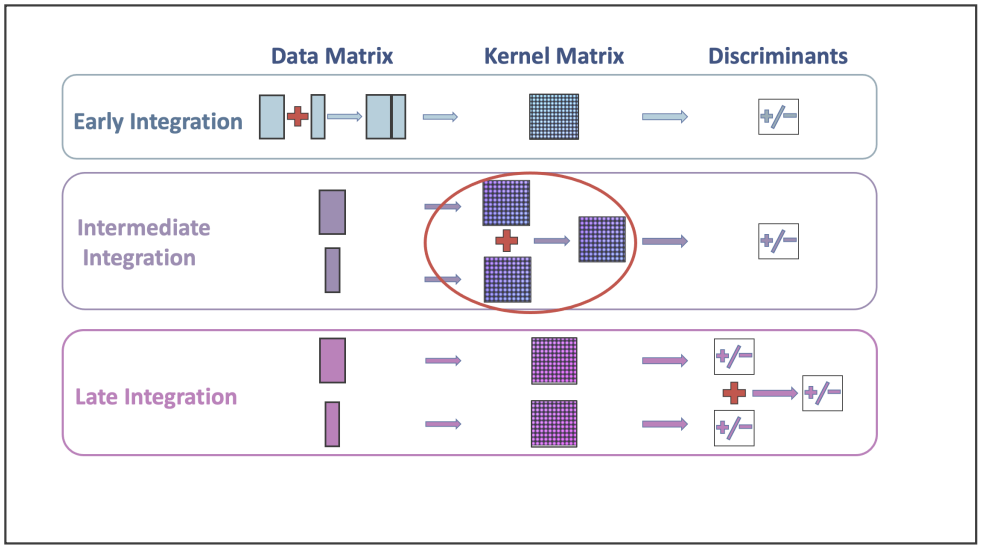


Figure 3.3: Kernel combination. The different approaches to data fusion. Showing how intermediate integration can be achieved by combining different kernels representations of the data.

3.1.5 Platt Scaling

SVM does not provide the probability of class membership for an object as its output, instead this can be calculated retrospectively by applying Platt scaling. This is a method to transform the outputs of a classification model into a probability distribution over classes. It works by using maximum likelihood estimation to fit a logistic regression model to the raw classifier outputs, and then uses this model to assign a probability for each object $P(y_n = 1|x_n)$.

3.1.6 Advantages of SVM

SVM has been successfully applied across a broad range of domains including face recognition, fraud detection, text classification and throughout computation biology.(Cervantes et al. 2020; Ben-Hur et al. 2008; Yang 2004). It is usually one of the top performing classifiers and has a number of advantages over other classification algorithms. SVM work very well for high dimensional data, that is when the number of features is much greater than the number of training samples. This is because the dual formulation optimises the decision boundary using a kernel hence they only need to learn one parameter per observation and the number of dimensions is irrelevant. As a result they work well with sparse data where there are few training samples and scale with the number of samples rather than the dimension of the data. SVMs are particularly good at generalising to new data and at dealing with outliers because the SVM algorithm maximises the margin between the classes based on a minimum set of support vectors in comparison to other algorithms that minimises the error of all the objects. Although, in cases with small numbers of training examples SVM can be very susceptible to outliers, especially if an outlier is used as one of support vectors.

As a kernel method, they can easily and effectively be applied to combine heterogeneous data. Finally, SVM training is relatively straightforward as the optimisation problem is convex, resulting in a single global minimum, even when the kernel is nonlinear. Despite the likelihood in of a decrease in performance of linear SVM over a more complex kernels they are much easier to interpret because the learnt coefficients are accessible and relate to the feature importance.

3.1.7 Evaluation

Evaluation is used to assess the classification performance of a model. This performed using a set of independent test data that has not been used in any part of the training process or pre-processing such as feature scaling. There are many alternative metrics that can be used for evaluation, many are based on the confusion matrix which quantifies the number of objects that are true positives (TP), true negatives (TN), false positives (FP) and false negatives (TN) in the results, (see top right panel of Figure 3.4).

Accuracy is the proportion of objects that classified are correctly and can lie between 0 and 1. Accuracy is not a good metric when the classes are imbalanced when high accuracy can be achieved by selecting the model that assigns all the objects to the majority class.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Area under the ROC curve Is a way of combining the sensitivity and specificity into one value and is the metric we use throughout this thesis. Most classification algorithms assign a real-valued output to each test object, a threshold is then applied to assign the object to the positive or negative class resulting in a TP, FN, FP or TP result. Altering this threshold will change the resulting numbers assigned to each quadrant. The ROC curve (receiver operator characteristic curve) tracks these changes as the threshold is changed from 0 to 1 by plotting the true positive rate (TPR) or Sensitivity against the false positive rate (FPR) or 1- Specificity. The area under this curve, the AUC, provides an aggregate measure of performance across all possible thresholds, Figure 3.4.

AUC represents the probability that a binary classifier will rank a randomly chosen positive object higher than a randomly chosen negative one. An AUC of 1 indicates that the probabilities of all the objects in the positive class are greater than those in the negative class. An AUC of 0.5 indicates that there is no separation in the classes and it is the equivalent of guessing. SVM does give real valued assignments in $\{-1, 1\}$ and the threshold is 0 which can be used to compute AUC but it is more usual to convert the class assignments

into probability scores with Platt scaling first.

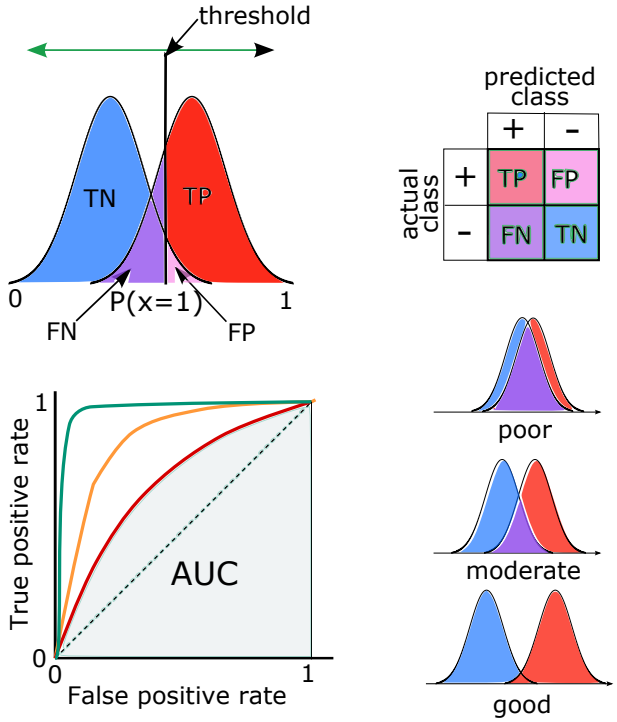


Figure 3.4: ROC curves and AUC. The top left panel shows how the pairs of values used to plot the ROC curves are computed from the confusion matrix as the threshold is changed from 0 to 1. The plot in the bottom left panel shows a comparison of the curves for a poor, moderate and good classifier.

AUC is equivalent to the two sample Wilcoxon rank-sum statistic, a non-parametric test to compare differences between two independent groups with no assumptions about their distribution, such that $U = AUC \times n_p \times n_n$ where n_p and n_n are the number of positive and negative objects. AUC is both threshold and scale invariant, meaning it can tell us if our features are discriminative without the need to pick the optimum threshold. It is not the best metric to optimize a classifier for all use cases, particularly when there are wide disparities in the cost of false negatives vs. false positives, such as in medical diagnosis.

3.2 Interpreting machine learning models

In Chapter 5 we explore the idea of interpreting a host predictive classifier in terms of the importance of each site of a protein sequence to the virus-host relationship. The use of supervised machine learning is often applied as a "black box" with researchers focused on optimising a method to make reliable predictions about unseen data. For example this

approach is widely applied in medicine, (Topol 2019). An additional goal is to use the underlying patterns learnt by the machine learning algorithms to gain novel biological insights from the massive and complex datasets. To successfully make predictions a model must be encoding some discriminative features contained within data, this encoding may provide previously unknown information which could be important to the phenotype/classes being studied. In this context the motivation behind interpreting a machine learning model is to extract this information with the hope of furthering our understanding the system being modelled.

The increasing power and complexity of machine learning across a huge range of applications throughout society has led to the rapid growth in the sub-field of interpretable machine learning which is focused on providing explanations of the workings of complex models (Gilpin et al. 2018; Watson 2021). As yet, there is no consensus as to the definition of interpretability in machine learning. It can be used in a context-free way to refer to how easy it is to understand the internal workings of a model, for example, explaining how a decision about an individual prediction is made. For example, "when feature 3 is high, the model predicts class 2". It can also used in a domain specific way to refer to how easy it is to understand the relationship between the features in the model and the system being modelled. For example "the amino acids coded for in this part of the sequence lead to the prediction that this virus can infect humans". In general, it can be thought of as extracting knowledge from a model that is relevant to the task at hand (Molnar 2022).

The are many different motivations for requiring an interpretation or explanation of a model; model transparency is important in many fields, for example in medicine where clinicians must trust the results of a model to inform clinical decisions and to explain their decisions to their patients; regulators need to trust that a model will always behave as expected and so must understand how a model works; troubleshooting is important in developing reliable models with no underlying and unwanted biases; finally, for knowledge discovery, as described above.

A major challenge for interpreting machine learning in biology where we are trying to understand the underlying biology behind is that correlation does not always equal causation. Many of the patterns uncovered by the model may be spurious associations that are not biologically relevant, leading to a high false positive rate. Unlike prediction, which has well established metrics to evaluate the performance of a model there is no rigorous metrics to evaluate the interpretability of a model. Estimating the error rates of a model explanation is particularly difficult in genomics where we do not have a full ground truth.

Another difficulty is that even an expert cannot intuitively understand how sequence data influences a phenotype. This is in contrast to understanding images which humans have evolved to quickly make sense of.

Models that are intrinsically interpretable have a clear path from the decision back to the input features. For example for any linear model such as a linear SVM we can compute $\boldsymbol{w}$ which directly links the model predictions back to the features. An increase in the feature weight either always leads to an increase or always to a decrease in the target outcome.

Unfortunately many real world problems cannot be modelled with a linear model, and as model complexity increases interpretability becomes more difficult. For example, for a small neural net with a simple architecture, the decisions can be traced back to both the input and internal nodes. Once the number of layers and nodes increase a single output may involve millions of calculations making it increasingly difficult and costly to interpret. Again, for a net with layers containing non-linear operators such as max pooling we lose the ability to understand a node's importance to the final output. Unlike intrinsically interpretable models, more complex models generally require a "post hoc" method to analyse the trained model.

The emphasis in biological research is to interpret machine learning models to gain novel insights about the biological system that is being modelled. The assumption is that the discriminatory patterns learnt by the model to predict the phenotype have a functional relationship with that phenotype. The goal of interpreting a model, whether to explain the the model or explaining the biological system that is being model will determine the approach taken and what is the appropriate balance the between the performance gains from using a more complex models against the drop in interpretability/explainability. In Chapter 5 we opted to use a linear SVM over using a more complex but potentially more informative model in order to allow such interpretation.

3.3 Bayesian Approach to Clustering

In this section I introduce probabilistic modelling and show how generative models can be used to construct and sample the Dirichlet multinomial mixture models that are the basis of the MVC model introduced in Chapter 6

3.3.1 Bayesian Statistics

In traditional frequentist statistics the world is described by fixed point parameters such as the average frequency of an event. In Bayesian statistics all parameters are treated as random variables and our current understanding of those variables is formalised in distributions. Parameters for which we know a lot might have very narrow distributions (low variance) whilst parameters about which we know little might have very wide distributions. In a Bayesian machine learning method, these uncertainties can be propagated through to predictions, ideally ensuring that the model has correctly quantified the degree of confidence in the predictions it makes.

3.3.2 Probability primer

In the remainder of the chapter we rely heavily on certain concepts from probability theory. They are briefly introduced here. A reader well versed in these concepts can jump to Section 3.3.4.

If the outcome of an event A can have one of J states we can write this as $A \in \{a_1, a_2, ..., a_J\}$
The probability of outcome j is $P(a_j)$ $0 \leq P(a_j) \geq 1$ and $\sum_j P(a_j) = 1$
The probability distribution of event A is $p(A) = P(a_1), P(a_2), ..., P(a_J)$

Joint probability for independent events

$$P(x_1, x_2, ...x_j) = P(x_1) \times P(x_2), ..., P(x_J) = \prod_{j=1}^{J} P(x_j)$$

Conditional probability is the joint probability for dependant events, in other words the probability of y given x

$$P(x, y) = P(y|x)p(y)$$

Marginal probability is the probability of A occurring whatever the outcome of B

$$P(A) = \sum_B P(A, B)$$

The chain rule allows you to build complex joint distributions from simple components,

for example, the probability of the events A ,B and C all occurring is given by

$$P(A, B, C) = P(A|B, C)P(B, C) = P(A|B, C)P(B|C)P(C)$$

Bayes rule is derived from the definition of conditional probability of two events occurring, it is common to frame it in terms of updating our belief about a model in light of new evidence such as observed data

$$P(\theta|X) = \frac{P(\theta)P(X|\theta)}{P(X)} \tag{3.2}$$

where:

$P(X)$ is the evidence, the observed data, this a normalising constant.
$P(\theta)$ is our prior beliefs about the model parameters.
$p(\theta|X)$ is the conditional probability of the model parameters given the data, the posterior probability.
$p(X|\theta)$ is known as the likelihood and is the conditional probability of the data given the model parameters.

Bayes rule shows us how to update the distribution over the model parameters θ to get the **posterior** distribution, $P(\theta|X)$. We use the likelihood of the data given the model parameters $P(X|\theta)$ to update initial **prior** distribution over the parameters, $P(\theta)$. We will use this to derive our Gibbs sampling scheme in Section 3.3.6 which is used to make inferences from our MVC model in Chapter 6.

3.3.3 Probability distributions

A probability distribution is a mathematical function that gives the probability of an outcome over all possible outcomes and is used to describe the characteristics of a random variable. There are different families of distributions that are used to for different experimental scenarios. Each family has a set of parameters that can be changed to alter the characteristics of a distribution, for example a Gaussian distribution is described by a mean μ which defines its position and a variance, σ^2, which defines its spread. We will focus on multinomial and Dirichlet distributions as these are the basis of chapter 6.

The multinomial distribution The multinomial distribution can be used to model the outcome of multiple categorical events, for example, of throwing a K-sided die N times. Let $\boldsymbol{x} = (x_1, x_2, ...x_K)$ be a random vector where x_i is the number of times the die lands on the kth side. The probability of a single throw being a k is $P(X_n = k|\theta) = \theta_k$. $\boldsymbol{\theta}$ is the vector $(\theta_1, ..., \theta_k)$ where $\sum_k \theta_k = 1$ and $\theta_k \geq 0$. The outcome of N throws has a multinomial distribution where N and θ are the multinomial parameters. The probability distribution is given by

$$p(\boldsymbol{x}|N, \boldsymbol{\theta}) = \left(\frac{N!}{x_1!...x_k!}\right) \prod_{k=1}^{K} \theta_k^{x_k}$$

The special case of only one draw from a multinomial distribution is known as the **categorical distribution**.

Dirichlet distribution The Dirichlet distribution defines a distribution over a simplex, that is all vectors with positive values that sum to one, $0 \leq \theta_k \leq 1$ and $\sum_{k=1}^{K} \theta_k = 1$. These are the same conditions as for the probabilities of a multinomial meaning that sampling a Dirichlet distribution can be used to select the parameters for a multinomial. The probability density function is given by

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = Dir(\alpha_1, ...\alpha_K) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \tag{3.3}$$

Dirichlet multinomial distribution Next we will use Bayes rule to derive the posterior distributions for a multinomial model with a Dirichlet prior, this is the distribution that is the basis of our model in chapter 6.

The likelihood of data given a multinomial distribution ignoring the constant because it doesn't involve $\boldsymbol{\theta}$

$$p(\boldsymbol{x}|\boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{x_k}$$

A suitable prior for θ, a vector of probabilities is the Dirichlet distribution, again ignoring the constant because it doesn't involve $\boldsymbol{\theta}$

$$p(\boldsymbol{\theta}|\alpha) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

Substituting these into Bayes equation 3.2, the posterior distribution for the multinomial parameter is given by

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \prod_{k=1}^{K} \theta_k^{x_k}$$
$$\propto \prod \theta_k^{\alpha_k + x_k - 1}$$

This is the unnormalised conditional density for $\boldsymbol{\theta}$, it is the product over the components of $\boldsymbol{\theta}$, each raised to the power of $\alpha'_k - 1$ where $\alpha'_k = \alpha_k + x_k$. This the exact form of the Dirichlet definition given in equation 3.3 and can be written as

$$(\boldsymbol{\theta}|\boldsymbol{x}) = Dir(\alpha_1 + x_1, ..., \alpha_K + x_K) \tag{3.4}$$

We see from this equation that the posterior distribution is another Dirichlet. This means that the Dirichlet is a **conjugate** prior of the multinomial. This is an important property of this distribution that greatly simplifies computations in probabilistic modelling.

Samples drawn from a K-dimensional Dirichlet distribution can be used as the parameters for a k dimensional multinomial, by changing the priors, α we can control how sparse or uniform the data sampled from the multinomial distribution. The relationship between a prior Dirichlet distribution and the multinomial parameter drawn from it are shown in Figure 3.5. It demonstrates how the prior parameters $\boldsymbol{\alpha}$ control the shape of the Dirichlet probability density function, with low values ($\alpha < 1$) concentrating the density at the corners leading to sparse multinomials. An α of 1 results in a uniform distribution ,whereas high parameters ($\alpha > 1$), concentrate the density in the centre leading to more dispersed multinomials. Non-symmetric priors with different values of alpha leads to multinomials favouring one dimension.

3.3.4 Mixture models

A mixture model is a statistical approach to clustering in which the model represents multiple sub-populations within data. Each cluster or component of the mixture is represented as a probability density function. These component distributions can then be combined into a joint distribution enabling us to model data that could not be modelled by a single component. For example, data that is multimodal. The use of a probability function to describe the model means that we can use any type of data, for example, continuous data

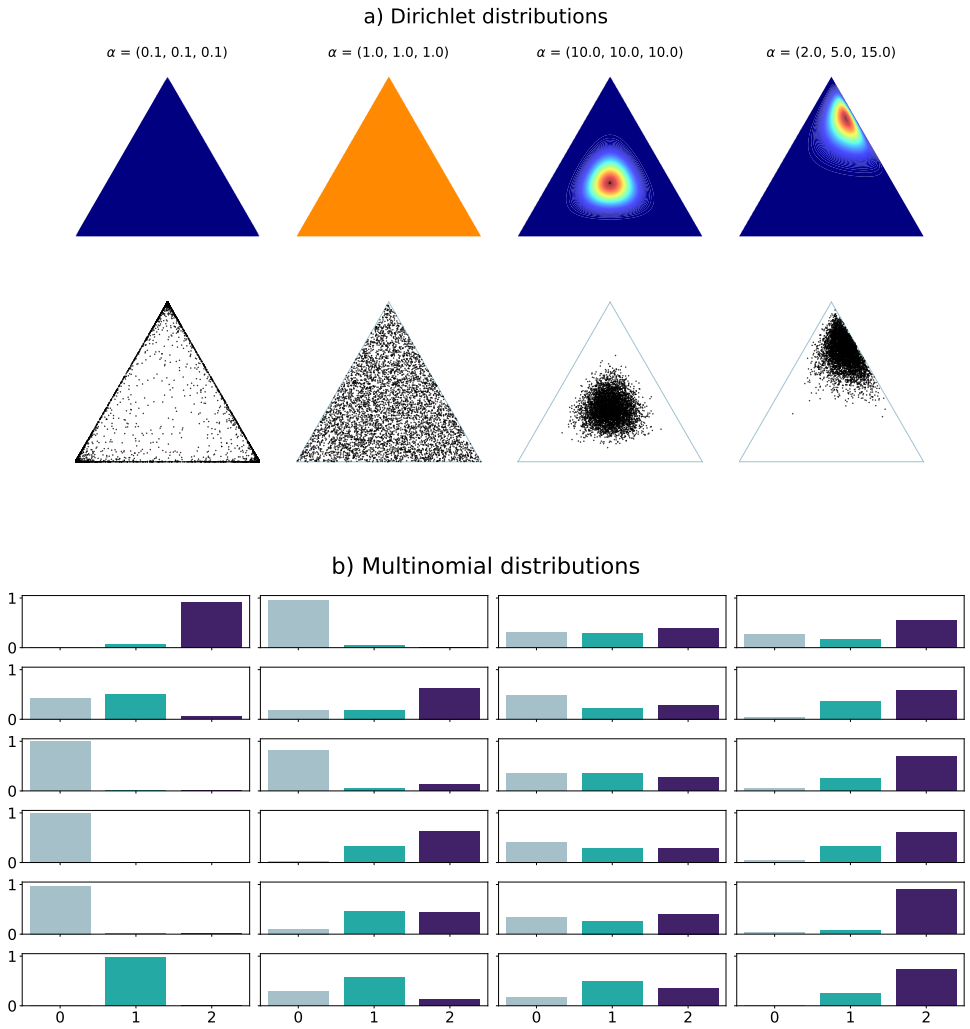


Figure 3.5: This demonstrates the effects different values of α a) on the Dirichlet distribution in top row where dark blue represents low probability and red high probability, b) the distribution of 1000 samples drawn from each Dirichlet distribution in the corresponding position in the second row, and c) 5 examples of multinomial distributions drawn from the corresponding distributions.

can be modelled with a Gaussian distribution, where as in our case we will model discrete data with multinomial distributions.

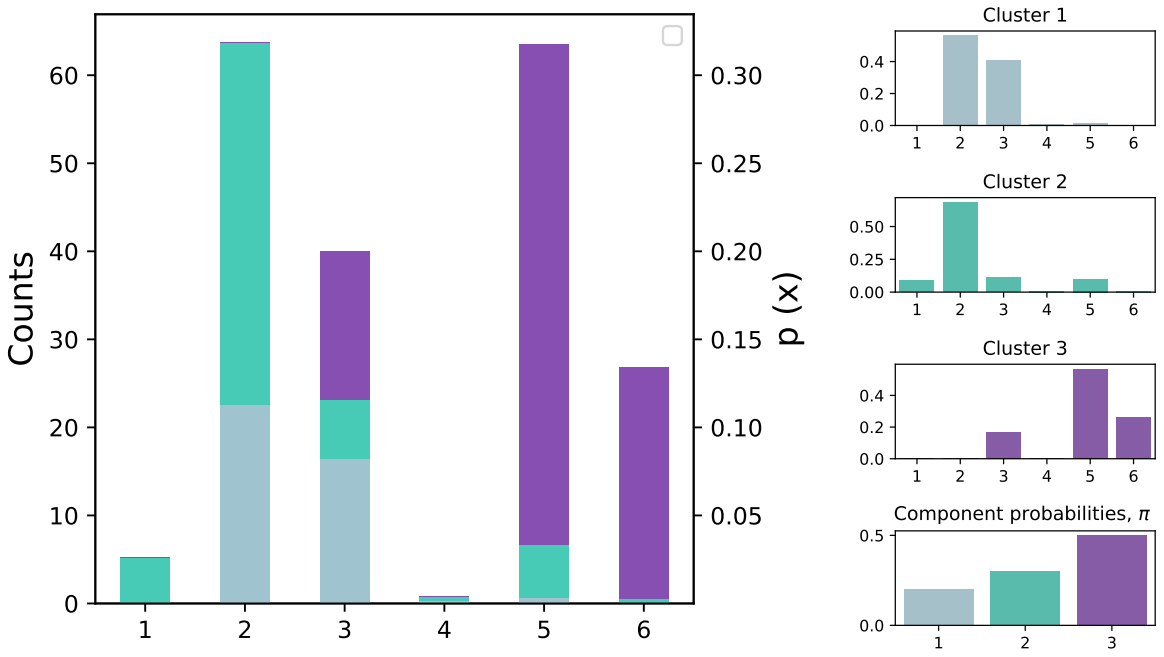
A finite mixture model assumes that each data point $\boldsymbol{x_n}$ is generated by a two-step process:

1. Select k, one the K components, with the component probability π_k such that $\pi_k \geq 0$ and $\sum_K \pi_k = 1$
2. Sample x_n from the distribution of the kth component, with parameters θ_k , that is from $p(\boldsymbol{x_n}|k)$

The joint probability distribution of a mixture model with K components given the model parameters, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ can be expressed by

$$p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\theta_k) \tag{3.5}$$

To illustrate this data generating process, imagine we have ten dice which belong to one of three subsets, $\{A, B, C\}$ with 5, 3, and 2 dice respectively giving component probabilities $\boldsymbol{\pi} = [0.5, 0.3, 0.2]$. Each of the subsets of dice has a different loading, or probability distribution over faces, $\boldsymbol{\theta^A}, \boldsymbol{\theta^B}, \boldsymbol{\theta^C}$. We randomly select and throw a die and record the throw replacing the die. If we sample a die 200 times in this way and record how many times each face is thrown we will get a distribution of counts over the faces. Figure 3.6.a below shows an example of count data generated this way from the model parameters shown in the left-hand panel.



(a) (b)

Figure 3.6: **The multinomial mixture model.**
a) The count data shows the totals of the observed counts on the left-hand axis, the proportion of counts that come from each cluster indicated by the colour (unknown), and the whole mixture model probability of throwing each face on the right-hand axis.
b) Model parameters, the multinomial probabilities for each cluster $\boldsymbol{\theta_A}, \boldsymbol{\theta_B}, \boldsymbol{\theta_C}$ (clusters 1,2, and 3) and cluster probabilities $\boldsymbol{\pi}$.

The goal of mixture models is to infer the unknown or latent parameters of the model from the data, in our toy example above these are $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$. One approach is to use the Expectation-Maximization (EM) algorithm to obtain a deterministic non-random estimates of the model parameters. This a two step iterative method to learn the maximum likelihood or maximum a priori (MAP) estimates of the parameters.

3.3.5 Bayesian inference for mixture models

In a Bayesian framework, both the data and the model parameters are assumed to be random variables. The first step in Bayesian inference is to construct the probabilistic model that we assume generated the observed data. Then we infer or learn the model parameters from the observed data. We interpret the variables in Bayes rule3.2 as the model parameters $\boldsymbol{\theta}$ and the observed data $\boldsymbol{X}$. We use this to update our prior beliefs about the distribution of the parameters, $p(\boldsymbol{\theta})$, with the likelihood of the data given those parameters, $p(\boldsymbol{X}|\boldsymbol{\theta})$ to get the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X})$.

3.3.6 Gibbs sampler for a Dirichlet-multinomial mixture models

Next we define Dirichlet multinomial mixture model (DMMM) and will show how we can sample the conditional model parameters. We will then show that by using Gibbs sampling we are able to remove the model parameters we are not interested in, enabling us to create an infinite MM where we do not have to specify the number of components in the model. It is this infinite DMMM that forms the core of the model used in Chapter 6.

The Dirichlet multinomial distribution has been successfully used to model categorical data in many domains and has been shown to be highly sensitive for the analysis of microbiome and other ecological count data (Harrison et al. 2019). In Chapter 6, we develop a multi view clustering approach that is based on a work by Paul Kirk (Kirk and Richardson 2021) that is built on multiple DMMMs. We use these DMMMs to attempt to capture the complex mixtures in the kmers counts that we have used as features throughout this thesis.

Gibbs sampling is a Markov Chain Monte Carlo algorithm for sampling the posterior probabilities of the model parameters from a complex joint distribution such as a mixture model (Gelfand and Smith 1990; Neal 2000). This enables us to infer the model

parameters when the posterior distribution cannot be derived analytically. It reduces the complex problem of sampling from a joint distribution to sampling each model parameter in turn from the probability of that parameter conditioned on all other variables. When sampling models based on conjugate pairs, such as a Dirichlet multinomial, we are able to marginalise or collapse some of the parameters. By removing model parameters, the collapsed Gibbs sampler(CGS) reduces the sampling space, thereby improving efficiency and accuracy. Furthermore, CGS allows us to build a sampler for an infinite mixture model which removes the need to specify the number of components in the mixture model.

Model Definition: The data $\boldsymbol{X}$, consists of the observations for N individuals. Each data point x_n is a single sample from a multinomial with L levels. For now, we assume that data is univariate, but this can easily be extended to multivariate data under the assumption that the covariates are independent and probability for J independent events is given by $P(x_1, ..., x_J) = \prod_{j=1}^{J} P(x_j)$

We assume that each data point originates from a single cluster k sampled from one of K clusters, with probability π_k. Each cluster is defined by a multinomial with probability f_{kl} where $\sum_{l=1}^{L} f_{kl} = 1$ so that $\boldsymbol{f_k}$ is a vector of length L and $\boldsymbol{f_k} \sim Dir(\beta)$

$$p(\boldsymbol{x_n}|\boldsymbol{f_k}) = \prod_{l=1}^{L} (f_{kl})^{(x_n=l)}$$

where $(x_n = l) = 1$ if $x_n = l$ and 0 otherwise.

We introduce an indicator parameter $\boldsymbol{Z}$ to track cluster membership, defined as a set of $N \times K$ binary parameters z_{nk} where $z_{nk} = 1$ if an individual n belongs to the kth cluster and 0 otherwise $\boldsymbol{z_n} = [z_{n1}, ..., z_{nK}]$ and $\sum_K z_{nk} = 1$.
The prior distribution for $\boldsymbol{z_n}$ is the multinomial $\boldsymbol{\pi} = [\pi_1, ..., \pi_K]$ where $\sum_K \pi_k = 1$ and $p(\boldsymbol{\pi}) \sim Dir(\alpha)$.

We have three sets of unknown parameters that we want to learn, the cluster priors $\boldsymbol{\pi}$, multinomial parameters $\boldsymbol{F}$, and the cluster membership $\boldsymbol{Z}$. Gibbs sampling will give us samples of each of these sets of model parameters from the joint posterior distribution.

Some distributions implicit from our model definition. Using Bayes rule given by equation 3.2 and ignoring the denominator which is a normalising constant we can express the full joint posterior distribution as

$$P(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{F}, \boldsymbol{X}|\alpha, \beta) \propto P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{F})P(\boldsymbol{Z}|\boldsymbol{\pi})P(\boldsymbol{\pi}|\alpha)P(\boldsymbol{F}|\beta) \tag{3.6}$$

$$p(\boldsymbol{Z}|\boldsymbol{\pi}) = \prod_n \prod_k \pi_k^{z_{nk}} \tag{3.7}$$

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{F}) = \prod_n \prod_k \left[\prod_l f_{kl}^{x_n=l} \right]^{z_{nk}} \tag{3.8}$$

$$p(\boldsymbol{F}_k|\beta) = Dir(\boldsymbol{f}_k|\beta) = \prod_l f_{kl}^{\beta-1} \tag{3.9}$$

To sample from the full joint distribution we sample one of the unknown parameters Z, $\boldsymbol{\pi}$, $\boldsymbol{F}$ at a time, conditioning on all other parameters. This is done in three steps:

1. Sample $\boldsymbol{\pi}|Z, \alpha$
   Removing all terms that are not dependant on $\boldsymbol{\pi}$ from from equation 3.6 we are left with
   $$p(\boldsymbol{\pi}|\boldsymbol{Z}, \alpha) \propto P(\boldsymbol{Z}|\boldsymbol{\pi})P(\boldsymbol{\pi}|\alpha)$$
   $$p(\boldsymbol{\pi}|...) \propto \prod_k \pi_k^{\sum z_{nk}} \prod_k \pi_k^{\alpha-1}$$
   As shown above the conditional conjugate pair means that the posterior $P(\boldsymbol{\pi}|...)$ must be another Dirichlet
   $$p(\boldsymbol{\pi}|\boldsymbol{...}) = Dir\left(\alpha + c_1, ..., \alpha + c_K\right) \tag{3.10}$$
   Where we use c_k to replace $\sum_n z_{nk}$,the count of individuals in the kth cluster.

2. Sample $F|\boldsymbol{Z}, \beta, \boldsymbol{X}$
   Removing terms not dependant on F leaves us with
   $$p(\boldsymbol{f_k}|...) \propto P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{F})p(\boldsymbol{f_k}|\boldsymbol{\beta}) \prod_k Dir(\boldsymbol{f}_k|\beta)$$
   Where the terms are given by equations 3.8 and 3.9 above. Again, we have a conjugate pair which will result in another Dirichlet.

$$p(f_k|...) \propto \prod_n \left[\prod_l f_{kl}^{x_n=l} \right]^{z_{nk}} \prod_l f_{kl}^{\beta-1}$$

We will use θ_{kl} to replace $\sum_n z_{nk}(x_n = l)$,the counts of number of data points in cluster k equal to l,i so that the conditional probability for $\boldsymbol{f}_k$ is an L-vector, the multinomial probabilities for cluster k.

$$p(f_k|...) = Dir(\beta + \theta_{k1}, ..., \beta + \theta_{kL}) \tag{3.11}$$

3. Sample $\boldsymbol{Z}|\boldsymbol{\pi}, \boldsymbol{F}, \boldsymbol{X}$

To sample the component allocation of each data point n, $z_n = 1$ conditioned all other allocations we compute probabilities of the observation occurring in each component and then draw a single component from the Categorical distribution given by these probabilities. Removing all terms from equation 3.6 that are not dependant on $\boldsymbol{z}_n$ we are left with

$$P(z_n = k, |\boldsymbol{\pi}, \boldsymbol{F}, \boldsymbol{X}) \propto P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{F})P(\boldsymbol{Z}|\boldsymbol{\pi})$$

$$p(z_n = k|...) \propto \pi_k \prod_l^L (f_{kl})^{x_n=l} \tag{3.12}$$

The algorithm for a standard Gibbs sampler is shown below. We collect the generated samples after each iteration and these collections can then be used to estimate the posterior probability of the latent model parameters.

Initialisation: Randomise $\boldsymbol{Z}$ such that $z_{nk} \in \{0, 1\}$ and $\sum_k z_{nk} = 1 \quad \forall n$;
for *the required number of iterations* **do**
  1. Resample $\boldsymbol{\pi}$ with equation 3.10
  2. Resample $f_{kj} \quad \forall\, k, j$ with equation 3.11
  for n *in* N **do**
    3. Resample z_n with equations 3.12 and
  end
end

Algorithm 1: Standard Gibbs sampling scheme

3.3.7 Collapsed Gibbs for DMMM

Finally we derive the equations for a collapsed Gibbs sampler for a Dirichlet multinomial mixture model, we use this scheme in Chapter 6 to sample the infinite mixture models that make up our MVC model. In the previous section we have derived the equations for a standard Gibbs sampler for sampling the model parameters from the joint distribution of DMMM. In clustering we are mainly interested in the individual to cluster allocations $\boldsymbol{Z}$. One goal is to have a model in which we do not have to specify the number of components meaning that K becomes a another latent parameter inferred from the data. To do this, we need to remove the parameters $\boldsymbol{\pi}$ and $\boldsymbol{F}$ from our sampler. Consider re-sampling the cluster membership of the nth individual, $\boldsymbol{z_n}$ and using use $\boldsymbol{Z}^{-n}$ mean all the assignments except for current individual, we can express the conditional probability of this assignment as

$$P(z_{nk} = 1, \boldsymbol{\pi} | \boldsymbol{Z}^{-n}, \boldsymbol{X}, \boldsymbol{F}, \alpha, \beta) \propto p(\boldsymbol{x}_n | z_{nk} = 1, \boldsymbol{F}) p(z_{nk} = 1 | \boldsymbol{\pi}) p(\boldsymbol{\pi} | \boldsymbol{Z}^{-n}, \alpha) \tag{3.13}$$

Marginalise $\boldsymbol{\pi}$ To marginalise $\boldsymbol{\pi}$ from equation 3.13 we integrate over π

$$P(z_{nk} = 1 | \boldsymbol{Z}^{-n}, \boldsymbol{X}, \boldsymbol{F}, \alpha, \beta) \propto p(\boldsymbol{x}_n | z_{nk} = 1, \boldsymbol{F}) \int p(z_{nk} = 1 | \boldsymbol{\pi}) p(\boldsymbol{\pi} | \boldsymbol{Z}^{-n}, \alpha) d\pi$$

To be able to update $\boldsymbol{z}_n$ without using π we need to evaluate the integral on right hand side of this equation. The first term is just π_k since $\boldsymbol{z}_n$ has a one in the kth element. The second term is the Dirichlet as derived in equation 3.10 but with the z_{nk} removed from the counts of individuals in clusters, denoted by C^{-n}. To make things clearer to read we will use $\alpha'_k = \alpha_k + c_k^{-n}$

$$P(z_{nk} = 1 | ...) = p(\boldsymbol{x}_n | z_{nk} = 1, \boldsymbol{F}) \frac{\Gamma(\sum_j \alpha'_j)}{\prod_j \Gamma(\alpha'_j)} \int \pi_k \prod \pi_k^{\alpha'_j - 1} d\pi$$

If we define $\delta_{jk} = 1$ if $j = k$ and 0 otherwise we can rewrite the integral

$$P(z_n = 1 | ...) = p(\boldsymbol{x}_n | z_{nk} = 1, \boldsymbol{F}) \frac{\Gamma(\sum_j \alpha'_j)}{\prod_j \Gamma(\alpha'_j)} \int \prod \pi_k^{\alpha'_j + \delta_{jk} - 1} d\pi$$

Given that we have shown that the posterior should take the form of a Dirichlet the integral looks like an unnormalised Dirichlet with parameters $\alpha'_j + \delta_{jk}$. For the integral to evaluate to 1 (As pdf) the constant must be the inverse of the Dirichlet constant. Our

expression for the posterior probability for $P(z_n k = 1|...)$ becomes

$$P(z_{nk} = 1|...) = p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F})\frac{\Gamma(\sum_j \alpha'_j)}{\prod_j \Gamma(\alpha'_j)}\frac{\prod_j \Gamma(\alpha'_j + \delta jk)}{\Gamma(\sum_j \alpha'_j + \delta jk)}$$

We can simplify this equation, firstly by using properties of δ so that $\sum_{j=1}^{K} \delta_{jk} = 1$ means that $\Gamma(\sum_j \alpha'_j + \delta jk) = \Gamma(1 + \sum_j \alpha'_j)$ and secondly because $\delta_{jk} = 0$for all $j \neq k$ then the product terms in the denominator cancel except when $j = k$. This leaves

$$P(z_{nk} = 1|...) = p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F})\frac{\Gamma(\sum_j \alpha'_j)}{\Gamma(\alpha'_k)}\frac{\Gamma(\alpha'_k + 1)}{\Gamma(1 + \sum_j \alpha'_j)}$$

Next using the property of a gamma function that $\Gamma(x+1) = x\Gamma(x)$ we can cancel further terms

$$P(z_{nk} = 1|...) = p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F})\frac{\Gamma(\sum_j \alpha'_j)}{\Gamma(\alpha'_k)}\frac{\alpha'_k\Gamma(\alpha'_k)}{\sum_j \alpha'_j\Gamma(\sum_j \alpha'_j)} = \frac{\alpha'_k}{\sum_j \alpha'_j}$$

If we substitute $\alpha'_k = \alpha_k + c_k^{-n}$ to get

$$P(z_{nk} = 1|...) = p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F})\frac{\alpha_k + c_k^{-n}}{\sum_j^K \alpha_k + c_k^{-n}}$$

The final equation for the conditional probability of individual $\boldsymbol{x}_n$ conditioned on all other individuals after marginalisation of pi is given by

$$P(z_{nk} = 1|...) \propto p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F})\frac{\alpha_k + c_k^{-n}}{\sum_j^K \alpha_k + c_k^{-n}} \tag{3.14}$$

This can be used in our sampling algorithm 1 and removes the need to re-sample π in step 1.

Marginalising F, the multinomial parameters for the data. Because we are using a multinomial which is drawn from Dirichlet priors we can repeat the same process to marginalise F from the conditional distribution $P(z_{nk} = 1)$. Starting with the equation

3.13 above we are now interested in all terms related to $\boldsymbol{F}$

$$P(z_{nk} = 1|...) \propto \int p(\boldsymbol{x}_n|z_{nk} = 1, \boldsymbol{F}_k)p(\boldsymbol{F}_k|\boldsymbol{X}^{-n}, \boldsymbol{Z}^{-n}, \beta)d\boldsymbol{F}$$

The first term of the integral is the likelihood of a single data point x_n and the second term is the Dirichlet with parameters $Dir(\theta_l^{-1} + \beta, ..., \theta_L^{-1} + \beta)$ where θ_{kl}^{-1} the counts of number of data points in cluster k equal to l without the current data point and replaces $\sum_m z_{mk}(x_m = l)$

$$P(z_{nk} = 1|...) \propto \int \prod_l f_{kl}^{x_n=l} \times \prod_l f_{kl}^{\theta_{kl}+\beta-1} d\boldsymbol{F}$$

$$P(z_{nk} = 1|...) \propto \int \prod_l (f_{kl})^{(\beta_l+\theta_{kl}-1)} d\boldsymbol{F}$$

By following the same steps as for marginalising $\boldsymbol{\pi}$ we end up with an final expression in which both $\boldsymbol{F}$ and $\boldsymbol{\pi}$ have been marginalised out

$$P(z_{nk} = 1|...) \propto \frac{\alpha_k + c_k^{-n}}{\sum_j^K \alpha_k + c_k^{-n}} \cdot \frac{\beta_l + \theta_{kl}^{-n}}{\sum_m^L \beta_m + \theta_{km}^{-n}} \tag{3.15}$$

This is the conditional posterior probability of an individual n being assigned to the kth cluster and can be used in a collapsed Gibbs sample where only step 3 of the Gibbs algorithm given above is performed using this equation (3.15). This simplified expression means that we just need to keep a track of the counts of individuals in clusters and of the number of data points at each level in each cluster.

3.3.8 Infinite DMMM

One of the big advantages of using a collapsed Gibbs sampler is that we no longer need to specify the number of clusters in the model. In this section, we will show how the expression for the posterior probability for the allocation of an individual z_n to cluster k derived above can be used to make an infinite mixture model (Rasmussen 1999). First, we assume that our prior parameters $\boldsymbol{\alpha}$ are all the same value and set this to $\frac{\alpha}{K}$ so that our sampling prior becomes

$$P(z_{nk} = 1|\alpha, \boldsymbol{Z}^{-n}) = \frac{\frac{\alpha}{K} + c_k^{-n}}{N - 1 + \alpha}$$

If we have an infinite number of components $K = \infty$ so that $\frac{\alpha}{\infty} = 0$ then the probability that the nth data point goes into one of the occupied components is given by

$$P(z_{nk} = 1|\alpha, \boldsymbol{Z}^{-n}) = \frac{c_k^{-n}}{N - 1 + \alpha}$$

We can compute the probability that data point n does not go in a currently occupied component because the total probability must equal 1. Using k_* to denote an empty cluster

$$P(z_{nk_*} = 1|\alpha, \boldsymbol{Z}^{-n}) = 1 - \sum_{k=1}^{K} \frac{c_k^{-n}}{N - 1 + \alpha} = 1 - \frac{N-1}{\alpha + N - 1} = \frac{\alpha}{\alpha + N - 1}$$

$$P(z_{nk_*} = 1|\alpha, \boldsymbol{Z}^{-n}) = \propto \begin{cases} \frac{c_k^{-n}}{\alpha+N-1} & \text{for currently occupied clusters} \\ \frac{\alpha}{\alpha+N-1} a & \text{for currently empty clusters} \end{cases} \tag{3.16}$$

The Gibbs sampling equations for the posterior probability for individual n being assigned to cluster k for an infinite Dirichlet multinomial mixture model are

$$P(z_{nk} = 1|...) \propto \frac{c_k^{-n}}{\alpha + N - 1} \times \frac{\theta_{kl}^{-n} + \beta_l}{\sum_{m=1}^{L} \theta_{km}^{-n} + \beta_m} \text{ for existing clusters} \tag{3.17}$$

$$P(z_{nk} = 1|...) \propto \frac{\alpha}{\alpha + N - 1} \times \frac{\beta_l}{\sum_{m=1}^{L} \beta_m} \text{ for new clusters} \tag{3.18}$$

This can easily be extended to multivariate data where each data point consists of J independent covariates such that $\boldsymbol{x}_n = x_1, ..., x_J$ and each covariate is drawn from separate multinomials f_{kjl}. Given that the joint probability for J independent events is given by

$$P(x_1, ..., x_J) = P(x_1) \times ..., \times P(x_J) = \prod_j P(x_j)$$

$$P(z_{nk} = 1|...) \propto \frac{c_v^{-n}}{\alpha + N - 1} \times \prod_j^J \frac{\theta_{kjl}^{-n} + \beta_l}{\sum_m^L \theta_{kjm}^{-n} + \beta_m} \text{ for existing clusters} \tag{3.19}$$

Chapter 4

Predictive features in viral genomes

This chapter is a copy of the paper, :

Young, Francesca, Simon Rogers, and David L. Robertson. “Predicting host taxonomic information from viral genomes: A comparison of feature representations”. In: PLOS Computational Biology 16.5, 2020".

My contribution to this paper: I contributed to the design of the study with my co-authors, I wrote all the code and carried out the analysis and wrote the draft manuscript.

4.1 Introduction

Determining which virus infects which host species is currently a major challenge in virology. Currently, there are no high-throughput methods available to make reliable virus-host associations and as such we are unable to keep up with the rapid pace of viral discovery. Fast, accurate computational tools are thus urgently needed to annotate these new viral genomes with host taxon information. Machine learning is an ideal approach but to maximise predictive accuracy the viral genomes need to be represented in a format (sets of features) that makes the discriminative information available to the machine learning algorithm. Here, we investigate different features derived from alternative representations of viral genomes for their ability to predict host information.

Computational approaches to virus host prediction fall into four broad strategies: searching for homologous sub-sequences in the hosts, such as prophage (Roux et al. 2015) or

CRISPR-Cas spacers (Edwards et al. 2016); looking for co-abundance between virus and host (Dutilh et al. 2014); distance based metrics of oligo-nucleotide or kmer composition, either with potential host genomes (Edwards et al. 2016; Galiez et al. 2017; Ahlgren et al. 2017), or with reference virus genomes (Villarroel et al. 2016); and machine learning methods using a variety of sequence derived features as described below. Although the first strategy can give high confidence predictions, the predictions are constrained by the limits of alignment approaches at low sequence similarity. Kmer profile comparison provides alignment free methods but because of lack of contrast when measuring proximity in high dimensional space they lose discriminative power. Additionally, all methods that rely on reference genomes are constrained by the genomes available in the databases.

Machine learning approaches offer alternatives that are not dependent on reference genomes or alignment, relying instead on a set of labeled training examples. To date, most machine learning approaches to virus host prediction have used features derived from oligo-nucleotide or kmer biases that are known to correlate with their host genomes, such as: CG bias (Mihara et al. 2016), CpG bias (Atkinson et al. 2014; Goff 2017; Simmonds et al. 2013) and di-codon bias (Carbone 2008). Di-nucleotide features, in particular, have been included in a wide range of virus host prediction tasks, from training on a single virus species or genera with multiple hosts such as rabies virus, coronavirus, and influenza A virus (Tang et al. 2015; Li and Sun 2018; Kapoor et al. 2010), to training on host taxa with multiple viruses (Babayan et al. 2018; Gałan et al. 2019). The potential for improved prediction by extending the length of the nucleotide kmers has been demonstrated by Zhang et al. (Zhang et al. 2017a). The nucleotide sequence contains the information needed for a virus to exploit its host: regulatory RNAs, amino acid sequences etc. The latter, through their biochemical properties, fold into three dimensional structures with functional properties mediated through molecular interactions. Although all of this 'functional' information is present in the nucleotide sequence, it is not necessarily in a form that is easy for machine learning approaches to extract. Only two machine learning approaches have previously demonstrated the potential of using alternative representations of the genome for virus host prediction: Raj and co-workers Raj et al. (2011) successfully used amino-acid kmers to predict the host kingdom of two RNA virus families, while Leite and co-workers Leite et al. (2018) included predicted domain-domain interactions in their features to predict phage-bacteria interacting pairs.

The phylogenetic signal due to the evolutionary relationship between viruses that infect the same host can also be predictive. babayanPredictingReservoirHosts2018 successfully made use of this signal by combining a measure of 'phylogenetic neighbourhood' with

selected features derived from nucleotide biases to predict the reservoir host of newly emerging viruses. Alignment free phylogenetic analysis (Zhang et al. 2017b) has shown that kmer composition contains sufficient phylogenetic signal to reliably infer evolutionary relationships. Protein domains have also been used to infer phylogeny, for example, Phan et al. (Phan et al. 2018) recently used domains to classify newly discovered coronaviridae genomes.

The aim of this chapter was to investigate the predictive power of different features sets derived from the alternative transformations of nucleotide sequences as described in Section 2.5. This is based on the hypothesis that these different representations of the viral genomes have the potential to improve prediction over nucleotide sequences as they make the complex nature of both the evolutionary and host-mimicry information more easily accessible to machine learning algorithms. We used the supervised machine learning workflow shown in (Figure 4.1) to compare the performance of SVM classifiers trained on the different feature sets to predict host taxonomic information for both prokaryote and eukaryote hosts.

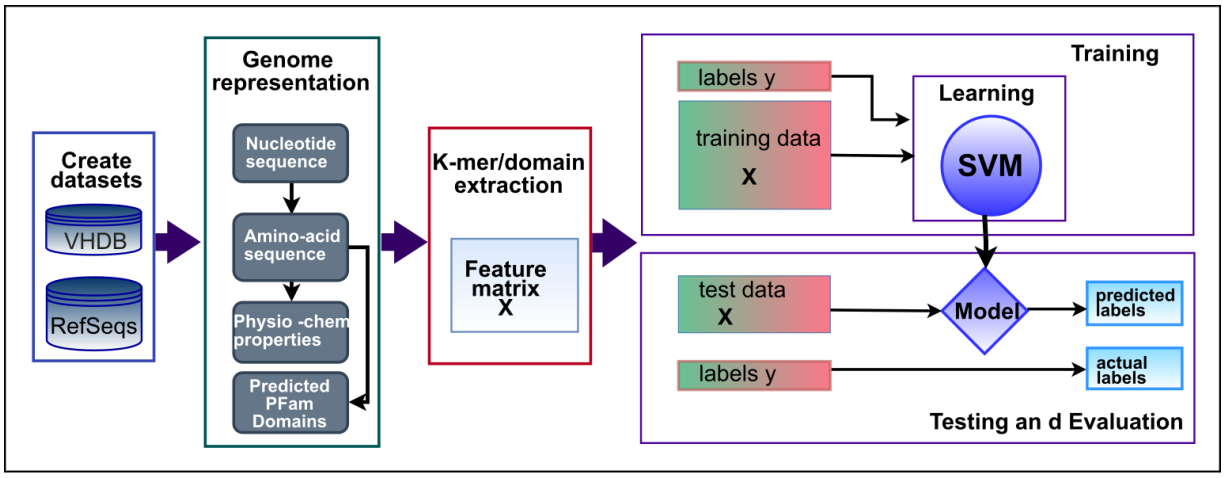


Figure 4.1: Workflow for extracting and testing different level feature sets for predicting host taxon information. Virus genome data was represented by four information layers and features derived from each. Binary classification with linear SVM was used on the equal sized positive and negative classes of virus-host association, split into training and test sets. Area under the ROC curve, AUC, score was measured for each dataset-feature set combination.

4.2 Methods

4.2.1 Data

We downloaded the Virus Host Database (https://www.genome.jp/virushostdb/) (Mihara et al. 2016) on 25/1/2019. The VHDB is a curated database of reported taxonomic interactions between viruses and their known hosts. It is regularly updated from Refseq/GenBank, Uniprot and Viralzone and includes manual annotations. The dataset included 9199 unique viruses associated with 3006 hosts and a total of 14229 interactions. The FASTA files of the reference genome sequences and the amino acid sequences of the coding regions for each virus are also included in the VHDB resources.

4.2.2 Generating Binary Datasets from the known Virus Host Interactions

A host taxonomic tree was constructed from all the hosts in VHDB using ETE3 (Huerta-Cepas et al. 2016) at the ranks of kingdom, phylum, class, order, family, genus and species. Each host node was annotated with the viruses known to infect it. The tree was 'pruned' to include only nodes infected by at least a minimum number of virus species. The minimum number of infecting viruses was set to 28 for a positive node. As we were comparing how predictive these feature sets were across all taxon ranks, setting this arbitrary threshold at 28 enabled us to include more examples of genus and species level datasets. For each binary dataset the positive class consisted of the viruses that infect a host node while the negative class contained an equal number of viruses selected from those that infect the rest of the hosts in its parent node, i.e., the node's most closely related hosts. For example, for primates the order primates made up the positive class, the viruses to form the negative class were selected from those that infect at least one host in the rest of the taxon class mammalia (Figure 4.2). In cases where the negative class comprised fewer than 28 viruses the class was widened to include the next taxa up until at least 28 viruses are present. This resulted in binary datasets of equal numbers of positive and negative viruses with a minimum dataset size of 56 viruses. These were then split into training and test data at a ratio of 0.8 to 0.2 respectively (or 0.75 to 0.25 where the total number of viruses in the dataset was less than 100).

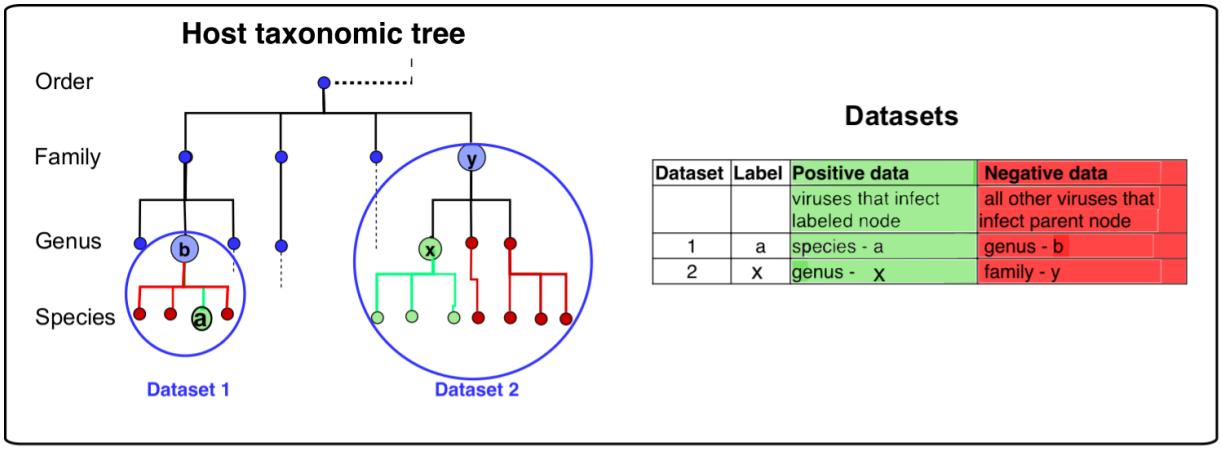


Figure 4.2: Generating datasets from the host taxonomic tree.
Datasets were generated from a taxonomic tree of all the hosts with more than 28 known infecting virus species. For each node the positive class consisted of the viruses that infect the labelled node while the negative viruses were selected from those that infected the rest of the taxon group of that node, for example, if the genus x made up the positive class, the viruses to form the negative class were selected from those that infect the rest of the genera in family y.

4.2.3 Genome representation

Four different representations of the genome were used: 1. Nucleic acid sequence kmers (DNAk) and their reverse complements were extracted from the raw genome sequence in the FASTA files. Segmented viruses were concatenated and treated as a single genome. 2. Amino acid sequence kmers (AA_k) were extracted from the amino acid sequence of the coding regions in the downloaded FAA files, (segmented viruses were again concatenated). 3. Physio-chemical properties of the amino acid sequence, kmers (PC_k) were extracted by first binning each amino acid into one of seven groups defined by their physio-chemical properties: (AGV, C, FILP, MSTY, HNQW, DE, and KR) (Shen et al. 2007). The kmers were then extracted using the seven bin labels as the alphabet. 4. Domains, the domain content for each genome was identified using the HMMER package (version HMM 3.2) (Eddy 2011; Finn et al. 2011). First a Pfam domain profile database was built using Pfam-A (release 31.0.), then all the amino acid sequences from each genome were scanned against this using the hmmscan command resulting in a list of predicted domains for each virus genome. For each genome in a dataset the domain composition vector was generated by counting the frequency of the unique domains in each virus across the set of all unique domains found within that dataset. These vectors were then normalised to sum to 1. The Hmmscan setting: $--cut_tc$ option was used, this uses the trusted bit score thresholds

from the model. The aim being to include the maximum number of possible domains with the expectation that the machine learning will be able to find the true signal above the noise.

4.2.4 Kmer extraction

The DNA_k, AA_k and PC_k sequence data were represented as a vector of kmer composition. These were generated by counting the number of times each possible kmer occurs within the sequence (and the reverse complement of the nucleotide sequences), and then normalising the resulting vector to sum to 1 to account for varying genome lengths. Zhang et al. (2017) showed that this simple method of representing kmer composition was as effective for predicting host as more complex representations that included background models of the sequence (Zhang et al. 2017a). The maximum length of kmers at each genome representation level was chosen to restrict the feature set size and keep the workflow computationally reasonable.

4.2.5 Supervised Classification

Linear support vector machine classifiers (SVM) were used for classification. The SVM is well suited to binary classification problems where the number of data points is much smaller than the number of features used (Cortes and Vapnik 1995). It is also able to cope well with sparse feature matrices where many of the elements are zero. Classifiers were trained with the training data from each dataset and then tested for the predictive power of features on the unseen test data. No optimisation steps were performed and a linear kernel was chosen for speed. The python library, Scikit-learn (Pedregosa et al. 2011), was used with a pipeline that included feature scaling with Standard Scaler. The penalty parameter, C, was left at its default value of 1.0.

Classifier performance was measured using the area under the ROC curve (AUC) score. AUC is a suitable metric for binary classifiers, giving a measure of both specificity and sensitivity without the need to set a threshold. Using a single metric makes it possible to compare the predictive power of the features across the large number of classifiers we tested. The specificity and sensitivity for each classifier are included in the supplementary tables of the results.

4.2.6 Creating 'holdout' datasets

To separate the virus and host specific signals our aim was to remove the phylogenetic signal coming from the most closely related viruses from the training data. Holdout datasets were created by first removing a 'related' group (or groups) of viruses from training data and then using these holdout viruses as the test data (Figure 4.8). Because the ICTV taxonomy does not capture phylogenetic family level relationships, a further step was taken to remove any viruses from the training dataset. We used average nucleotide identity (ANI), a similarity measure between pairs of genomes to filter the training sets. Viruses that had more than 75% ANI to any of the holdout viruses were removed from the training data. An ANI matrix of all against all phage was generated using FastANI (Jain et al. 2018), this matrix was then used to filter viruses that were greater than 75% ANI over 10% AF to any of the "holdout" viruses. FastANI was used with a minFrac=0.1 (minimum alignment fraction of genomes) reduced from default of 0.2 as viruses share few common genes and fragment length of 1000, by reducing this from the default of 3000 we aimed to increase the number of shared fragments found.

Again both the training and test/holdout data contained equal numbers of positive and negative viruses that infected hosts at bacterial host taxa at phylum and order level that had interactions with multiple virus groups at family level. These 'holdout' classifiers were then compared with the standard 'all' classifiers - generated as described above - using a subset of the most predictive feature sets. DNA: $k = [2, 6, 9]$, $AA : k = [3, 4]$, $PS : k = [5, 6]$ and domains, (DNA_2 was included for comparison).

4.2.7 Kernel Combination

The goal of this study was to investigate whether using features extracted from different levels of viral genome representation were predictive of host. To check whether these features are redundant we combined the features from the different genome levels. As SVM is a kernel method we can combine the kernels from the different classifiers in linear combinations to generate a new kernel(Lanckriet et al. 2004). We tested a range of kernels on a single dataset by using different weights. $K = \sum_{i=1}^{4} w_i k_i$ where $\sum_{i=1}^{4} w_i = 1$ Where k_1 to k_4 are kernels derived from each genome representation: DNA_9, AA_4, PC_5 and Domains respectively.

4.3 Results

4.3.1 Data

Datasets for different host taxa, or labels, were created using sequences for positive and negative viruses, that is, viruses that are either known or not known to infect the labelled taxa. To ameliorate the problems caused by overlapping or redundant data we kept a minimum distance between sequences by using only the reference sequence for each viral species. The Virus Host Database (Mihara et al. 2016) was used to identify known species-level virus host interactions for both prokaryote and eukaryote hosts at different host taxonomic levels. We created a balanced binary dataset for each host taxa for which there were more than a minimum number of known interacting viruses. Known interactions made up the positive labelled class. The negative class was drawn from the remaining viruses that infect hosts in the parent taxa of the positive class. By setting a low threshold of 28 viruses as the minimum class size, giving a total dataset size of 56, we were able to analyse multiple datasets at species and genus level. For the prokaryote hosts this resulted in 65 datasets (all for bacteria hosts - which we refer to these as the 'bacteria datasets'), corresponding to Baltimore class I, dsDNA viruses.

For the eukaryote hosts, this procedure resulted in very few host taxa below class reaching our threshold for minimum dataset size. We therefore used the following two strategies: combining viruses from all Baltimore classes; and combining all RNA viruses, respectively, for a particular host taxa into a single dataset. This resulted in a total of 116 eukaryote datasets covering 57 host taxa over all taxonomic ranks, from kingdom to species level and the different Baltimore, and combined classes of the viruses. These include 48 datasets comprising all RNA viruses for host groups that include many at family, genus and species level.

Each of the 224 datasets was randomly split into training and test partitions with a ratio of 0.8 to 0.2 prior to extracting the 20 different feature set matrices from the viral genomes, see Table 4.1. Each of these feature matrices was used to train and test an SVM classifier, resulting in AUC scores for over 3740 classifiers.

Table 4.1: The 20 feature sets generated from the four representations of the viral genomes.

Genome representation	[Letter set] (Alphabet size)	kmers (k) lengths tested	Feature set size for maximum k
DNA	[A,C,G,T] (4)	[1-9]	262,144
AA	[Amino Acid single letter code] (20)	[1-4]	160,000
PC	[t-z] (7)	[1-6]	117,649
Domains	[All unique domains predicted in each dataset]	1	Total number of unique domains in all the viral genomes in VHDB 2200

4.3.2 All features levels are predictive of host across all hosts

To test the predictive capacity of the different levels of the genome representation we trained and tested a binary classifier for each of the 20 feature sets on all of the datasets described above. The results of the evaluation of all the classifiers demonstrate that all levels of genome representation contain a signal predictive of host taxa across the host tree. Heatmaps comparing the AUC scores for the prokaryote (Figure 4.3 and eukaryote (Figure 4.4) classifiers show that apart from DNA kmers of length 1, all feature sets are consistently predictive. In particular, omitting results for DNA kmers of length 1, 82% of the dataset-feature set combinations have an AUC of 0.75 or more (74% with AUC of 0.8 or more) Any AUC score above 0.5 (random classification) indicates the presence of a predictive signal. A score of 1 demonstrates the potential for a perfect classifier where all predictions are correct. Most host taxa have many feature sets that contain a predictive signal (146 out of the 180 datasets have at least one feature set with a score of greater than 0.90). Some hosts are more challenging to predict with none of the feature sets giving good performance, (6 out of the 180 datasets have a maximum score of less than 0.80). This is most apparent at the lower taxonomic ranks of species and genus where we are trying to separate the viruses of more similar hosts and for some Baltimore classes. Overall, the results show that a genomic signature that predicts host taxonomy is present at all levels of biological information representation tested in our study.

Figure 4.4 shows a Comparison of the results for all the eukaryote datasets across all the feature sets and for all Baltimore groupings (indicated by the inner colour bar on the right). Legend. The heatmap shows that most of the feature sets contain some predictive signal, AUC $>$ 0.5, for the majority of the eukaryote datasets and for all Baltimore groupings (

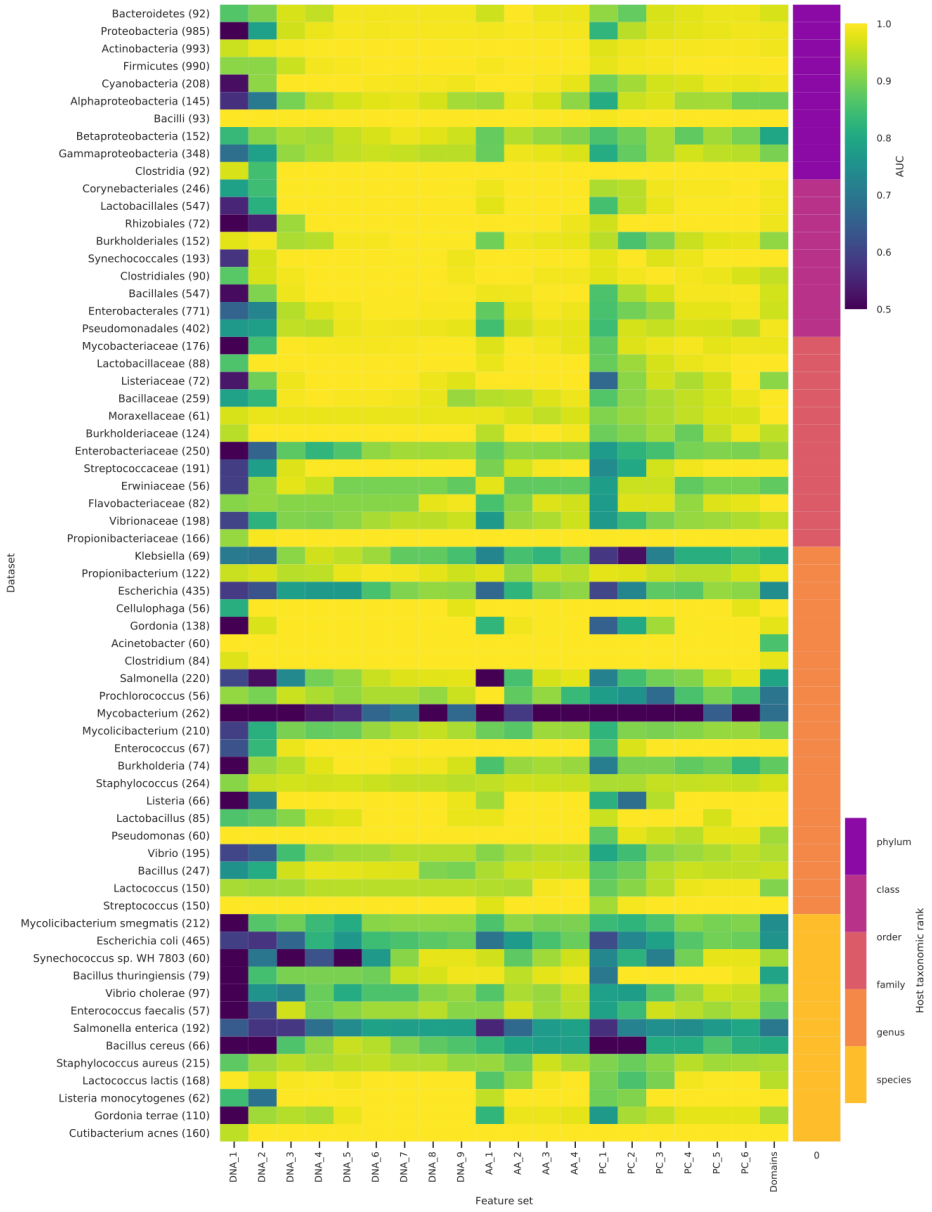


Figure 4.3: Comparison of the results for all the bacteria datasets for all the feature sets. The heatmap shows that all feature sets contain some predictive signal with an AUC $>$ 0.5 for the majority of the bacteria datasets. The rows each correspond to a dataset and are ordered by taxonomic rank (indicated by the colour bar on the right) and each column a feature set. The feature set labels the letters indicate the genome representation and the number the kmer size. DNA - nucleotide sequence; AA - amino acid sequence of CDS regions; PC - Physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property; Domains - presence of PFAM domain in the sequence. The colour indicates the AUC score for each classifier. All AUC scores of less than 0.5 were set 0.5, i.e., no predictive signal. The number of viruses in each dataset is in brackets.

indicated by the inner colour bar on the right). Each row corresponds to a dataset and are ordered by taxonomic rank (indicated by the outer colour bar on the right) and each column corresponds to a feature set. For the feature set labels the letters indicate the genome representation and the number the kmer size. DNA - nucleotide sequence; AA - amino acid sequence of CDS regions; PC - Physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property; Domains - presence of PFAM domain in the sequence. The colour indicates the AUC score for each classifier. All AUC scores of less than 0.5 were set 0.5, i.e., no predictive signal. The number of viruses in each dataset is in brackets.

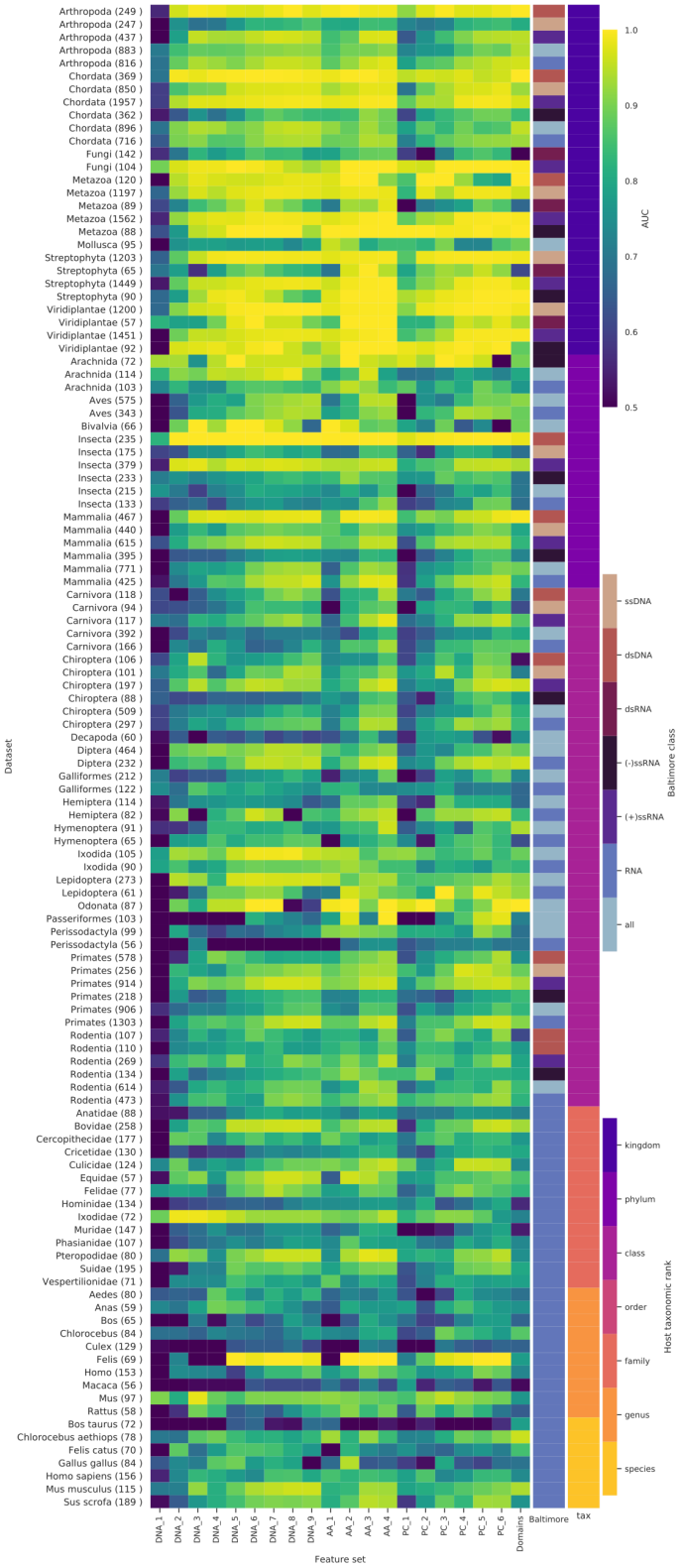


Figure 4.4: Comparison of the results for all the eukaryote datasets across all the feature sets and for all Baltimore groupings (indicated by the inner colour bar on the right).

4.3.3 Longer kmers of all genome representations are more predictive.

To compare the effect of kmer length on prediction accuracy we tested a range of kmer lengths for the sequence representations of the genomes (nucleic acid, amino acid and physio-chemical properties). The results show that for all feature levels, prediction improves with increasing kmer length (Figures 4.5 and 4.6). This is despite the exponential growth in the size of the feature sets, e.g., the DNA_2 feature sets have 4^2 =16 possible kmers, compared to DNA_9, which has 4^9 = 262144. These larger feature sets are very sparse. For example, the DNA_9 Mammalia dataset for RNA viruses has a sparsity of 0.91, i.e., over 90% of the elements in the sequence by kmer matrix are zero, although over the 425 virus genomes almost all of the possible kmers occur in at least one genome.

Prediction appears to be more difficult for the eukaryote datasets (Figure 4.6). This is perhaps due to the fact that eukaryote hosts are infected by viruses from across all seven Baltimore classes. The alternative replication/life-cycle strategies used by viruses from different classes will involve dissimilar sets of molecular interactions. It is therefore likely that they will acquire disparate host-derived signatures in their genomes, making the classification task more challenging. The problem is further exacerbated by the size of the datasets with few host taxa being available below the rank class when split on Baltimore class to meet our minimum dataset size. When testing the datasets formed by combining all RNA viruses, or all Baltimore classes, we were able to widen the range of hosts tested. Although prediction is better when using individual Baltimore classes, there is still a predictive signal when using the combined datasets (Figure 4.6).

Comparing classifiers for datasets moving from higher to lower taxonomic levels, i.e, from phylum to species level in the host tree, prediction becomes less accurate and less consistent across all the feature sets. For the bacteria datasets at phylum level all of the feature sets with the exception of DNA k=1, are highly predictive with an average AUC of 0.86 and standard deviation of 0.07 (Figure 4.5.i). In contrast, the species level classifiers have an average AUC of 0.67 and standard deviation of 0.15 (Figure 4.5.vi). One possible reason for this drop in predictive power (and increased variance) is the decrease in size of the datasets, as the data is stratified into a larger number of smaller subsets (Figure 4.2). This is confirmed by comparing the AUC scores against dataset size (Figure 4.7). Although many of the smaller datasets achieve a high AUC (towards top left of the plot), the worst performing classifiers all correspond to smaller datasets (bottom left of the plot). Finally,

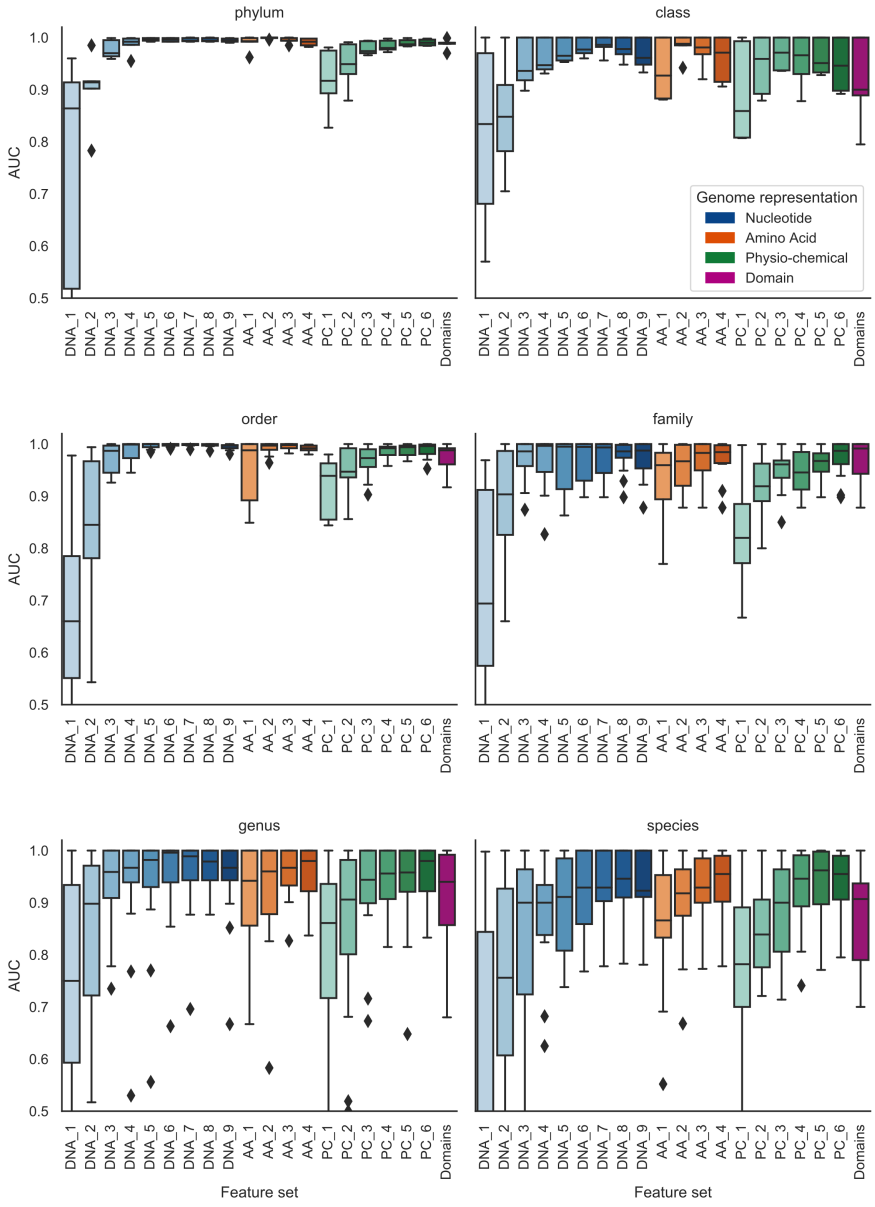


Figure 4.5: The effect of kmer length on prediction across host taxonomic ranks for the bacteria datasets. The boxplots show how prediction improves with increasing kmer length for all representations of the genome and that prediction gets more difficult at lower taxonomic ranks. Genome Representation is indicated by colour and kmer length by depth of colour:DNA - nucleotide sequence (blue); AA - amino acid sequence of CDS regions (orange); PC - Physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property (green); Domains - presence of PFAM domain in the sequence. Any AUC scores of less than 0.5 were reset to 0.5, i.e., no predictive signal.

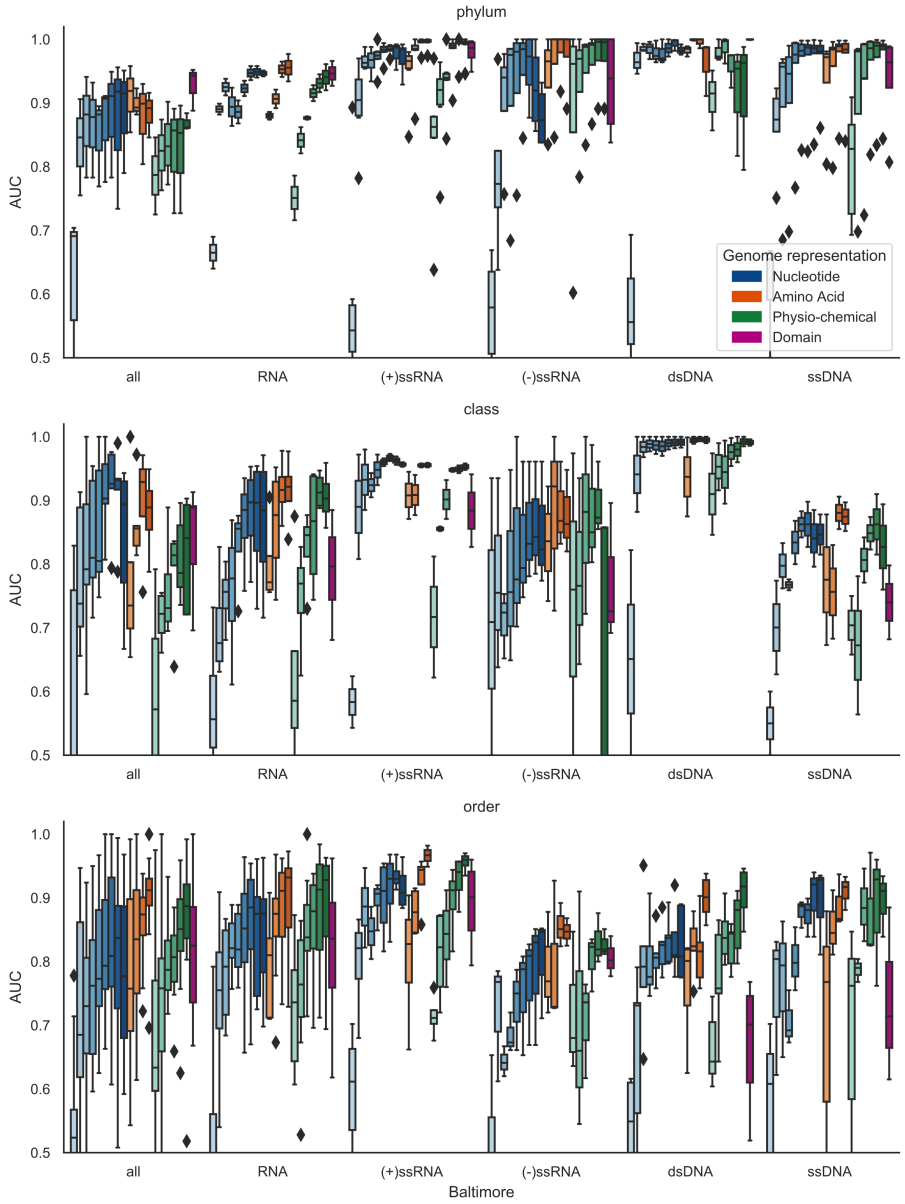


Figure 4.6: The effect of kmer length on prediction for the eukaryote datasets. As with Figure 4.5 we see prediction improves with increasing kmer length comparing prediction across the different Baltimore groupings. This boxplots show how prediction improves with increasing kmer length for all representations of the genome and that prediction gets more difficult at lower taxonomic ranks. Genome Representation is indicated by colour and kmer length by depth of colour: DNA - nucleotide sequence (blue); AA - amino acid sequence of CDS regions (orange); PC - physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property (green); Domains - presence of PFAM domain in the sequence. Any AUC scores of less than 0.5 were reset to 0.5, i.e., no predictive signal.

many of the worst performing small datasets were generated by including all RNA viruses (denoted by crosses in Figure 4.7). By their nature, these polyphyletic datasets will likely contain a wider range of host-derived mimicry signals than datasets comprising individual Baltimore classes, and it seems reasonable that they would therefore suffer more from the lack of a large number of training examples.

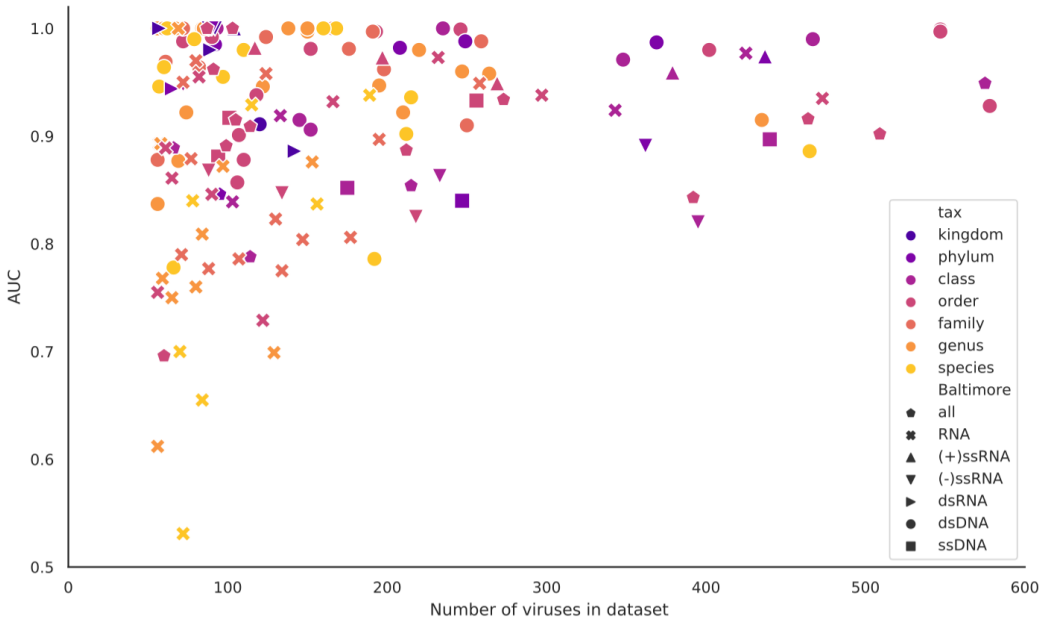


Figure 4.7: Comparison of the AUC scores against the size of the datasets. The scatterplot shows that most of the classifiers achieve good AUC scores (above 0.85). This is the case even for the small datasets and for those at family level and below. The points are coloured by the host taxon level and shaped by Baltimore group. All the classifiers are for AA_4 feature sets. All AUC scores of less than 0.5 were reset to 0.5 ie. no predictive signal.

4.3.4 The predictive signal contains both phylogenetic and convergent elements.

Next we performed experiments to determine if we were finding more than just a phylogenetic signal embedded in the virus genomes. Our goal was to separate the co-evolutionary signals due to the phylogenetic relationship between the infecting viruses (the virus-specific signal) from those embedded by convergence of viruses mimicking host functionality (the host-specific signal). We developed a novel cross validation method where, rather than stratifying data randomly into training and test sets we withheld one complete virus family from training and then used it to test the resulting classifier (Figure 4.8). Our aim was, as

far as possible, to holdout a group of closely related viruses. High predictive performance on the holdout family would imply a strong host-specific signal, i.e., predictive signals can be generalised across viruses for the same host. Poor predictive performance would indicate a signal that was specific to that particular virus family. High predictive performance on the holdout family would imply a strong host-specific signal, i.e., predictive signals can be generalised across viruses for the same host. Poor predictive performance would indicate a signal that was specific to that particular virus family.

We used the bacteria phylum to order taxa groups that are infected by the three major Caudovirales families. This choice was made to allow us to create datasets with enough viruses of both positive and negative classes, in equal numbers, in both the training and holdout/test datasets to train and test 'holdout' classifiers. Although the ICTV virus taxonomy for bacteriophage does not reflect phylogeny at family level there is currently no consensus on a method to phylogenetically classify viruses at levels deeper than genus. Using the multiple genera within the families as a group enabled us to select large enough datasets to reasonably train and test a 'holdout' classifier. While phylogenetic based classification systems such as Victor (Meier-Kolthoff and Göker 2017) and Gravity (Aiewsakun et al. 2018) do broadly support the ICTV assignments, finding distinct clades at genus level, they also demonstrate that there are inconsistencies in bacteriophage classification with some unrelated viruses in the genera.

Despite holding out a virus family, it could still be the case that there are viruses in the training set of very high similarity to those that had been held out. To remove these, we filtered training sets using average nucleotide identity (ANI) with a threshold of 75% identity across an alignment of at least 10% of the shortest genome. ANI has become established as a reliable and robust method for identifying phylogenetic relationships (Jain et al. 2018) and recently it has been proposed that pairs of viruses with ANI of greater than 95% over an alignment fraction of 85%, are part of the same species (Roux et al. 2019). Because alignment becomes unreliable at lower sequence similarity FastANI has a minimum cutoff of 75%. To ensure that we were removing more distantly related viruses we reduced the alignment fraction to 10%. Our experimental setup is depicted in Figure 4.8.

To create the holdout datasets we selected bacteria taxa that had multiple Caudovirales families infecting them in large enough numbers to form reasonable sized training and test sets. This requires that the holdout groups and remaining viruses must have a mixture of both positively and negatively labeled viruses (S3 Table 3). As we were expecting a

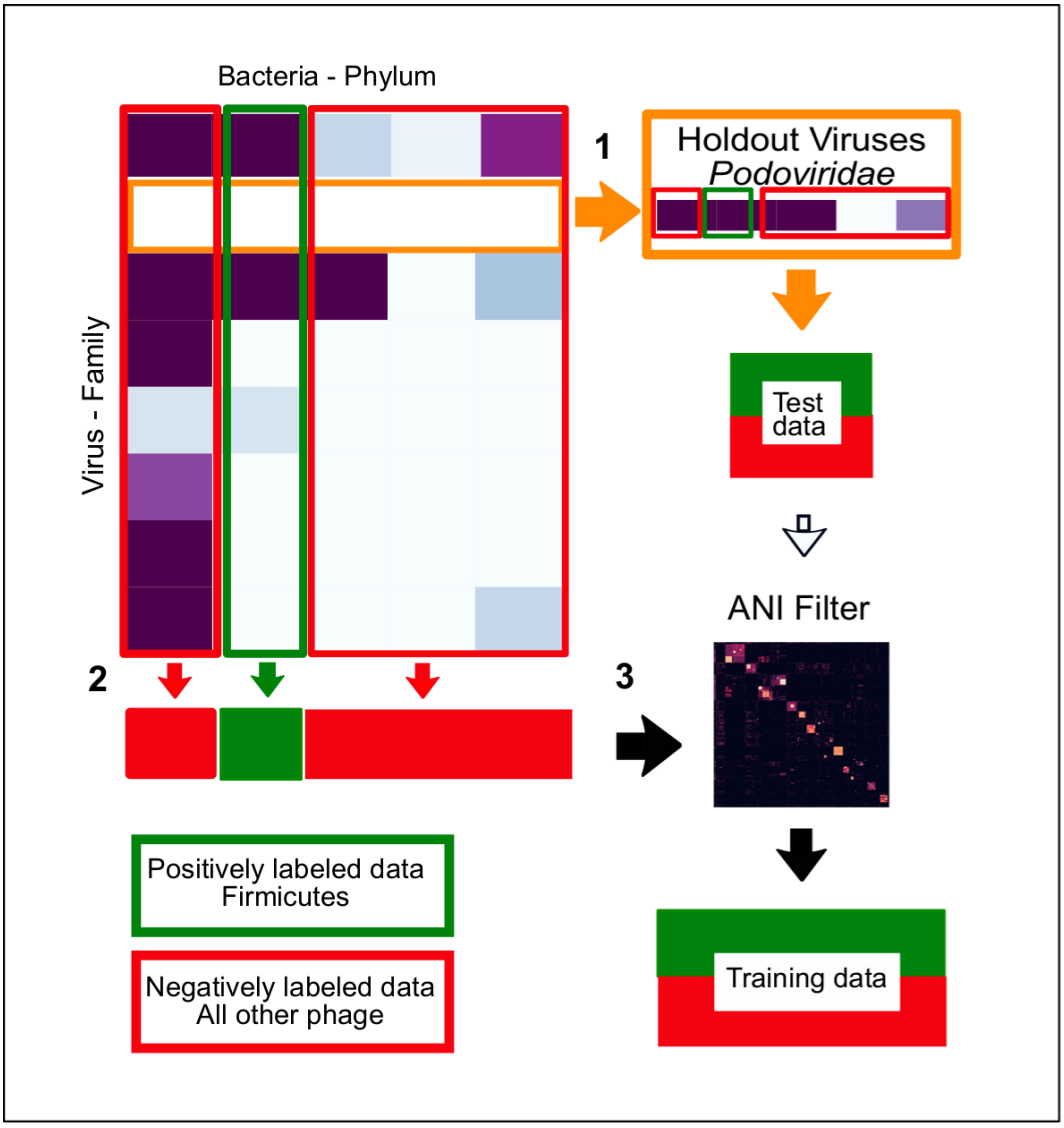


Figure 4.8: Creating the holdout datasets.
This shows an example of how a holdout dataset was created. Using the virus host interaction matrix for bacteria hosts at the phylum level and the viruses at family level, the holdout datasets were made by: (1) Removing a family of viruses, here Podoviridae, from the data. These holdout viruses are made up of infecting/non-infecting viruses and are then used as the test data. (2) The rest of the viruses that infect/don't infect the labelled host. Here the phylum Firmicutes are used to form the training set. And, (3) The training viruses were then filtered to remove any viruses that have greater than 75% ANI to any of the holdout/test viruses.

big loss of signal we chose the most predictive feature sets of each genome representation, along with di-nucleotides to serve as a baseline.

To assess the performance of these 'holdout classifiers', we compared them with our previous classifiers (referred to as 'all'), where a random split of all the viruses was used to form both the training and test sets (Figure 4.9). Interestingly, while we observed a small drop in AUC performance across all feature sets, we found that the majority of the 'holdout' classifiers retained a predictive signal. The mean ratio between AUC scores of the holdout classifier and standard classifier for the same dataset was 0.86. Across the different feature sets tested this mean ratio ranged from 0.77 (PC_6) to 0.94 (DNA_9) (Figure 4.10). This small drop in performance demonstrates that a predictive signal is still present, suggesting a common signal that is specific to viruses that infect the labelled host. This indicates there is convergence on a set of host-specific mimicry signals, such as molecular interactions, that is shared across all virus families that infect the host taxa (including the holdout 'family'). In a few cases there is a complete loss of signal, which we hypothesise to mean that the signal learned when training the classifiers on all the viruses includes an element that is specific to the holdout family, and this part of the signal will be absent when training the holdout classifier.

In terms of comparing the different genome representations, it is difficult to identify a consistent pattern as to which feature sets have the biggest signal loss, although protein domains (Domains-1), and physio-chemical property derived features (PC_5, PC_6), have more datasets where the holdout classifiers have a big signal loss (Figure 9) than the other representations. Whilst for the majority of Domains and PC datasets remain predictive, roughly a quarter of the datasets have a large drop in AUC, with ratios of less than 0.75. As a comparison, none of the DNA_6, DNA_9 and AA_3 datasets had a ratio of less than 0.75. Physio-chemical features are not changed by conservative amino acid substitutions. One possible explanation for the drop in performance of PC features is that as sequences diverge, they will remain more similar at the PC level than at nucleotide and AA levels. Likewise, protein domains remain more identifiabley homologous in divergent genomes, whereas convergence of domains is rare (Gough 2005). Removing the signal originating from the phylogenetic relationships between viruses in the holdout datasets may therefore lead to a larger drop in AUC for these more evolutionary-linked features. Cases where the domain signal is not lost may indicate a distant phylogenetic relationship or be due to shared domains arising as a consequence of horizontal gene transfer (HGT).

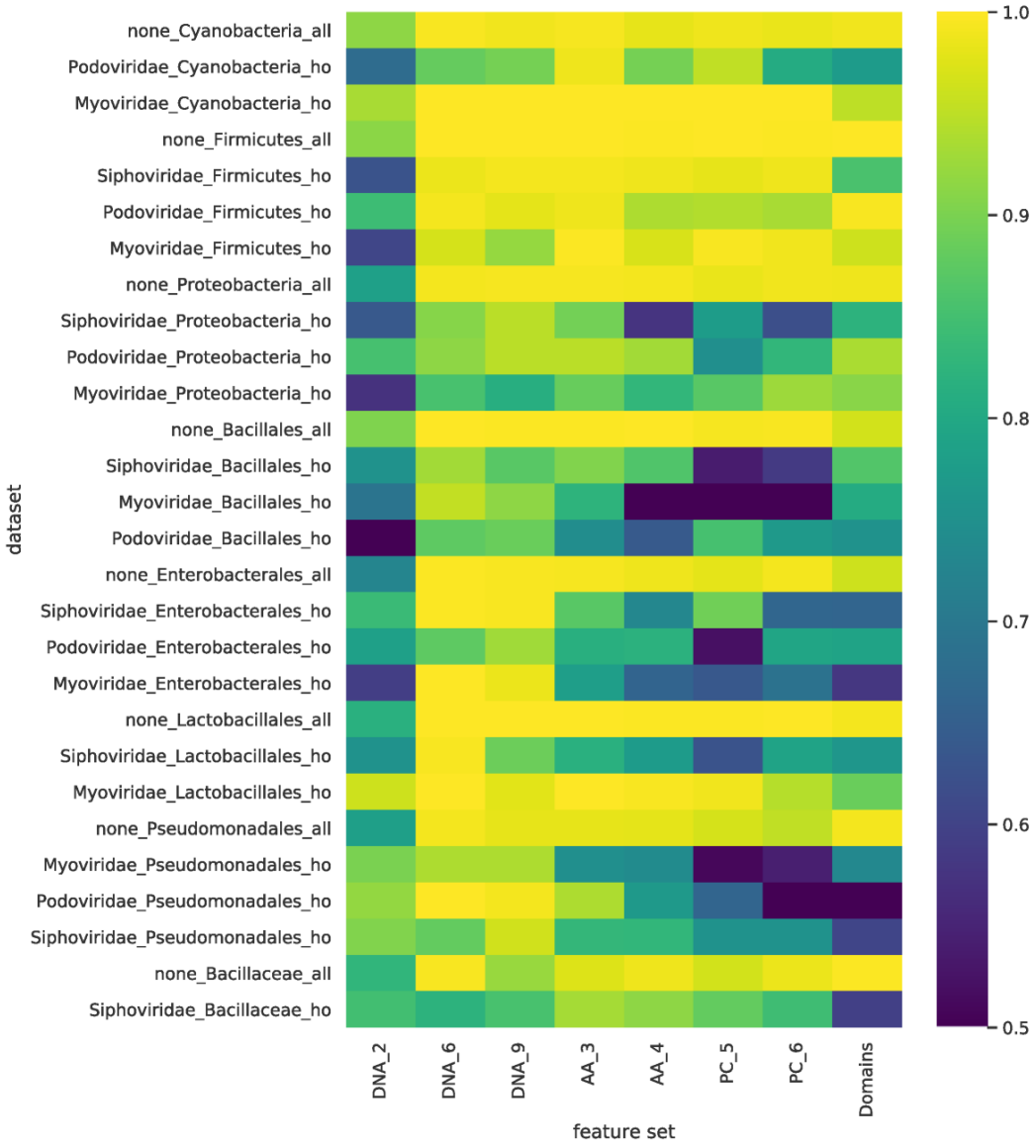


Figure 4.9: Comparison of the 'holdout' and 'all' classifiers showing the signal loss. Comparison of holdout and the standard (labelled 'all') classifiers for each dataset. For the majority of datasets there was a small loss in predictive power, implying that both classifiers are learning a shared signal. In a minority of cases there was a complete loss in predictive power implying the lack of a common signal. Each row corresponds to a dataset and each column a feature set. The feature set labels the letters indicate the genome representation and the number the kmer size. Genome representation:DNA - nucleotide sequence; AA - amino acid sequence of CDS regions; PC - physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property; Domains - presence of PFAM domain in the sequence. The colour indicates the AUC score for each classifier. All AUC scores of less than 0.5 were set 0.5, i.e., no predictive signal.

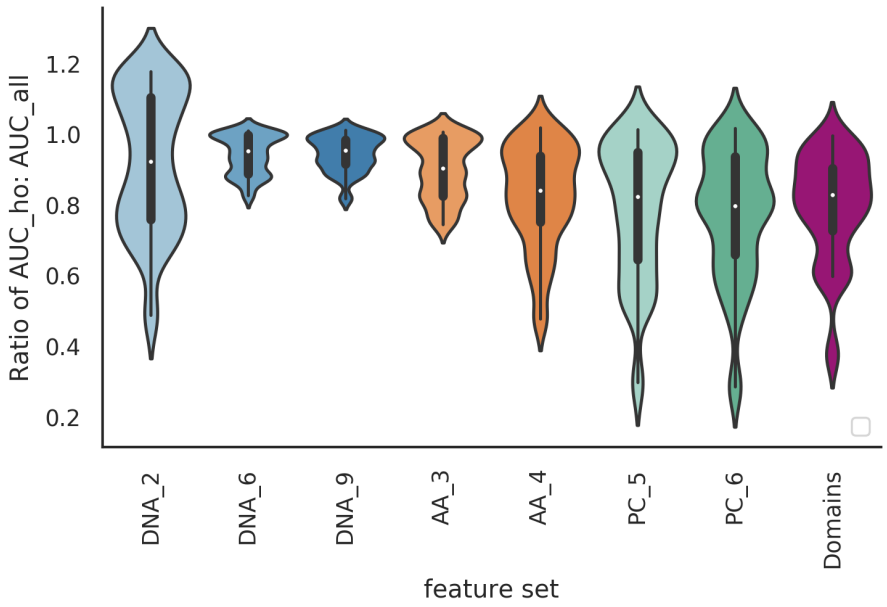


Figure 4.10: The signal loss for holdout classifiers. Violin plots of the ratios of the AUC scores for holdout (AUC-HO) to standard (AUC-all) classifiers for each dataset showing the variation in signal loss for the different feature sets. For the feature set labels, the letters indicate the genome representation and the number the kmer size. Genome Representation:DNA - nucleotide sequence (blue); AA - amino acid sequence of CDS regions (orange); PC - physio-chemical properties, each amino acid residue binned into one of seven bins based on its physio-chemical property (green); Domains - presence of PFAM domain in the sequence.

4.3.5 Feature sets from the different genome representations contain complementary information.

The overall aim of our study was to investigate whether using features extracted from different levels of viral genome representation were predictive of host. To check whether these alternative features are redundant or provide complementary information we combined feature sets from the different genome levels. A property of kernels, as used by SVMs, is the fact that it is straightforward to combine feature sets by creating composite kernels (Lanckriet et al. 2004). We thus combined the most predictive kernels from the DNA_9, AA_4, PC_5 and Domain feature sets, in different linear combinations. Weights for each kernel were varied between 0 and 1 (in steps of 0.05) with the sum of the weights across the four kernels constrained to equal to 1. To test if this improved prediction we selected a poorly predicted dataset as an example. We used the holdout classifier for the host label Bacillales and holdout group Siphoviridae. The results for the single kernel clas-

sifiers were DNA_9 = 0.91, AA_4 = 0.85, PC_6 = 0.59, Domains = 0.86. A summary showing the results for each classifier, grouped by the number of kernels contributing to the combined kernel, is shown in Figure 4.11 This demonstrates that overall, prediction improves as more kernels, drawn from the different genome representations, are included in the combined kernel. Furthermore, using different kernels could be used as a method to tune the classifier on the metric of importance. For example, Figure 4.12 which shows the same data as the false discovery rate (FDR) against true positive rate (TPR) for each classifier. Conversely, when different kmer lengths from the same genome representation were combined, no improvement in prediction was seen (results not shown).

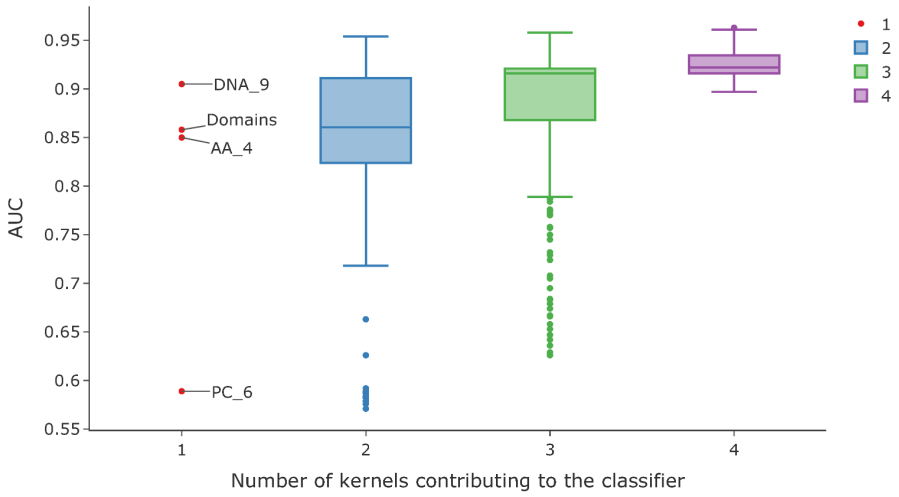


Figure 4.11: Combined kernel classifiers. This shows an example of how prediction can improve with the number of kernels contributing to the SVM classifier. This shows the results for all the iterations for combining kernels grouped by the number of kernels contributing to the combined kernel, for the dataset for the host order Bacillales with holdout group Siphoviridae. The red points are the results for the single kernels classifiers: DNA_9 - nucleotide sequence kmers length 9; AA_4 - amino acid kmers of length 4; PC_6 - physiochemical properties of amino acid sequence kmers length 6 ; Domains - presence of PFAM domain in the sequence.

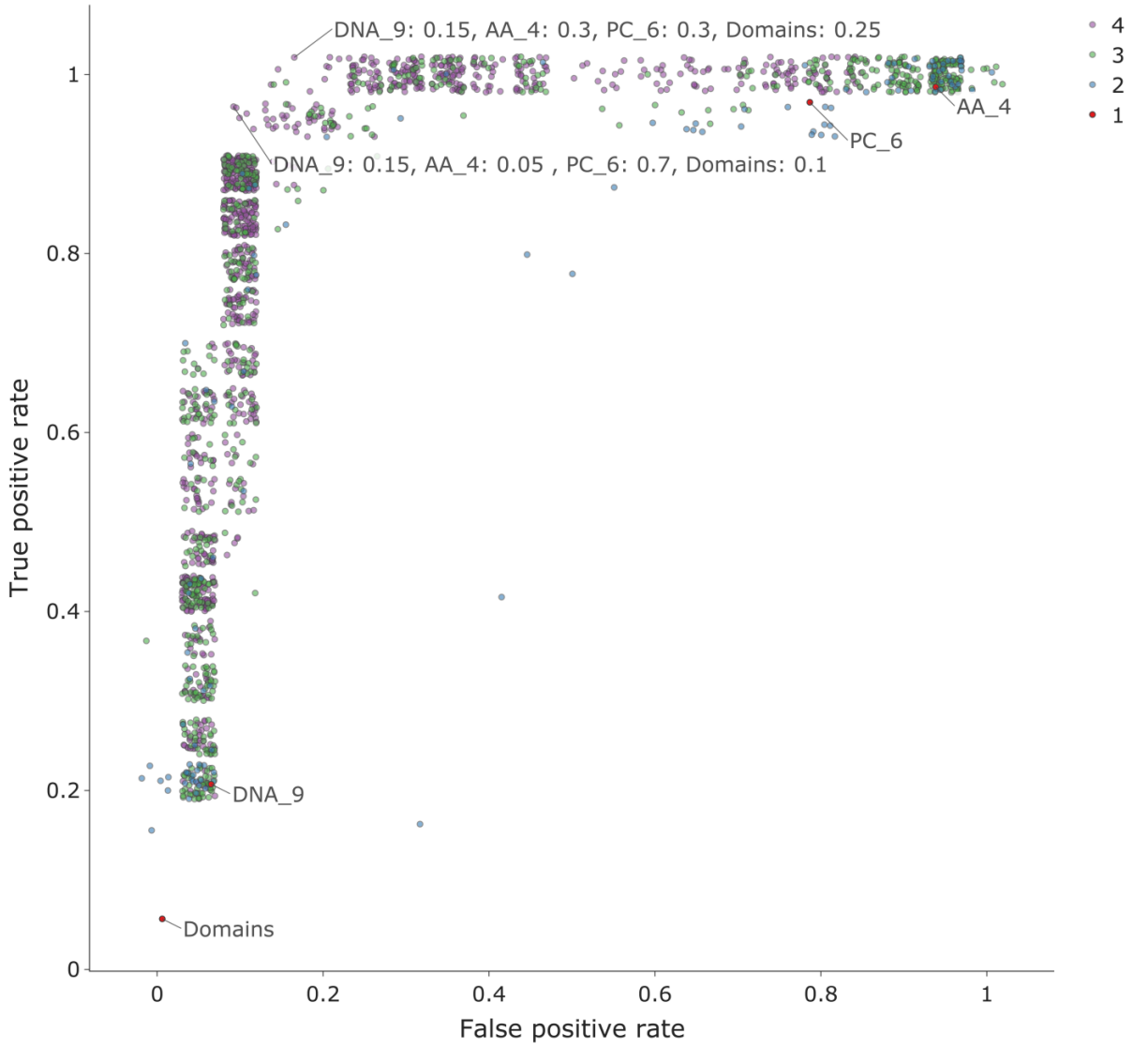


Figure 4.12: A plot of false positive rate (FPR) versus true positive rate (TPR) for the combined kernels of one dataset.By adjusting the contribution of the different kernels we can alter the specificity (1- FPR) and sensitivity (TPR) of the classifier. Each point represents the results for a classifier, each with a different combination of kernel weights, with the number of kernels shown by the point colour. The red (labelled) points are the results for the original single kernel classifiers. Additionally two of the best classifiers have been labelled with the kernel contributions. This shows the results for all the iterations for combining kernels for the dataset for the host order Bacillales with holdout group Siphoviridae. The data points have been 'jittered' to reduce the overlap. The kernels used were: DNA_9 - nucleotide sequence kmers length 9; AA_4 - amino acid kmers of length 4; PC_6 - physio-chemical properties of amino acid sequence kmers length 6 ; Domains - presence of PFAM domain in the sequence.

4.4 Discussion

The aim of this study was to compare the predictive power of a wide range of features for use in machine learning approaches to virus host prediction. We generated 20 feature sets from multiple representations of viral genomes and tested their capacity for host prediction. We found that features derived from all representations are predictive of host taxon for both bacteria and eukaryote hosts (Figures 4.3,4.4, 4.5 and 4.6), and that different features contain complementary signals that can be combined to improve prediction (Figures 4.11 and 4.12). Through a phylogenetically aware stratification scheme (Figure 4.8), our results strongly suggest that the features capture both phylogenetic and convergent signals (Figures 4.9 and 4.10).

The majority of previous machine learning approaches to virus-host prediction have focused on information from nucleotide sequences only (Tang et al. 2015; Li and Sun 2018; Kapoor et al. 2010; Gałan et al. 2019; Zhang et al. 2017a), which although predictive of host, ignore the rich information contained within alternative representations of the genomes. Through a process of convergent evolution, viruses are known to mimic their host's molecular interfaces at domain-domain and domain-motif interaction sites (Franzosa and Xia 2011; Daugherty and Malik 2012; Zheng et al. 2014). Such mimicry will be reflected in the amino acid sequence and domain content. Our results show that features derived from these genome representations can be successfully used for prediction, as demonstrated in previous studies (Raj et al. 2011; Leite et al. 2018).

Although we see evidence of a predictive signal in all of the representations across the host tree, there is no universal best feature set. In addition, some datasets are more challenging to predict with none of the feature sets achieving good performance. This is most apparent at the lower taxonomic ranks of species and genus where we are trying to separate the viruses of more similar hosts. Some Baltimore classes are easier to predict than others. For example, classifiers for both bacteria and eukaryotic dsDNA viruses consistently achieve higher AUC scores than RNA viruses, presumably because the prevalence of HGT means that similar sequences occur in viruses with a shared host. Conversely all of the eukaryote RNA datasets will be affected by their fast mutation rates leading to loss of sequence similarity.

Our novel holdout method suggests that the predictive signal embedded in viral genomes is made up of both phylogenetic and convergent signals. We removed, as far as possible, the signal coming from the phylogenetic relationships between the viruses infecting a host

and found that the majority of our 'holdout' classifiers still contained a predictive signal. Possibly the signal remaining is due to the convergence of both the training and 'holdout' viruses on common host molecular interfaces or other host factors. Hence, we hypothesise that the predictive signals in the viral genomes are a combination of two elements: the phylogenetic or virus-specific element which is removed from training causing the loss in prediction; and the convergent or host-specific element which remains still allowing for some prediction. By contrast, 'holdout' classifiers that have a complete loss in signal, indicate that there is no convergence and that the 'holdout' viruses are probably using different molecular interactions than the 'training' viruses.

Our results show that increasing the length of the kmers improves prediction with all sequence representations. Although many machine learning approaches have used di-nucleotide features (Tang et al. 2015; Li and Sun 2018; Kapoor et al. 2010; Babayan et al. 2018; Gałan et al. 2019), other computational approaches have shown that using longer kmers (length 6 and 8) is beneficial to prediction (Galiez et al. 2017; Ahlgren et al. 2017; Villarroel et al. 2016). Zhang et al. (2017a) found that with random forest classification, increasing nucleotide kmers up to a length of 8 improved prediction. Interestingly, even though we might expect long kmers to perform badly due to mismatches, we found that for the ranges of kmer lengths we tested, prediction increases with length. This is in accordance with HostPhinder (Villarroel et al. 2016) that successfully used co-occurring kmers of length 16 to predict hosts and the finding that, even after controlling for HGT, much longer nucleotide sequences co-occur in viruses and their host across all classes of viruses and host (Zheng et al. 2014).

Longer kmer features and domains - which only occur once or a few times in a genome - have the capacity to encode information about local virus-host molecular interactions, such as motif-domain or domain-domain interfaces. This is opposed to the shorter oligo-nucleotides which occur multiple times in a genome and therefore gives a global measure of biases over the whole genome. Changes in the occurrence of these local features caused by single mutations as a virus adapts to its host, will have a big impact on kmer composition, whereas global genome-wide biases will take many mutations over the whole genome to have a significant effect on the kmer composition. This will result in low dimensional biases being slower to match their host's bias, broadly agreeing with the findings of Di Giallonardo et al. (2017) that di-nucleotide composition of a virus is more closely related to its family taxa than its host species.

Machine learning requires suitable training examples. We are therefore constrained to

making predictions about the small fraction of cellular life that have many known viruses. This data is biased towards well studied organisms such as humans or pathogenic bacteria. To overcome this we have pooled hosts into higher taxa. For all feature sets, prediction gets more difficult for datasets at lower taxonomic ranks, with AUC scores being consistently higher for the phylum level datasets than those at species level. There are three possible causes for this deterioration. Firstly, the volume of data available for training decreases, although we see that all the worst scoring classifiers are for smaller dataset, many of the smallest datasets also achieve high scores. Secondly, the negative data are viruses that are not known to interact with the host and may include viruses for which interactions have not yet been observed, i.e., there may well be false negatives in our training/testing sets which can result in predictions incorrectly labeled as false positives. In addition, because viruses tend to infect closely related hosts this mislabeling will be more likely to occur at lower taxonomic ranks. Finally, as we move from higher to lower host taxonomic ranks, we are trying to discriminate between the viruses of more similar hosts – a more challenging problem. The lack of correlation between the size of the dataset and score is very apparent at genus and species level, (Fig 2), with the classifiers for the datasets S.enterica (192 viruses), E.coli (465 viruses) and the genus mycobacterium (262 viruses) performing particularly badly across all feature sets. This poor performance may be due to the fact that these groups of viruses are highly diverse and mosaic (Pope et al. 2015) or that they contain a high number of viruses that infect multiple hosts confounding any host specific signal. All these factors limit the specificity of our predictions and as virus host interactions tend to be species specific, or for bacteria, strain specific, this will limit the applications of this approach. While we restricted our study to using species reference sequences, a wider study using all available host labelled data from databases such as MVP database (Gao et al. 2018) or the NCBI Virus genome resource (Brister et al. 2015) should enable higher resolution prediction.

In this study, we have limited the sequence composition derived features (nucleic acid, amino acid and physio-chemical properties) to fixed kmers, not allowing mismatches. This is a rigid representation of the sequences where information from closely related kmers, such as those differing by a single mutation, is lost. Most functional elements in biology are better represented by motifs, where some positions in the short sequence are crucial to function and are conserved while others are more variable. Using a motif representation or relaxed kmers with mismatches, such as used by Raj et al. (2011) may be better at generalising across closely related sub-sequences and ultimately improve performance.

Future development and deployment of classifiers for different virus host prediction do-

mains would require task dependent optimisation of the models, and their operating thresholds. Various model optimisations are possible, including combining multiple feature sets. Our results show the potential of combining sets of features from different genome representations but that there is no consistent pattern as to which feature set works best for different classification tasks. Along with other model parameters, kernel weights would need to be optimized with respect to the most important error metric for the task in hand (Figure 4.12). For example, in environmental metagenomics minimising type 1 errors (the false discovery rate) is most important. Conversely, when trying to identify the reservoir source of a spillover virus, reducing type 2 errors is critical. This optimisation should include some measure of the prevalence of the data in order to take account of class imbalances. For specific tasks it may also be important to test the effects of partial or incomplete genomes on the performance of the classifiers to ascertain the usefulness and robustness of these features for use in metagenomics. Due to the extensive nature of this study, with over 3500 classifiers trained and tested, we did not perform any optimisation steps in our machine learning workflow. We would expect significant improvement in prediction as a result of using an optimisation process that involves both feature selection/combination and hyperparameter optimisation such as multi-kernel learning.

In conclusion, our results demonstrate that features derived from all four representations of viral genomes are predictive across the host tree. Combining a broader range of features that encapsulate the multiple layers of information held within viral genomes can lead to improved accuracy of virus-host prediction. This use of complementary features will lead to higher confidence assignments about host taxon information for the ever growing numbers of viruses with unassigned hosts from metagenomics studies and, for example, to identify the reservoir source of a spillover event. Furthermore, the local nature of domain and longer kmer features have the potential to be informative of the mechanisms leading to virus host specificity.

Chapter 5

Interpreting host-specific signals on viral sequences

5.1 Introduction

In the previous chapter, we discovered that different representations of the viral genomes were predictive of host and that the longer kmer feature sets were more predictive. These longer kmers are likely to occur once or rarely within a sequence making them locatable. In this chapter, we investigate whether we can interpret these local features learnt through supervised classification of host to identify regions of a viral sequence that are associated with host specificity.

5.1.1 Motivation

The emergence of SARS-CoV-2 in early 2020 and the ongoing COVID-19 pandemic has been a wake-up call to the potential of zoonotic viruses to cause global health and economic crises. SARS-CoV-2 is a member of the genus *Betacoronavirus*, part of the of *Coronaviredea* family that infect mammals and birds and circulate widely in wildlife and farm animals. SARS-CoV-2 is one of seven Human coronaviruses (HCoVs), all of which have arisen through zoonosis. Four of these viruses, OC45, HKU1, 229E and NL63, are now endemic, causing only mild respiratory infections. In contrast, SARS, MERS and SARS-CoV-2 that emerged in the last two decades and all cause severe disease with high

fatality rates of 10%, 37% and 5% respectively (Guarner 2020). Efforts at surveillance of potential zoonotic viruses is hampered by our lack of understanding about what factors enables a virus to switch host and to achieve onward transmission within the new host species.

Coronaviruses are enveloped positive-sense single-stranded RNA viruses with a genome of around 30kb encoding in the region of 29 proteins. The Spike protein protrudes from the membrane surface of the virion and is responsible for binding the host receptor and mediating viral entry. As well as this critical role in the viral life cycle, Spike is the most exposed and immunogenic viral protein making it an ideal target for drug and vaccine design.

Since the start of the pandemic there has been a massive global research effort into on every aspect of COVID-19 and SARS-CoV-2. This has led to the rapid sharing of information gained from experimental and computational analysis on public domains including twitter and the preprint servers. As of June 2020 there were over eleven thousand SARS-CoV-2 related articles being deposited on the BioRxiv and MedRxiv. There is a need for computation methods based on sequences alone that can quickly identify which features in viral sequences are most important to the virus-host relationships. These methods could quickly identify regions of interest to focus experimental researchers including antiviral and vaccines development.

5.1.2 Interpreting machine learning models

The identification of the discriminant features that are responsible for the phenotype being studied is a fundamental task for biologists to further our understanding of the underlying biological mechanisms at play. Modern sequencing techniques have resulted in massive amounts of genomic data being held in public databases and has led to advances in our understanding of mechanisms in evolutionary and functional biology. Supervised machine learning has become a standard tool for analysing these increasingly large volumes of sequence data. To fully leverage the potential of this data we need methods that are not only capable of prediction but that are also interpretatable. Prediction is the task of learning the patterns in the data that discriminate between phenotype but often these patterns remain hidden, in other words the model is treated as a 'black box'. Ideally, we would like to interpret the models by extracting these learnt patterns to discover which features underlie the biological mechanism behind the phenotype to gain novel biological

insights.

When selecting a supervised machine learning algorithm for a task we need to consider the trade-off between model performance and its interpretability. This is most evident in deep learning where the power of state-of-the-art methods is gained by increasingly complex and flexible models that render them opaque to human understanding. This has driven the recent growth in the new field of interpretable ML and the development of many "post hoc" methods to explain the increasingly complex models. In contrast, less complex models are much easier to interpret, for example, the model weights of a linear SVM model are easily accessible and the relative weights of a feature are an indication of their importance.

In the previous chapter, we made a comprehensive comparison of multiple feature sets derived from different viral genome representations and found that longer kmer features were highly predictive of host taxonomic information. Features comprising of longer kmers are in a sense "local", that is they occur only once or rarely in a viral sequence. This means that we can locate the specific position of a kmer on a sequence, as opposed to kmers that occur globally throughout a sequence, for example di-nucleotide kmers.

Our aim in this chapter was to explain a host predictive model as regions of a SARS-CoV-2 Spike sequence that are associated with its relationship its host. We used "local" features to train classifiers to discriminate between human and non-human hosts. The learnt model weights were extracted and used to transform a protein sequence into a "host signal". Interpreting host predictive classifiers in this way has the potential to identify sites important to virus-host interactions and to provide novel insights as to what changes have enabled host switching.

5.2 Methods

5.2.1 Data

All betacoronavirus genomes annotated with host taxonomic information were downloaded from NCBI viral zone on 16th of February 2020. Any genomes with ambiguous or no host information were discarded. In order to have large enough data sets for machine learning we grouped the viruses by host at the taxonomic rank of order. This gave us four groups

of viruses. We made whole genome datasets comprising of all the ORFs and Spike datasets using only the Spike ORF. These were used test and compare host prediction.

To remove near identical virus genomes from the data we used amino acid sequence identity to filter out genomes that had more than 99.5% similarity in the spike sequence. An all against all matrix was generated of the pairwise alignment of all the genomes using Biopython's global alignment tool.

5.2.2 Features

We used four representations of the genomes were used: the nucleotide (NA) ; amino acid sequences (AA); and physio-chemical (PC) and functional (FN) properties the binned amino acid sequence. The FN and PC features are alternative ways to group amino acids based on their physio-chemical properties: PC - (AGV, DE, FILP, HNQW, KR, MSTY and C) (Shen et al. 2007); FN is an alternative binning of amino acid residues (AMV, DE, ILNQST, GP, HKR, FWY,C). Kmers were extracted from each of the sequences in a dataset by counting the number of occurrences of each unique kmer of length k in the sequence. Different length kmers were used each to generate a separate feature matrix as described in Chapter 4.2. In this analysis the counts were unnormalised because all the sequences are approximately the same length.

5.2.3 Host classification

We used linear SVM to train a binary classifier for each of the host labels. SVM was designed for binary classification and performs well even when the number of features is much larger than the number of data points as in our datasets. Linear SVM models have the advantage that we can easily access the feature coefficients and results in a non-sparse model in which all the features are assigned a coefficient. We followed the methods described previously in Section 4.2, where we showed that linear SVM performed well our high-dimensional sparse feature matrices.

As done previously, we tested the multiple feature sets on binary datasets created by using the viruses known to infect the host order as the positively labelled data and all other viruses as the negative data. A Scikit-learn pipeline was implemented with five-fold cross validation including scaling the features with "standard scalar". The default value for

the penalty parameter C of 1.0 was used. In order to compensate for the imbalanced data sets the "class-weight" parameter was set to "balanced weights". To assess how the performance of these classifiers various evaluation metrics were collected including Area under ROC curve (AUC) scores, the confusion matrix (TP,TN,FP,FN) and the probability score of each genome for each classifier, obtained via Platt scaling. Having established that the feature sets were predictive, a classifier was trained using all the data in each of the datasets using each of the feature sets.

5.2.4 Transforming model parameters to position specific signals

Linear SVMs learn the coefficients that define the decision boundary between the positive and negative classes. To transform the sequences (AA, PC or FN) into sequences of weights, we extracted the coefficient for each kmer and mapped this onto the spike sequence giving each position a weight. The absolute size of the weight gives an indication of that positions importance and the sign indicates if it is positive or negative association with the host.

5.2.5 Selecting the non-phylogenetic kmers

To separate the phylogenetic from the non-phylogenetic signal in the feature sets as discussed in Chapter 4 we adapted a software package, Scoary(Brynildsrud et al. 2016). This was developed to score genes in microbial pan-genomes for their association with a phenotypic trait while accounting for the evolutionary relationship between genomes. It works by passing the candidate genes though a series of filters to remove genes that are spuriously associated to a trait via inheritance. The first filter uses the Fisher exact test to remove genes with low correlation with a trait. Next, is the phylogenetic aware step. The correlated genotype variants are collapsed into a single node and then Scoary counts the minimum number of times of a given gene-trait combination have independently co-emerged in the tree by looking for pairs of genomes that contrast in both genotype and trait. Finally, random permutations of the trait data are used to produce a null model which is used to calculate a test statistic for the last filter. The output for each trait is a single list of significant genes sorted by p value.

This software requires three files as input: 1. A traits file, that gives the traits for a set of genomes, we use the host label as a trait; 2. A gene-presence-absence file which indicates

the presence/absence for each gene in each genome, we use kmer in place of gene; 3. A phylogenetic tree of all the genomes. We built a Newark for all the viruses in the data-set using MAFFT (Katoh et al. 2002) to generate an alignment and FastTree 2 (Price et al. 2010) to build a tree. The output from Scoary gives a set of the highest scoring kmers on their likelihood of being part of the non-phylogenetic set. Each kmer is scored with Bonferroni and Benjamini-H p-value and only includes kmers with a naive p-value of less than 0.05.

5.3 Results

5.3.1 Data

To create data sets that contain sufficient examples of viruses of different hosts we defined the host label at the taxonomic rank of order. We made binary data sets by selecting viruses that infect the host taxa as the positively labeled data set and all other viruses as the negative data. The virus genomes are very unevenly distributed across the different betacoronavirus lineages, with over half of the genomes belonging to the MERS lineage. To reduce this bias we removed genomes that had more than 99.5% amino acid identity in the Spike ORF. The number of viruses in the final dataset are shown in table 5.1.

Table 5.1: Number of viruses in each data set: by a) per host b) per lineage

(a)

Host order	number of viruses
Primates	140
Chiroptera	72
Artiodactyla	71
Rodentia	20

(b)

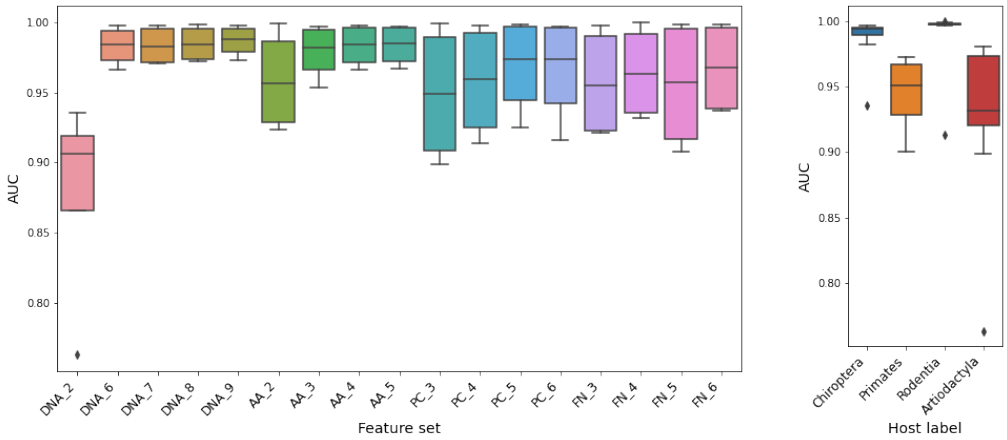
Lineage	Human	Other
HKU1	15	3
OC43	72	0
MERS	47	38
SARS	2	37
SARS-CoV-2	1	0

5.3.2 Exploring appropriate feature sets.

To identify the regions of the viral sequences that are potentially associated with host specificity we need to use features that have a number of properties. They must be predictive of host, locatable, and capture the functional elements of the sequence. We compared how a range of features sets match these requirements.

5.3.2.1 Predictive features

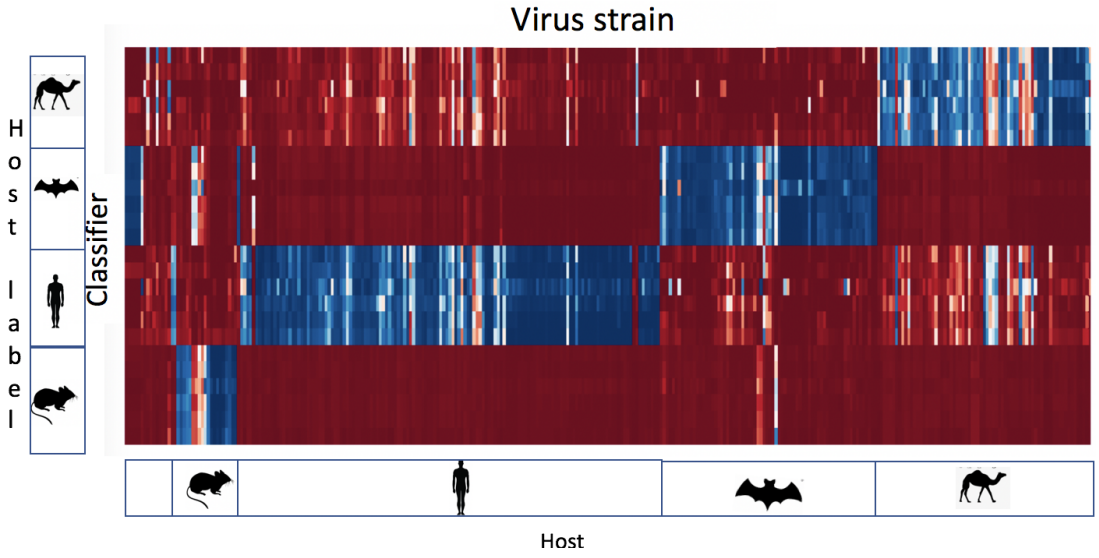
First, we needed to establish if the feature sets are predictive of host. We trained multiple SVM classifiers each testing a different combination of feature set and host labels. SVM is well suited to sparse data where the number of features is much greater than the number of samples. A linear kernel was used so that we could easily access the model weights. We compared the using whole genome and spike only sequences. We found that all the tested feature sets were predictive of host labels, all achieving an AUC score of greater than 0.90 Figure 5.1. The spike only datasets resulted in about 0.05 drop in the AUC score. Spike has a central role in determining host specificity because it is responsible for attachment to the host receptor (ACE2). For the remainder of this analysis we use the only Spike protein sequence.



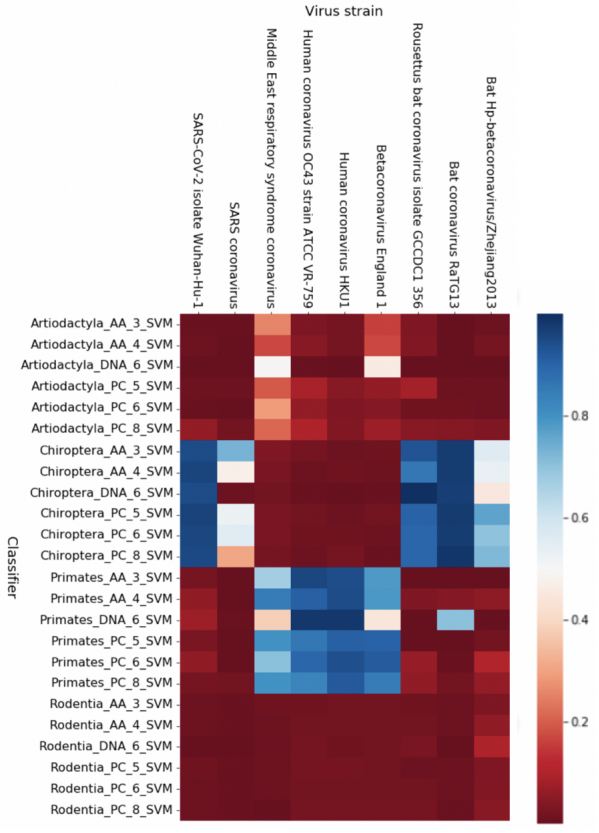
(a) Grouped by feature set (b) By host

Figure 5.1: Comparing the AUC scores for all the classifiers

Although we achieve strong classification there were a consistent number of false positive and false negative results in all the different classifiers. To investigate which viruses are being incorrectly classified we extracted the probability scores from the classifiers, that is the probability of a virus being assigned to the positive class. We found that none of the feature sets were able to correctly predict Primates as the host for SARS-CoV-2 or SARS viruses Figure 5.2b. Interestingly, SARS-CoV-2 in particular is being strongly predicted as a bat-like virus.



(a)



(b)

Figure 5.2: The probability scores for each of the viral genomes for each classifier. a) For all viruses and b) for a subset of viruses of interest. Each element represents the predicted probability of a virus belonging to the host label. Each row for a host-feature set classifier and a column for each virus (grouped by host in fig b, the unlabelled group are hosts from other orders. The colour indicates the probability of the virus in infecting the labeled host, blue is positive and red is negative.

5.3.2.2 Local features

To identify the regions on the viral sequences that are associated with we need the features to be "locatable". These kmers should preferably only occur once or rarely within a sequence. This will increase the likelihood that we can uniquely place any predictive signal associated with a particular kmer. Table 5.2 shows the number of the kmers occurring more than once within any of the spike sequences.

The expected probability of a kmer occurring more than once within a sequence is computed as 1- the probability of a kmer being unique:

The number of possible unique kmers in a feature set with kmers of length k from a fixed alphabet is:

$$K = |alphabet|^k$$

The number of kmers in a sequence of length L is:

$$N = L - k + 1$$

Then the probability of a kmer being non-unique:

$$P = 1 - ((K-1)/K)^{N-1}$$

Genome representation	Kmer length	Number of unique kmers in feature set	Number of unique kmers in data set	Proportion unique kmers	Expectation that kmer is unique	kmers in <5% of genomes
NA	2	16	16	0.0	0.0	0.0
NA	6	4096	4096	0.03	0.39	0.0
NA	7	16384	16384	0.2	0.79	0.11
NA	8	65536	65424	0.53	0.94	6.99
NA	9	262144	249498	0.81	0.99	36.14
AA	2	400	400	0.2	0.04	1.0
AA	3	8000	6442	0.86	0.85	43.64
AA	4	160000	26933	0.99	0.99	81.03
AA	5	3200000	37806	1.0	1.0	86.52
PC	3	343	341	0.24	0.02	1.17
PC	4	2401	2107	0.6	0.59	23.59
PC	5	16807	8840	0.89	0.93	55.19
PC	6	117649	21695	0.98	0.99	76.56
FN	3	343	342	0.3	0.02	3.8
FN	4	2401	1982	0.63	0.59	28.76
FN	5	16807	7330	0.83	0.93	57.75
FN	6	117649	16357	0.93	0.99	73.31

Table 5.2: Number of unique features in each feature set

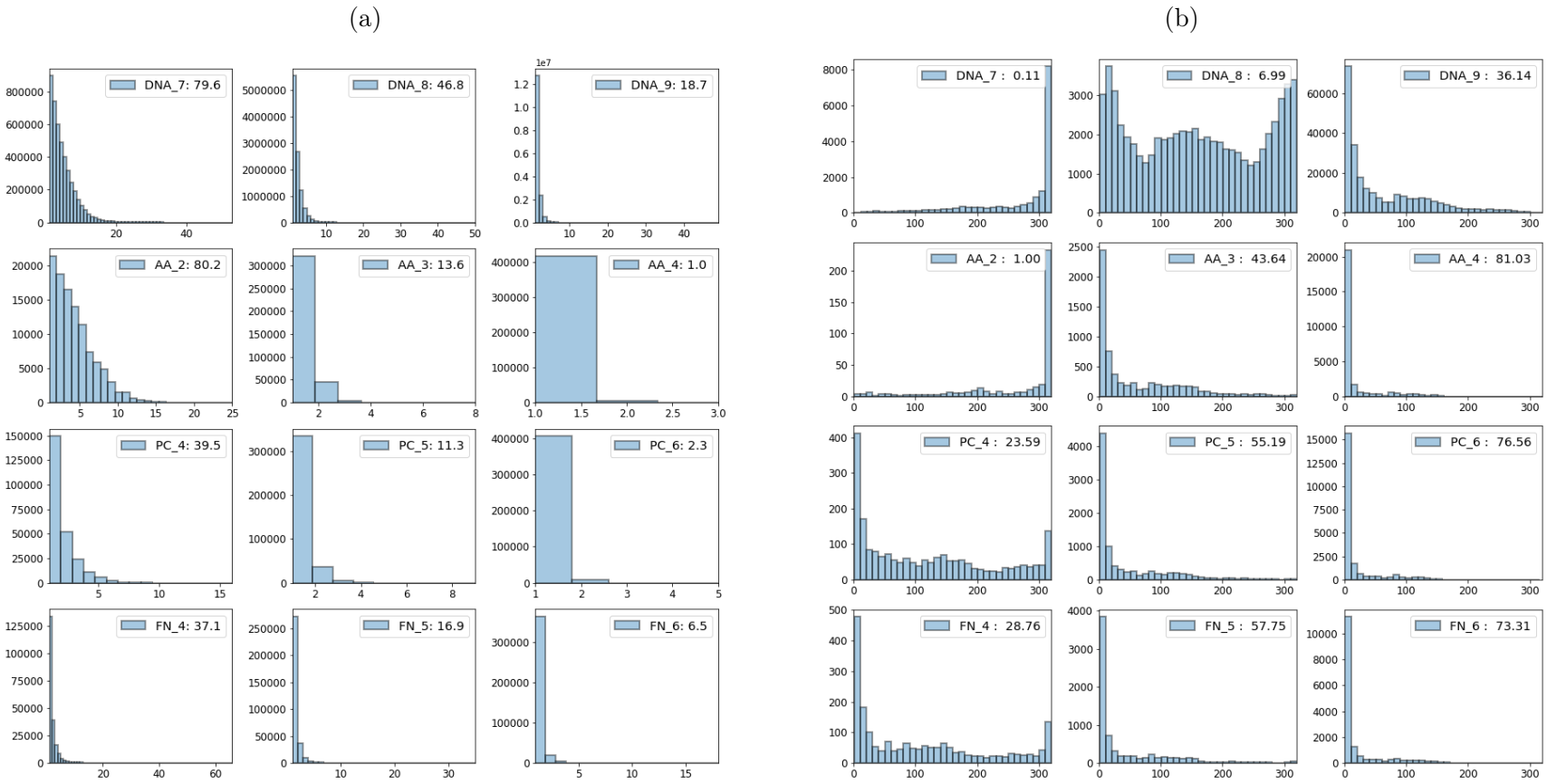


Figure 5.3: Comparing the frequency of occurrence of the kmers in the different feature sets (excluding zero's). Figure5.3a indicates the proportion of multiply occurring kmers for each feature set. Figure 5.3b shows the frequency that kmers occur in multiple genomes, the proportion of kmers that occur in less than 5% of the sequences is indicated for each feature set

There is a trade-off when selecting the length of the kmers: too short and they will not be unique, too long they will not be robust to mutations. One effect of using a longer kmer sizes is to make the feature matrices more sparse, features that only occur in very few sequences are unlikely to contribute to prediction. The last column in Table 5.2 and Figure5.3b show the proportion of the kmers that occur in less than 5% of the spike sequences in the data se (15/360 viruses). This trade-off is shown in the relationship between uniqueness an sparseness of the different datasets in Figure 5.4.

5.3.2.3 Functional features

Ideally, the features should capture the functional elements of the sequences, in other-words they should be robust to mutations that don't effect function. Our strategy is to use the different representations of the genome as described in the previous chapter, including two alternative ways of binning the amino acids, the physio-chemical (PC) and functional (FN) representations. Amino-acid kmers already negates the issue of synonymous mutations, binning the amino acid by their properties negates the issue of functionally conservative substitutions.

5.3.3 Transforming model weights into position specific signals on viral sequences.

Next, to interpret the predictive models we transformed the parameters that define the decision boundary learnt by the SVM algorithm into site specific signals on the spike protein of SARS-CoV-2. The model coefficients associated with each kmer were extracted into a dictionary and normalised by dividing them by the maximum absolute coefficient in the model. Each position i in a sequence is then assigned the weight of the kmer at positions $[i : i + k]$ to obtain a position specific predictive signal. Figure 5.5 shows these signals for the different feature sets on the RBD of the spike protein of SARS-CoV-2. The sign of the weight indicates if it is associated with the positive or negative class, this is because the feature matrix consists of values > 0. The magnitude of the weight of a feature is an indication in its relative importance in driving prediction.

Given that our models have been trained to discriminate host, regions of the protein with high absolute values are likely to be important in determining virus host specificity. This is not a direct relationship, especially if there are groups of highly correlated features, which

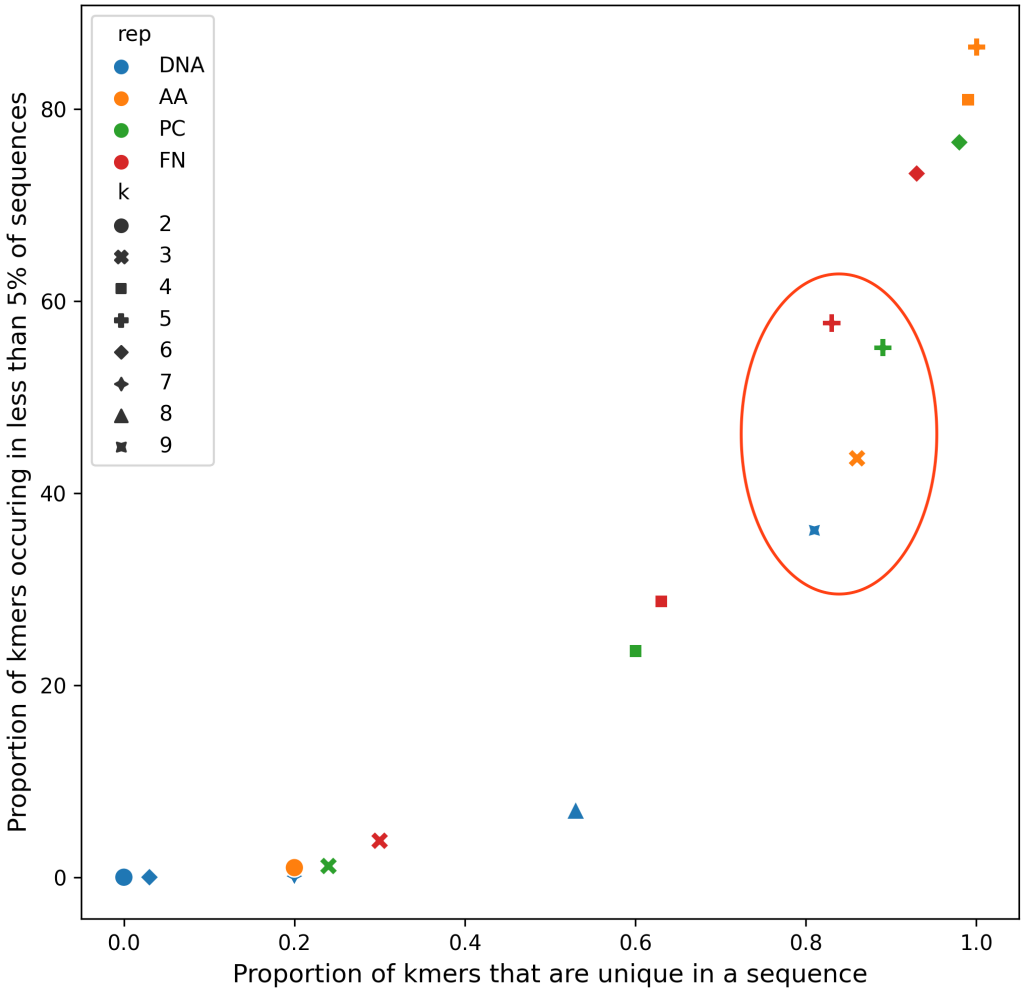


Figure 5.4: Trade-off between "uniqueness" and sparsity of the feature sets. Feature sets in the bottom right of the plot are the least sparse and are also more likely to be robust to mutation but the majority of kmers are non-unique. Feature sets in the top right of the plot consist of mainly unique kmers but are very sparse. Those circled in the middle have a high proportion of unique kmers ($> 80\%$) and occur enough genomes to contribute to prediction.

has the effect of reducing the weights of features in these groups (Toloşi and Lengauer 2011). Nevertheless, the higher weighted regions are still the most important in driving prediction.

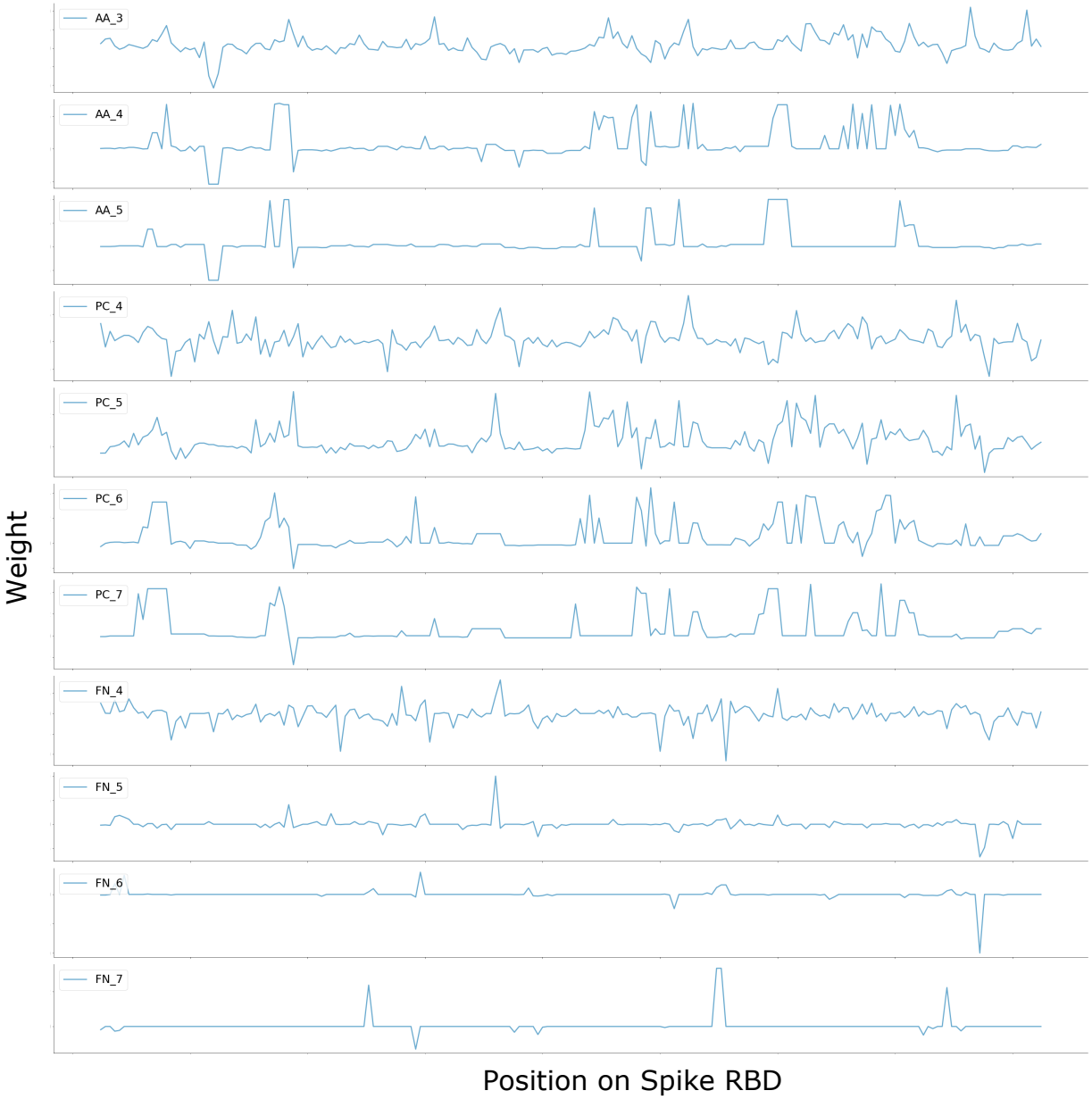


Figure 5.5: Signals from the different feature sets shown on the receptor binding domain of Spike protein on SARS-CoV-2.

5.3.3.1 Stability/reproducibility

Stability is another factor that should be considered when selecting both the algorithm used and the most appropriate feature set for reproducibility. This is the stability of the model parameters to perturbations in the data. If the regions being identified are easily affected by changes in the data the implication is that theses signals are more likely to be an artefact of the data rather than the underlying biology that we are interested in, (Jiang et al. 2020; Nogueira et al. 2018). By extracting the weights for classifiers produced during cross validation, we can compare the weights for 5 different random samplings of the data and visually assess the stability of the feature sets by comparing the alignment of the peaks across all 5 samplings. Figure 5.6. We also tested alternative classification algorithms including SVM, random forest, logistic regression and elastic net. Random forest was less stable than SVM, whereas regularised logistic regression and elastic net were very stable but only produced very sparse peaks.

5.3.4 Are the predictive signals informative?

5.3.4.1 The signals are more informative than a Null model.

To establish whether or not the signals from our host predictive classifiers are likely to contain biologically meaningful information we first assessed whether they are different from the signals you would see from a random non-predictive classifier. To do this, we generated "real" and "random" classifiers for each of the feature sets. The random classifiers were trained using the same feature matrices used to generate the real classifier but the labels were permuted by randomly rearranging their order before training. The coefficients were then extracted and transformed to weights as for the real classifiers. This resulted in one real and 100 random classifiers and associated signal for each feature set.

A comparison between the mean weight at each position for all the real signals (blue) and the mean weight for the random signals (orange) i s shown in Figure 5.7. This demonstrates that there is some coherence between the real signals with many of the peaks from individual classifiers coinciding resulting in a strong peak. This is in contrast to the random signal which is consistently much weaker with no strongly defined peaks.

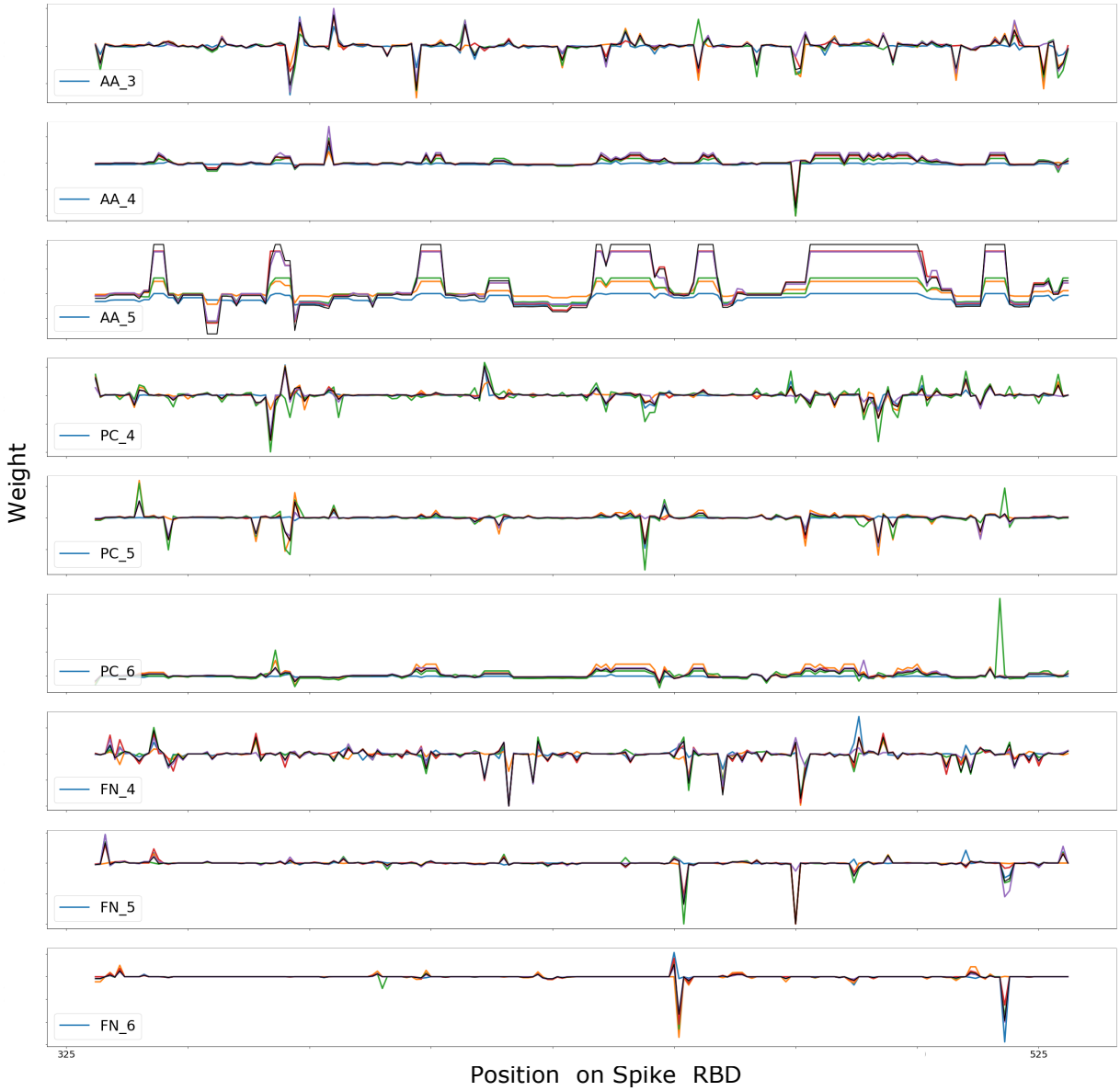


Figure 5.6: Stability: Comparing the signals for the different subsets of the data for each feature set.

5.3.4.2 The signal contains functionally relevant elements.

Next, to assess if the signals are likely to contain functionally relevant information we compared our predictive signals with the results of a deep mutation scan (DMS) of the receptor binding domain (RBD) of the Spike protein of SARS-CoV2 (Starr et al. 2020). This experiment mutated each position in the RBD to every other possible amino acid and measured the effect on binding affinity to the ACE2 receptor and on expression levels, as a proxy for structural stability. To compare our signals with this data we calculated

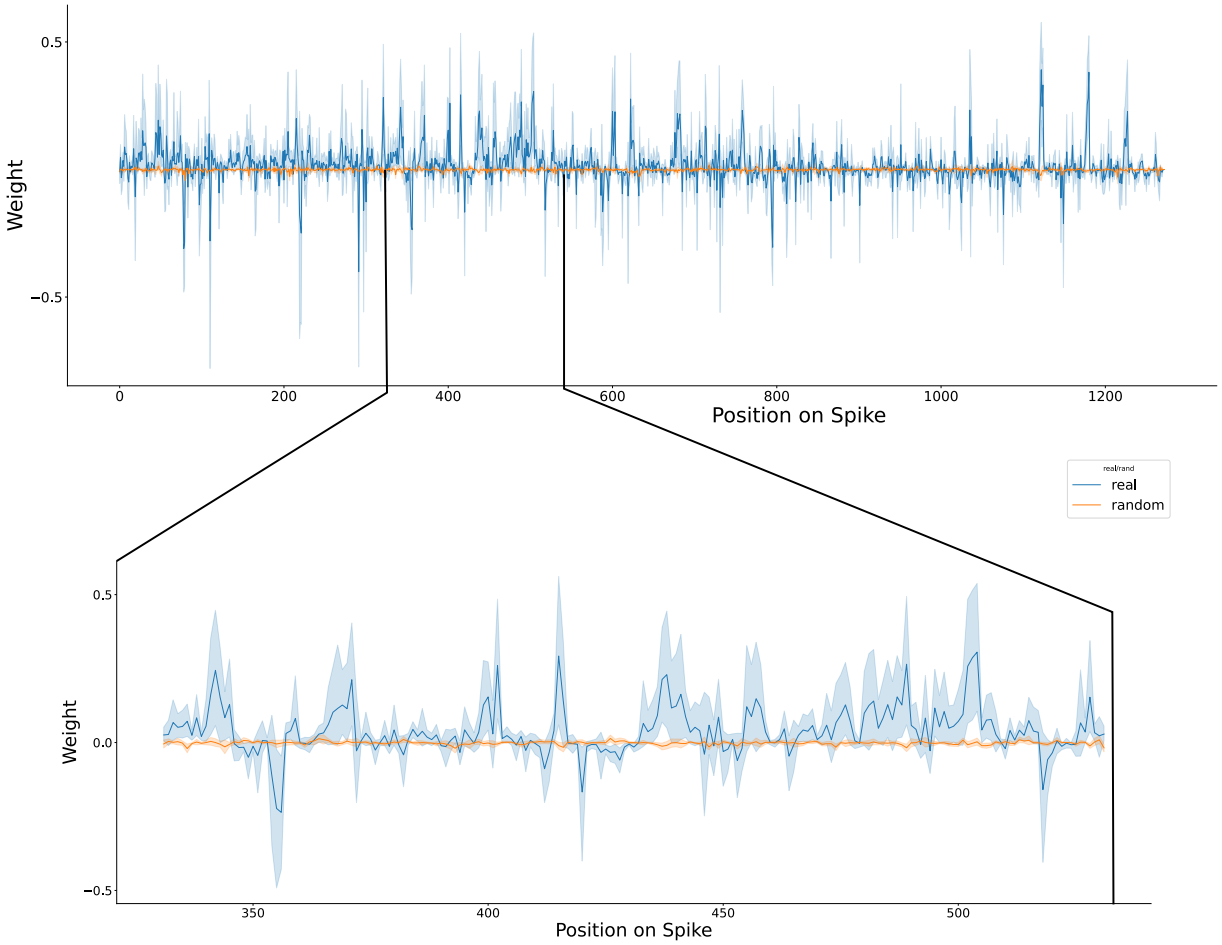


Figure 5.7: Comparison between the signals of multiple 'real' classifiers (blue) with the multiple random signal (orange) on a) Spike protein and b) focusing on the receptor binding domain (RBD).

the mean effect of all mutations at each position. The DMS results show that mutations in some positions having a deleterious effect on ACE2 binding due to inducing structural instability. Although mutations in these positions effect ACE2 binding they may not be directly involved in binding as demonstrated by the strong correlation between ACE2 binding and expression with a Pearson's correlation of 0.71. To account for this we used the difference between the mean effects on ACE2 binding and expression.

Figure 5.8 shows the non-parametric Spearman's correlation between the DSM signal and both real and random signals for each feature set. All the feature sets apart from one show weak negative correlation with the DMS data. The median correlation of all the real signals is -0.25 compared to the median for the random signals of 0.06. The correlation of the signal of the mean of the weights from all the feature sets at each position is -0.29. If our signals were drawn from a null model the probability that (9/9) correlations are less

than the median is 0.002 indicating that although the correlation is weak it is unlikely to be a random effect.

The correlation is negative as expected because sites that have a negative impact on ACE2 binding should have a positive association with the host and therefore a positive weight in our signals. Correlation is expected to be weak because both our predictive signals and the DMS data contain a complex mixture of confounding factors. The predictive signals will contain phylogenetic elements that non-functional due to the sequence homology within the different virus lineages. The DMS signal is complicated by any mutations causing a negative effect on binding due to structural changes.

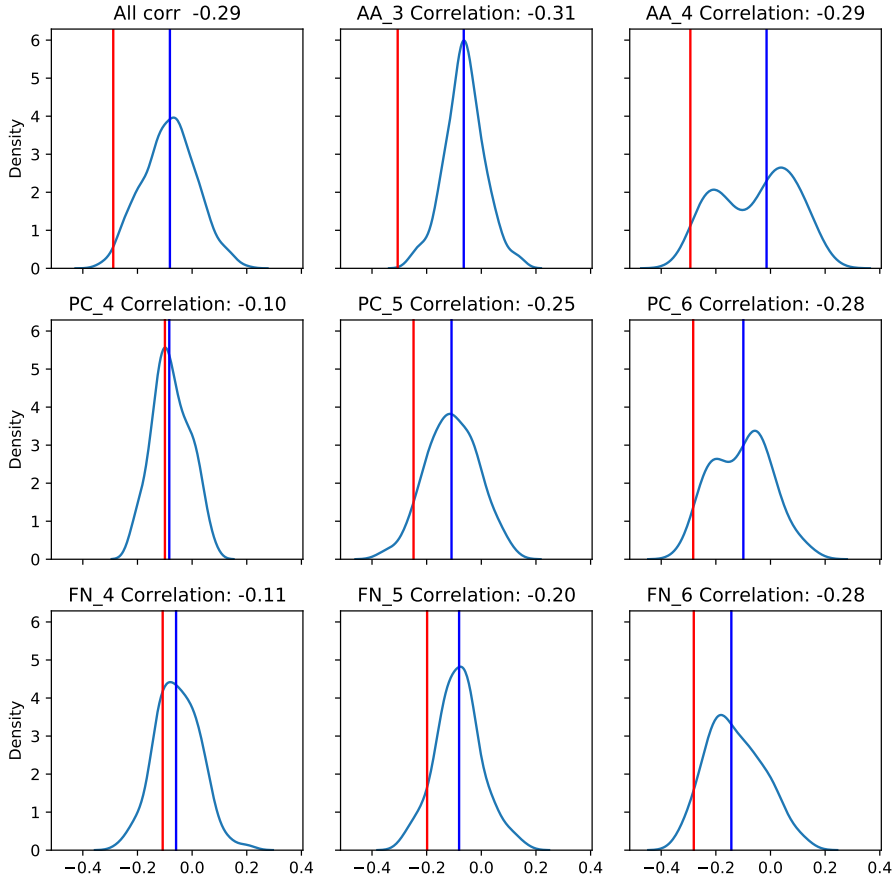


Figure 5.8: Spearman's correlation between our signals and the DMS signal. The subplots are the results for a single feature set. The correlation of DMS with the single real classifier is marked by the vertical red line and the distribution of the correlation with each of 100 random classifiers is shown by blue line with the median of the random correlations shown by the vertical blue line.

5.3.5 Non-phylogenetic element of the signal is predictive of host.

In the previous chapter we used a holdout method to demonstrate that the host specific signals in the viral genomes are a mixture of phylogenetic and non-phylogenetic signals. We believe that the non-phylogenetic features are more likely to be associated with functional elements, assuming they have arisen by selection onto some host factor vital to the viral life-cycle. It is these features that are of particular interest because they are potentially driving phenotype and may be indicate what changes have allowed a virus to switch.

Despite our feature set achieving high AUC scores for classifying all hosts as shown in Figure 5.1, none of the feature sets correctly classify the SARS-CoV-2 or SARS viruses as a human virus. Both SARS viruses are in the Sarbeco clade, the majority of which infect bats and are only distantly related to other human viruses, meaning that the classifiers are likely using signal of phylogenetic origin. We compared the weights on two alignments of the Betacoronaviruses, one ordered by host and one by the order in the phylogenetic tree. The dominance of phylogenetic element of the signal becomes very apparent with the clustering in the weights being much stronger when the in "tree" as compared to when ordered by their host, Figure 5.9.

Next we wanted to investigate if the feature sets contain kmers that may have arisen through convergence. To identify non- phylogenetic features we adapted the Scoary software (Brynildsrud et al. 2016), this method scores all the features for their association to observed traits while accounting for the phylogenetic relationships within the data. The r kmers are scored on their probability of being phylogenetic and those achieving a Bonferroni corrected p-value of less than 0.05 are selected.

We re-tested the classification using only the Scoary selected kmers and found only a slight change in the AUC scores as compared to using all the kmers. Again, using only the selected features no feature sets were able to correctly predict that SARS-CoV-2 is a human virus. Nevertheless, the classifiers are still highly predictive of host and we assume that predictive signal remaining due to the non-phylogenetic kmers is the result of convergence onto some host specific factors.

Interestingly, despite the SARS-CoV-2 having a predicted probability of being a Primate virus of 0.001 compared to 0.92 for a Chiroptera virus, regions of the receptor binding domain are positively weighted for the Primate classifier and negative in the Chiroptera classifier, Figure 5.10.

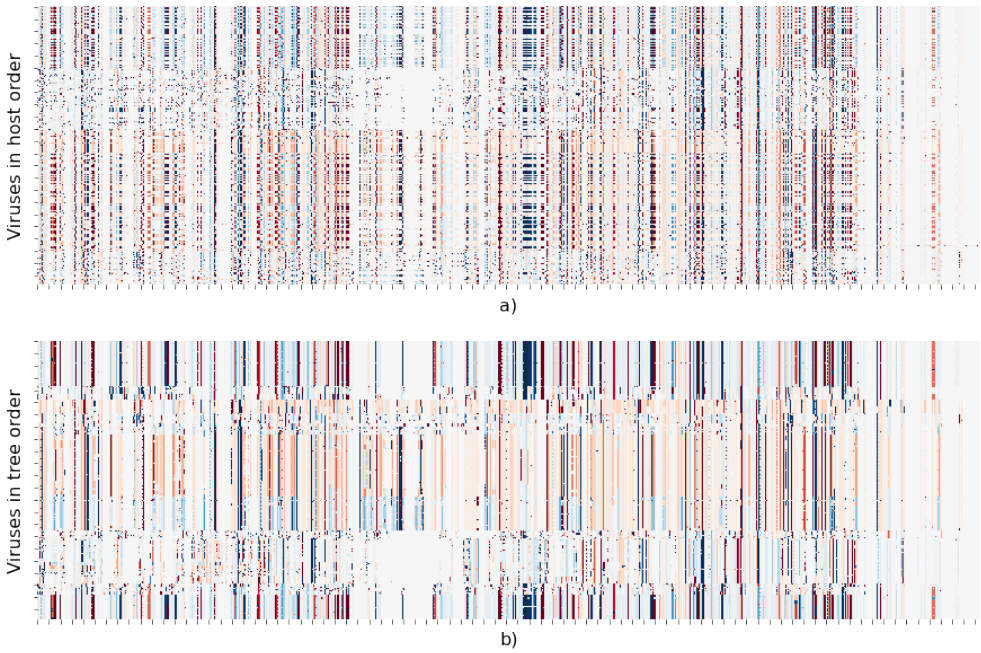


Figure 5.9: Comparison of the weights on a section of alignments of the Spike sequence, ordered by host and phylogeny. The weights plotted as a heat map on a short section of an alignment (x-axis is position) of the spike sequence for all the virus genome (y-axis). These are the weights from the classifier for the Primates label using the PC5 feature set. a) in host ordered b) ordered by tree. The colour indicates the weight at each position, red is positive and blue is negative. The min and max values reset to +/- 0.01 so that small absolute values are visible.

To establish whether these non-phylogenetic kmers were a likely to be a result of convergence, we compared the intersection of the sets of Scoary selected kmers in each of the five human lineages to the intersection of non-Scoary selected kmers. A kmer was included in a lineage set if it occurred in at least one of the sequences of that lineage. Figure 5.11 shows the intersection of the sets of kmers present in the different lineages a) for the Scoary only kmers and b) for the non-Scoary kmers. Although a much higher proportion of the kmers in the Scoary selected sets intersect with other lineages there are still many intersecting kmers in the non-Scoary set indicating that this method may not be ideal for this problem. Interestingly, the estimated times of emergence of OC43 and HKU1 are around 129 and 70 years ago respectively (Forni et al. 2017); we found that the largest groups of unique Scoary selected kmers occurred in the OC43 and HKU1 lineages. This is as would be expected, with more adaptions accumulating over the longer periods of evolution in their

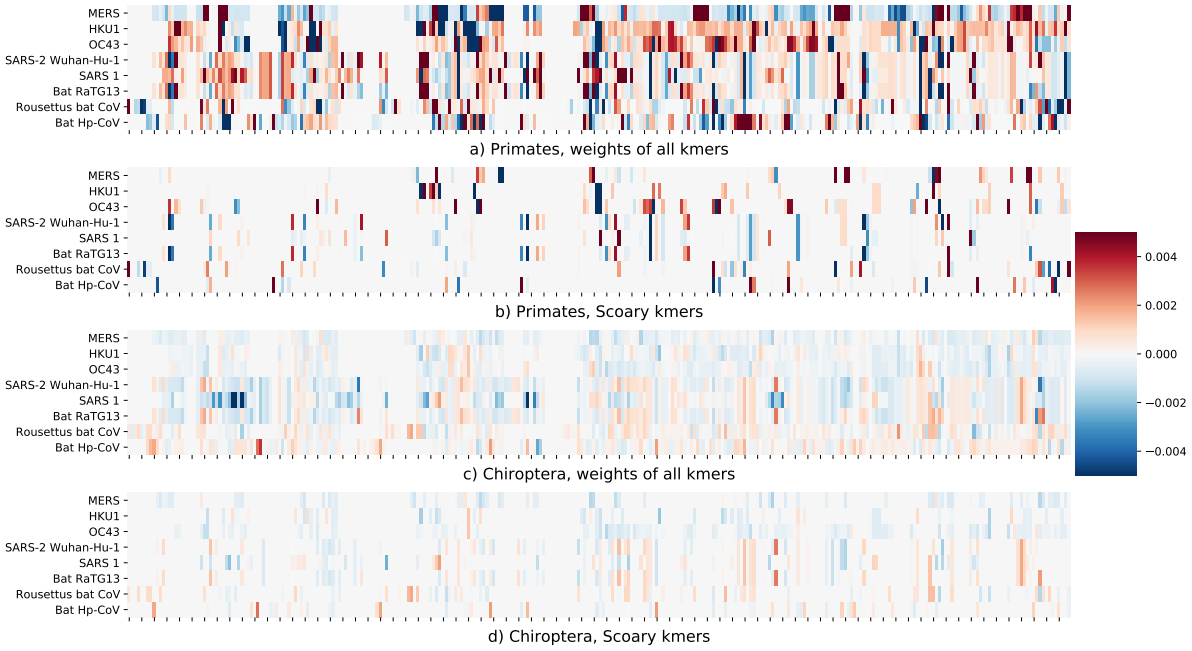


Figure 5.10: Heatmap of weights on an alignment for part of the RBD for a subset of viruses comparing the human (a and b) and bat classifiers (c and d) for all kmers (a and c) and Scoary only kmer (b and d).

human host.

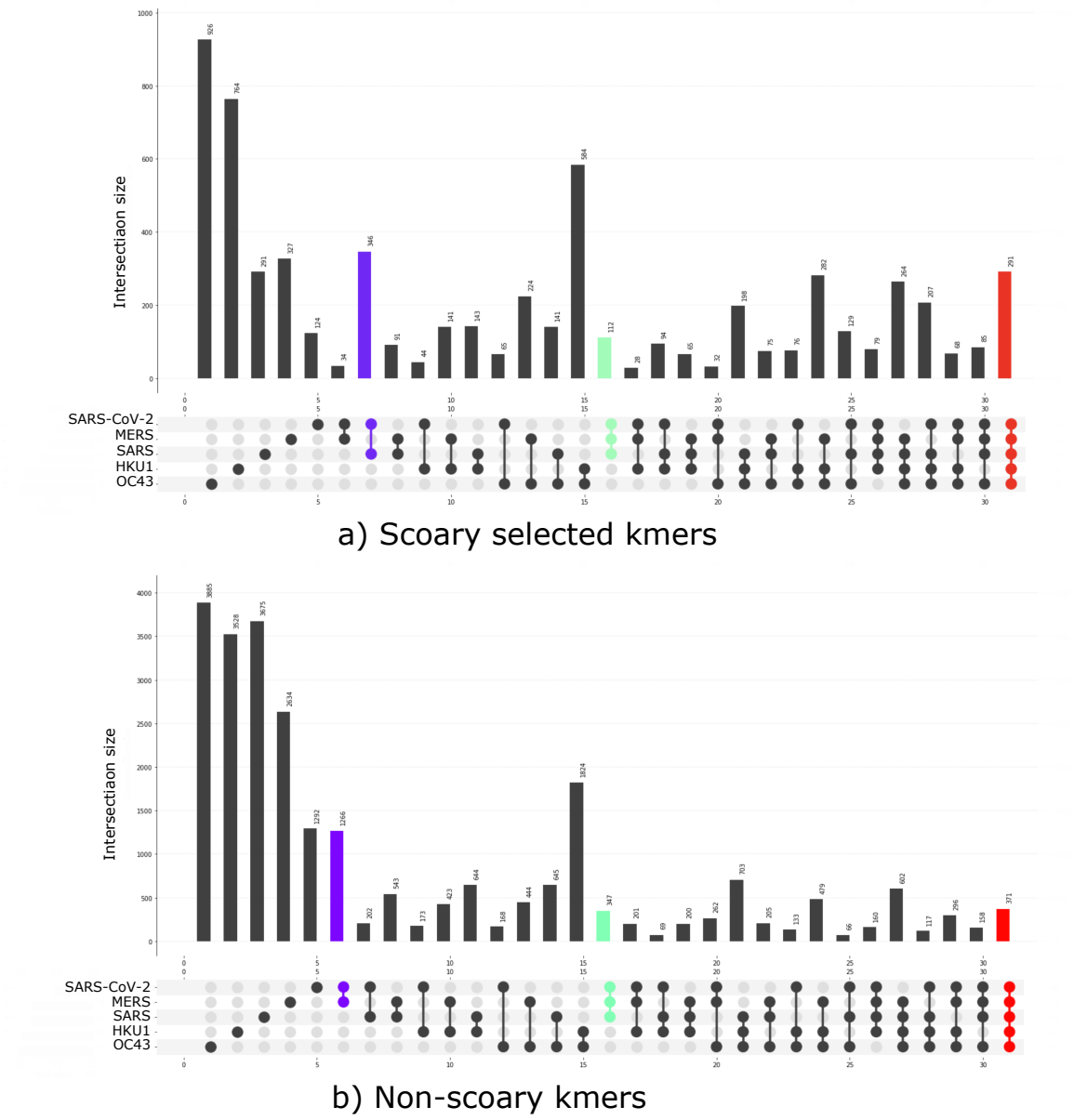


Figure 5.11: Intersection of the sets of kmers that occur across the different human virus lineages a) for the kmers selected by Scoary and b) the kmers not selected by Scoary.

5.4 Discussion

In this chapter we present a model-agnostic method that interprets host predictive classifiers by transforming a viral sequence into a host signal that indicates the relative importance of a site to host prediction. We used a range of appropriate feature sets, that can be uniquely located on the SARS-CoV-2 spike protein sequence, to train classifiers to predict different hosts of Betacoronaviruses. All of the classifiers achieved high AUC

scores indicating these feature sets contain discriminative features. We used the coefficients learnt by the model to assign a weight to each position in a sequence. This in effect transforms the sequence to a host signal indicating which regions are important for discrimination. By comparing these signals to a null model we showed that they contain real information. Furthermore, the correlation with the results of a DMS study indicates that some of this signal is functionally important. Finally, we used a phylogenetically aware method to identify features that are most likely to have arisen by convergence. Classifiers trained with this small subset of features were still highly predictive of host; these results support the idea that the predictive signals are a complex mixture embedded by different evolutionary mechanisms.

Despite achieving high AUC scores, all the classifiers incorrectly predicted that both SARS or SARS-CoV-2 have a bat host. This is in agreement with phylogenetic analysis that showed that the closest relatives of both SARS viruses are circulating in bats, indicating that these are bat viruses that have recently switched to humans, (Boni et al. 2020). Interestingly, even though they are predicted to have a bat host, Our results indicate that there are regions in the RBD with a stronger human signal than bat signal backed up by the correlation with the DMS signal. Given that was the original SAR-CoV-2 genome this perhaps indicates that it was primed for spillover to a human host.

A comparison of the mean of signals from the real classifiers to the signals from a random model indicates that the real signals were significantly more informative. In Figure 5.7 the real signal shown is the median of the signal of all the feature set weights at each position. The peaks in the combined signal strongly suggest that the weights from all the classifiers are strongly correlated giving further evidence that the signals contain real information. We used the results of a DMS study as the ground truth as to which sites of the RBD are important to binding the ACE2 receptor. Our signals show a weak but consistent negative correlation with our predictive signals. This correlation was strongest for the AA£, PC5 and FN5 feature sets, that is, those that balanced the trade off between being "locatable" and predictive, Figure 5.4.

One limitation of this approach is that virus datasets/genomes are inherently noisy because of their high rate of random mutations that will partially obscure any signals of selection. This means that random non-synonymous mutations causing a substitution in the amino acid sequence will change the kmers that include that position and any information about similarity with the starting kmer will be lost. We sought to minimise this loss by using a binned representation of the amino acid sequence for our feature sets allowing for functional

substitutions without breaking the kmers. Although this approach is more robust than either nucleotide or amino acid features, a more flexible way of capturing the nature of the interacting motifs would be preferable.

The signal is also complicated by the fact that there is not a direct linear relationship between the absolute size of SVM coefficients and their importance. It is known that for correlated groups of features their weights decrease as the sizes of the groups increase, thereby lowering their perceived importance and complicating the interpretation of the signal, (Toloşi and Lengauer 2011). Although this reduction in weight results in losing some of the important features, the higher weight features are still important in driving prediction.

Another source of complications is introduced by using all the betacoronaviruses as a data set. First, not all betacoronaviruses use the same domain to bind their receptor with examples of using either the CTD or NTD domains, (Zhu et al. 2018). Secondly, not all betacoronavirus bind the ACE2 receptor for cell entry, for example MERS uses DPP4. A comparison between the receptor binding domains of MERS-CoV and SARS-CoV show very different binding interfaces, (Wang et al. 2013). The receptor for many non-human viruses remains unknown making it difficult to select only viruses that use ACE2.

Further analysis needs to be done to assess the validity of this method. First by comparing our signals with data from other computational methods that identify potentially functional sites, such as: known motifs from the ELM database, predicted disorder, and surface accessibility. As the results of more studies into SARS-CoV-2 are released we will be able to use these to further corroborate our signals.

One issue that complicates interpretation of the signals is that classifiers do not distinguish between correlation and causation. Additionally, they contain a mixtures of signals, the ultimate goal for this type of approach is to understand which sites in a sequence are functionally important to the virus relationship. To do this we need to separate the features that indicate selection from those that have occurred purely due to phylogeny. In the next chapter we develop a method that aims to untangle these complex mixtures of signals.

Chapter 6

Multi-view Clustering

6.1 Introduction

Previously we have shown that virus genomes contain a complex mixtures of signals and that some of these signals are associated with their phylogeny and some are associated with their host. In Chapter 4 we developed a holdout method that demonstrated that there was a non-phylogenetic component to the the host predictive signal. This was backed up by further evidence in Chapter 5, where we adapted a method developed to link genes to traits while accounting for the phylogenetic stratification for our virus datasets and showed that it was able to select non-phylogenetic features. In this chapter we aimed to further investigate these mixtures by developing a Bayesian clustering method to tease apart this complex mixture of signals.

The complex mixture of signals in viruses sequence results in different subsets of features that will cluster the viruses into the alternative groupings or "views", as illustrated in Figure 6.1. In view 1, the similarity between viruses is defined by the virus phylogeny, as represented by the colour of the "body" of the virus, and reflects the 2 virus clades in the virus tree. In view 2 the similarity is defined by the viruses host as shown by the different "spikes" and reflects the two different hosts in the host tree.

There are different types of signal embedded in the viral sequences due to the different processes that influence viral evolution as discussed in Section 2.4. The phylogenetic signal which will cluster our virus according to group 1 in figure 6.1. This signal is due to divergent evolution between related viruses resulting in homologous features. The

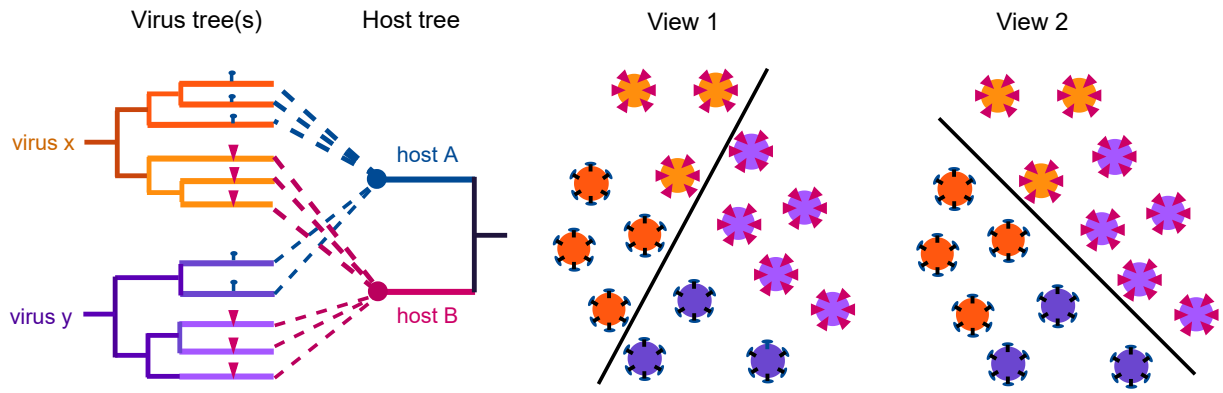


Figure 6.1: Different views of viruses.

non-phylogetic signals which will cluster our virus according to group 2 is a result of convergent evolution wherein distinct clades of viruses converge onto some host specific factor resulting homoplasic features. There is also a phylogenetic element that can group viruses by host. This is due to closely related viruses infecting the same or closely related hosts which will result in purifying selection maintaining features that are vital to the virus-host relationship. This is represented in the virus tree of figure 6.1 by the different tones used for the sub-clades of viruses that infect the different hosts. The challenge is that the "host" features are likely to be a minor effect in our high dimensional feature sets and that the majority of the features are likely uninformative due to random mutations that will obscure any common signals over time as viruses diverge. Based on our findings in chapter 5 we assume that the majority of the informative features are likely to be from the phylogenetic signal leaving only a very small subset of features that stratify the data by host.

6.1.1 Approaches to clustering

Here we give a broad overview of different approaches to clustering both for grouping individuals and for grouping covariates. For the rest of this chapter, we will use the term covariate as an application agnostic term for features, in other words the observed variables of an individual. Clustering or cluster analysis is a fundamental data exploration task which aims to discover the hidden groupings within data. The datasets used as the input to clustering are typically represented as a table, where each row contains multiple observations about an individual, with a column for each feature or covariate.

There are many different approaches for cluster analysis both in terms of grouping indi-

viduals based on similarity of the features or alternatively grouping the features based on correlation between them.

Traditional methods such as K-means and hierarchical clustering use a distance metric to group the most similar individuals into clusters. Although these methods are widely and successfully used there are major limitations: each cluster is represented as a single point; there are often multiple alternative clusterings and the one identified may be a local minima; as the number of dimensions increase distance measurements will converge. Bayesian mixture models take a probabilistic approach that increases flexibility by incorporating shape into the clusters and accounting for the uncertainty in the both the number of clusters and the alternative clusterings.

These methods find structure in the individuals, alternative approaches to find structure in the covariates.

Factor analysis is an approach used to uncover the structure in large sets of covariates. It aims to describe the variability among observed, correlated variables in terms of a lower number of latent variables called factors. A major limitation is interpreting the latent factors as multiple attributes can be highly correlated with no apparent reason. (Argelaguet 2018; Pournara and Wernisch 2007)

The bi-clustering is an approach to uncover groups of genes that are co-expressed in only a subset of samples using gene expression data. (Xie et al. 2019). Bi-clustering uses homogeneous gene expression to simultaneously cluster the samples and genes with the aim of identifying local patterns thereby uncovering more complicated clustering structure such as overlapping groups.

In all these methods the clustering is driven by the most dominant variation in the data that reflects the major factors influencing the individuals.

Approaches using Bayesian frameworks are very suited to be biological problems as it takes into account the natural distribution of the biological data. Harrison et al. (2020) show that a Dirichlet-multinomial model outperforms alternative methods for analysing shifts in relative abundance of groups from count data in ecological. The topic modelling method latent Dirichlet allocation, LDA, has been applied in a wide variety of biological contexts. (Liu et al. 2016) Words, (covariates), are allocated into latent topics as simultaneously documents,(individuals), are allocated a topic distribution. The topics are in effect groups of covariates and the distribution over topics can be used to group individuals. A major

limitation of LDA is difficulty in interpreting whether the discovered topics have any objective meaning. The results need careful validation against some ground truth, which is difficult in biology where we often have little or at best incomplete information.

All the methods described so far are unsupervised, any metadata about the individuals is not used in the clustering process. This information is used after clustering is complete to assess whether the resulting clustering structure make sense. Bayesian profile regression mixture models, PRM, (Molitor et al. 2010) links the disease outcome variable for each patient to a subset of correlated covariates. It uses the outcome variable to guide clustering and removing covariates that do not contribute to the structure. Again, profile regression tends to select the covariates that define the most dominant clustering. Paul Kirk has been developing multi-view clustering, MVC, an outcome guided approach, as yet unpublished (Kirk and Richardson 2021). This extends the profile regression model to a multi-view setting. It aims to overcome the issues when applying PRM to high dimensional data where the influence of the outcomes may be overwhelmed by the influence of non-informative covariates.

6.1.2 MVC for clustering virus genomes

In this chapter we investigate the idea of using multi-view clustering as an exploratory method to tease apart the different signals in the virus genomes. MVC is a Bayesian model based clustering method developed to identify biomarkers from a variety of OMICS data that stratify patients in the context of precision medicine. MVC is a generative approach built around multiple mixture models that each models an alternative clustering structure within data. The outcomes or other labels of interest are used to guide the model in an attempt to identify those covariates that may be associated with lesser effects in the data by assigning them to the "relevant view".

To adapt MVC to our virus problem we map the virus data to the precision medicine setup, as follows:

- The individuals: the patients → virus genomes.
- The covariates: the genes → kmer features.
- The response: the disease sub-type → viruses labeled by host taxa.

- View → The viruses grouped into one of alternative clusterings defined by a subset of the kmer features.
- The relevant view → The view that aligns with viruses grouped by their host label.

So that while precision medicine might be interested for example, in identifying genetic variants that stratify patients by a particular disease sub-type or outcome, we are interested in identifying features that stratify the viruses by host. By using features chosen to capture the functional properties of a protein sequence we hope to be able to identify regions of interest in a sequence that are functionally important to host specificity. Figure 6.2 shows how the multi-view clustering might work.

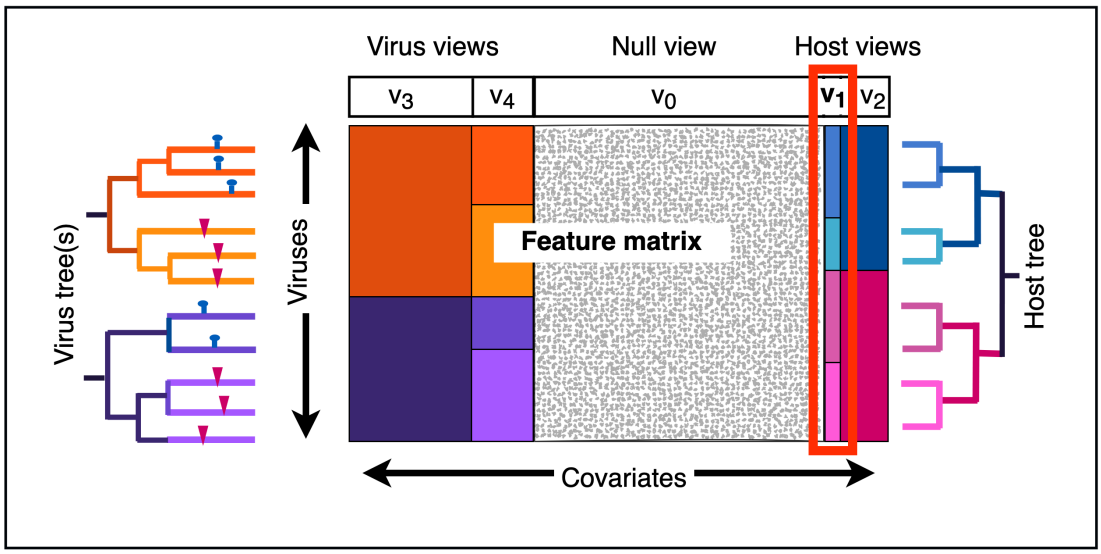


Figure 6.2: Proposed MVC workflow applied to virus genomes. The different views output from the MVC model. Each view is a subset of the covariates with a different clustering structure where the viruses have been sorted by their cluster assignment in that view, note that the order of the viruses (rows) in each view will be different. Here view 3 is a "virus view" of the data as in grouping 1 and view 2 is a "host view" as in grouping 2 from Figure 6.1 above.

The aim of this chapter was use MVC to uncover the alternative clustering structures in our virus data and to explore it's potential to identify the functional features associated with host. We hoped to use the MATLAB implementation shared by Paul Kirk but due to a discrepancies between the maths extracted from the code and our derivation of the model equations we re-implemented the model in python. The workflow for applying MVC, Figure 6.3, to virus data consists of three stages. Firstly pre-processing the viral genomes into categorical feature matrices and labelling the genomes with metadata of interest.

Next the the joint probability distribution of the mixture model is sampled using a Gibbs sampling scheme. Finally the inferred latent model parameters must be interpreted. For example posterior probability of covariates being a view or an individual in a cluster.

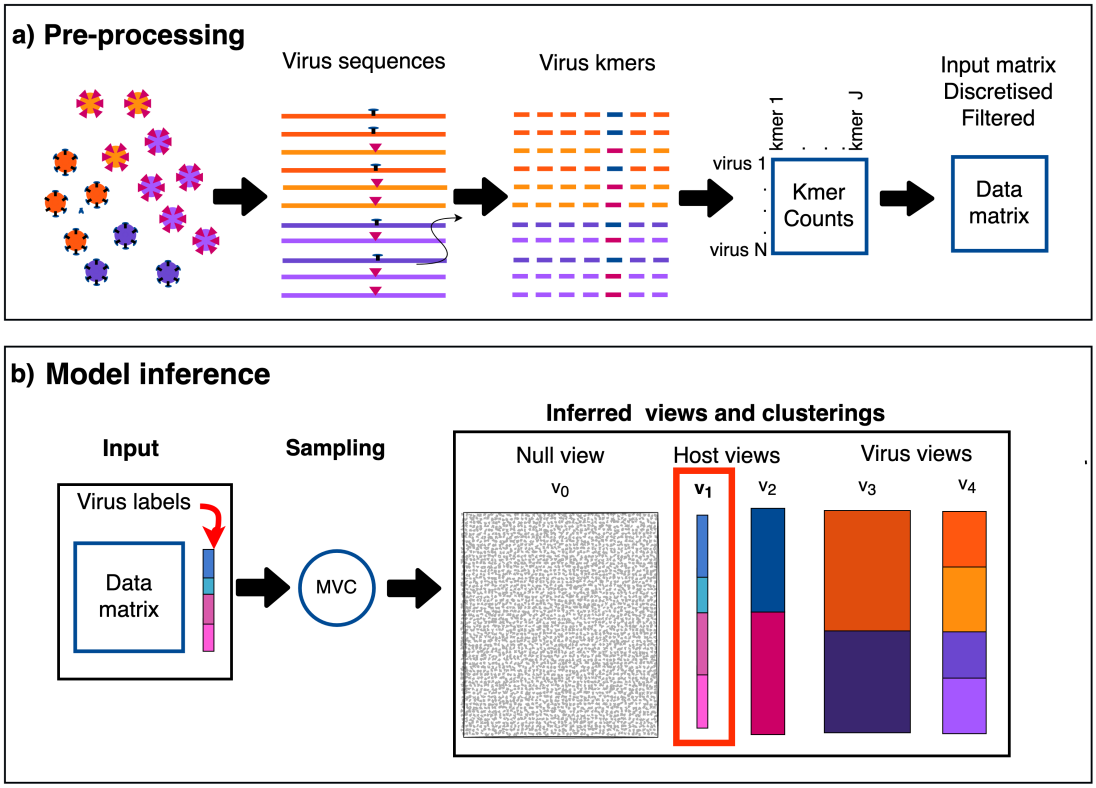


Figure 6.3: Proposed MVC workflow applied to virus genomes.
a) Pre-processing the viral genomes into a suitable data matrix for the MVC model.
b) MVC is used to model the input data generated above, the clustering inferred from the model is the allocation of covariates to views and the clustering of viruses within each view.

6.2 The Multi-view Clustering Model

The multi-view clustering method is a Bayesian model can be thought of as two step process. In the first the data is split into views by covariates, such that each view only uses a subset of the covariates. In the second the data is split by individuals into clusters, with each view having a different clustering structure. The model is shown the plate diagram, where the shaded variables are the observed data X and y,the rest are latent variables that are sampled from the posterior distribution.Figure 6.4.

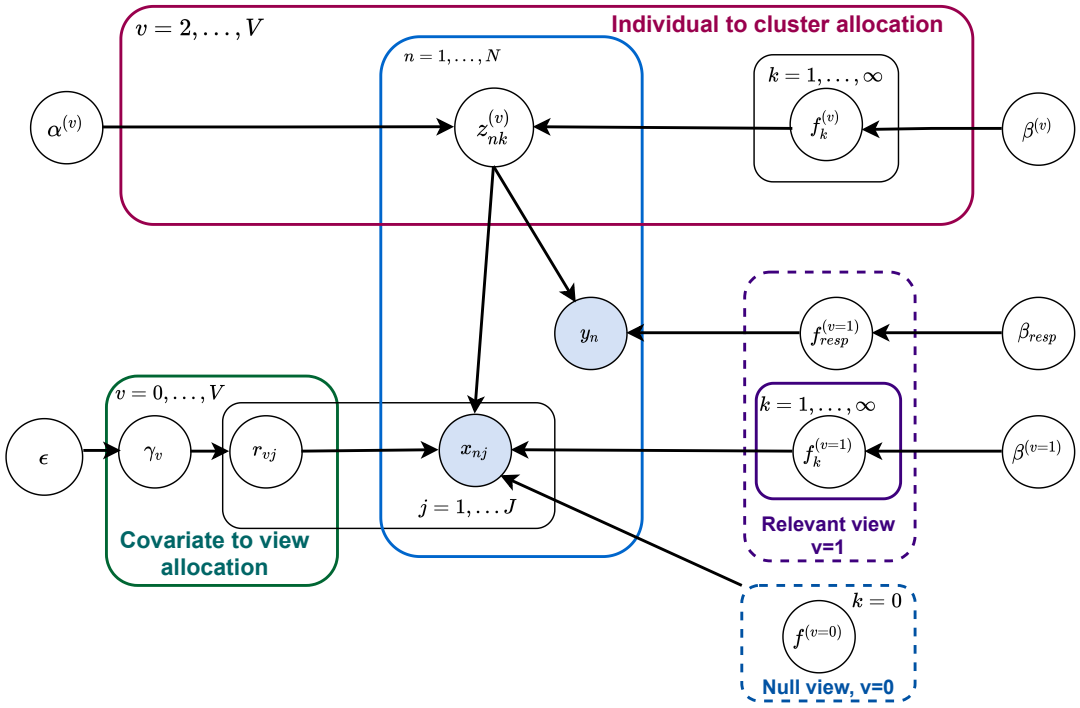


Figure 6.4: Plate diagram of the multi-view clustering model showing the two steps. 1. Individual to cluster allocation within each view. 2. Covariate to view allocation in which each view is assigned a subset of the covariates. The two special views, the Relevant view and the Null view. The shaded variables are the observed data.

6.2.1 The input

Given a dataset X consisting of n individuals $\in \{1...N\}$, each described by $j \in \{1...J\}$ categorical covariates which can take on levels $l \in \{1...L\}$. So that $x_{nj} = 2$, means that the covariate j for individual n has a value of 2.

The data is split into views, $v \in \{1, ..., V\}$ by covariates such that each view only uses a subset of the J covariates. Let r_{vj} be a binary indicator variable, $r_{vj} = 1$ if covariate j is in view v, otherwise 0, $\mathbf{r_j} = [r_1, ..., r_v]$ and $\sum_v^V r_{vj} = 1$.

Within each view the individuals are assigned to one of K_v clusters. Let z_{vnk} be a binary indicator variable so that $z_{vnk} = 1$ if individual n is in cluster k of view v, $\mathbf{z_{nv}} = [z_1, ..., z_{K_v}]$ and $\sum_k^{K_v} z_{nvk} = 1$.

Each cluster in each view is described by a multinomial for each covariate with probability $\mathbf{f}_{vkj}$, a vector of length $L \in \{0, ..., L\}$ where $\sum_l^L f_{vkjl} = 1$.

Let $\boldsymbol{F}$ be a matrix containing the probabilities for all the distributions within the model,

then $\mathbf{F}_v$ is a $K_v \times J_v \times L$ matrix where, K_v is the number of clusters in view v, and J_v is the subset of covariates assigned to that view.

6.2.2 The model

The data described above is assumed to be generated as follows

Each covariate to view allocation is one draw from a multinomial with parameters $\boldsymbol{\gamma}$

$$\boldsymbol{\gamma} \sim Dir(\epsilon)$$

$$r_{vj}|\gamma_v \sim Multinomial(1, \gamma_v)$$

Each individual to cluster allocation for each view v is one draw from a multinomial with parameter $\boldsymbol{\pi}$

$$\pi_v|\alpha \sim Dir(\alpha_v)$$

$$z_{vnk}|\pi_v \sim Multinomial(1, \pi_v)$$

Each variable in $\boldsymbol{X}$, x_{nj} is one draw from the multinomial with parameter $\boldsymbol{f_{vjk}}$

$$f_{vkj} \sim Dir(\beta)$$

$$x_{nj} \sim Multinomial(1, f_{vjk})$$

The full joint distribution for the model is

$$p(X, r, f, z|...) = \left[\prod_{n=1}^{N}\prod_{j=1}^{J}\left[\prod_{k=1}^{K}\left[\prod_{l=1}^{L}(f_{vkjl})^{x_{nj}=l}\right]^{z_{vnk}}\right]^{r_{vj}}\right] p(\mathbf{r}|\gamma)p(\gamma)p(\mathbf{F}|\beta)p(\mathbf{Z}|\alpha,\beta) \quad (6.1)$$

6.2.3 Inference

A Gibbs sampling scheme was implemented for inference from the joint posterior distribution 6.1 above. As described in section 3.3.8, each parameter is sampled iteratively conditioned on the values of all the other parameters. This is done via a two step process: first a Gibbs sampler is used to update the covariate to view allocations, r_{vj}, these are sampled with a standard mixture model conditioned on the clustering structure from step

two; in the second step a collapsed Gibbs sampler is used for each view, the individual to cluster assignments z_{vnk} are sampled via an infinite mixture model using the subset of the covariates assigned in step one.

6.2.3.1 Covariate to view allocation

A standard mixture model is used to sample the covariate to view allocation, r_{vj}, this means that the number of views must be defined by the user. The multinomial parameters, f_{vkjl}, have not been marginalised. In this step, at each iteration we are sampling r_{vj} and f_{vkjl}.

The prior probability for r_{vj} is given by $P(r_{vj} = 1|\gamma = \gamma_v)$ where $\gamma_v \sim Dir(\epsilon)$ is a multinomial such that $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_V]$ and $\sum_v^V \gamma_v = 1$. As shown in 3.3) the posterior probability for a multinomial with a Dirichlet prior is another Dirichlet, this is given,

$$P(\gamma|...) = Dir(\epsilon_1 + \sum_j r_{1j}, ..., \epsilon_V + \sum_j r_{Vj}) \tag{6.2}$$

where $\sum_j^J r_{vj}$ is the number of covariates in a view. We can marginalise γ so that the posterior probability the covariate to view allocation becomes:

$$P(r_{vj} = 1|\gamma) = \frac{S_v^{-j} + \epsilon}{J - 1 + \epsilon V} \tag{6.3}$$

where S_v^{-j} is the number of covariates in a view excluding the current covariate.

Similarly the conditional distribution of the multinomial probabilities f_{vkjl} is given by

$$P(f_{vjkl}) = Dir(\beta + \theta_{vjk1}, ..., \beta + \theta_{vjkL}) \tag{6.4}$$

where θ_{vjkl} is the counts of the levels in each covariate in each cluster.

The view allocation is sampled from the posterior distribution for r_{vj} conditioned on X,F,Z and γ where the prior probability for each view is γ and the likelihood of each covariate is given by q_{vj}

$$P(r_{vj} = 1|...) = \frac{\gamma_v q_{vj}}{\sum_{v'}^{V} \gamma_{v'} q_{vj'}} \tag{6.5}$$

where q_{jv} is the likelihood for covariate j being in view v with clustering structure defined by z_{nvk} is given by

$$q_{vj} = \prod_{n=1}^{N} \prod_{k=1}^{K_v} \left[\prod_{l=1}^{L} (f_{vkjl})^{x_{nj}=1} \right]^{z_{nvk}} \tag{6.6}$$

We also tested using a standard mixture model for sampling and updating γ prior to step 1.1 in algorithm 2 below, followed by sampling r_{vj} conditioned on $\mathbf{F}$, steps 1.1 and 1.2. This means that the likelihood of each covariates in a view is independent of the rest of covariate allocations allowing us to use a block update, this is known to overcome the problem of poor mixing in models with high numbers of covariates.

6.2.3.2 Individual to cluster allocation

Each view is treated as a separate infinite mixture model, as described in section3.3.8 that uses a subset of the covariates, j_v given by r_{vj}. In a standard mixture model, using all covariates $j \in \{1...J\}$, the probability that individual n belongs to cluster k is given by:

$$p(z_{nk} = 1|\mathbf{F}) = \pi_k \prod_{j=1}^{J} \prod_{l=1}^{l} (f_{vkjl})^{(x_{nj}=l)}$$

where $x_{nj} = 1$ if $x_{nj} = l$, otherwise 0. By raising this to the power of r_{vj} we will only include the covariates switch on in the current view. As shown in background/appendix? that we can marginalise on both π and f in equation 6.1 and assuming an infinite number of clusters to get a conditional

$$P(z_{nvk} = 1|...) \propto \frac{c_{kv}^{-n}}{\alpha_v + N - 1} \times \prod_{j=1}^{J} \left[\frac{\theta_{vkjl}^{-n} + \beta_l}{\sum_{m=1}^{L} \theta_{vkjm}^{-n} + \beta_m} \right]^{r_{vj}} \quad \text{for existing clusters} \tag{6.7}$$

$$P(z_{nvk} = 1|...) \propto \frac{\alpha_v}{\alpha_v + N - 1} \times \prod_{j=1}^{J} \left[\frac{\beta_l}{\sum_{m=1}^{L} \beta_m} \right]^{r_{vj}} \quad \text{for new clusters} \tag{6.8}$$

Let Z_v^{-n} denote the cluster assignments for all individuals excluding individual n in view

v. The number of individuals in cluster k of view v is c_{vk}^{-n}, and θ_{vkjl}^{-n} is the count for the covariates j, of levels l, in cluster k of view v, again $-n$ indicating that the current individual is excluded from both counts.

6.2.3.3 Special views

There are two special views, **"the relevant view"** which uses cluster membership to link subsets of correlated covariates to the labels associated with the data, and the **"null view"** which is fixed with one cluster with the aim of gathering all the non-informative covariates.

The relevant view uses the likelihood of an individual being in cluster given the labels as and additional term when updating individual to cluster allocation. Each individual is assigned a label $a \in \{1...A\}$ so that for the relevant view only equations 6.7 and 6.8 have the additional products to be included in equations 6.7 and 6.8 respectively.

$$P(z_{nvk} = 1|...) \propto \frac{S_{kr}^{-n} + \beta\_res_a}{\sum_{m=1}^{A} S_{km}^{-n} + \beta\_res_m} \text{ for existing clusters} \tag{6.9}$$

$$P(z_{nvk} = 1|...) \propto \frac{\beta\_res_a}{\sum_{m=1}^{A} \beta\_res_m} \text{ for new clusters} \tag{6.10}$$

Where β_res is the Dirichlet prior for the responses and S is the count of the response levels in each cluster.

The null view is used to collect any uninformative covariates, that are assumed to be described by a single multinomial for all individuals in the data set. In the MVC model the null view is effectively a finite mixture model with one cluster so that Step 2 of the Gibbs sampling algorithm 2 below is skipped.

6.2.3.4 Initialisation

The covariate to view allocations are initialised with all covariates in the null view, i.e. $r_{vj} = 1, \forall\, j$ where v=0, the null view and $r_{vj} = 0$ for all other views.
For the relevant view the individual to cluster allocation, Z, are set to match the labels of the data.

All other views are infinite mixture models and the number of clusters and the individual to cluster allocation are initialised using the Chinese restaurant process(CPR). This is a stochastic process whereby the probability of an individual joining a cluster is proportional on the number individuals already in the cluster,c_k or α new clusters. α is set high to encourage lots of clusters.

$$P(k) \propto \begin{cases} c_k & \text{for currently occupied clusters} \\ \alpha & \text{for currently empty clusters} \end{cases}$$

Initialisation: Set **r** such that all covariates start in the null view, $r \in \{0, 1\}$ and $\sum_v r_{vj} = 1 \; and \; \forall \; j$
Set Z for the relevant view to the labels of the data. For all other views initialise Z following CRP;
for *required number of iterations* **do**
  1. **Update and sample the covariate to view allocation r.**
  1.1 Update $f_{vkj} \; \forall \; vjk$ with equation 6.4;
  1.2 Update r_{vj} with equation 6.5;
  for *view in V* **do**
    2. **Update and sample the individual to cluster allocation Z_v.**
    for *n in N* **do**
      Update Z_v with equations 6.7 and 6.8, for relevant view only include terms 6.9 and 6.10 respectively;
    end
  end
end

Algorithm 2: Gibbs sampling scheme

6.2.4 Interpreting the Inference results

Inference within the model is performed by the Gibbs sampling scheme set out in above. The main model parameters we are interested in inferring is the covariate to view allocation from step 1. This allows us to calculate the following metrics.

Posterior probability of view membership. We calculate the posterior probability of covariate to view membership by taking the mean number of times each covariate was

assigned to a view during the sampling. We define the the final view allocation as that view with the maximum posterior probability. Of special interest is the posterior probability of a covariate being assigned to the relevant view. This is in effect a measure of its association with the labels of interest, in our case the host.

Co-occurrence of covariates in views. The posterior probability that two covariates are allocated to the same view is computed using co-occurrence matrix. This matrix is generated by adding the pairwise co-occurrence of covariates in a view from each of the posterior samples . By sorting the covariates in this matrix by view membership generated above we can visualise how well the model is able to identify views.

Co-occurrence of individual in clusters. To compute the clustering structure of each view we compute the co-occurrence of individuals in clusters. Label switching that occurs in an infinite mixture model means that we cannot use the the cluster membership directly but within each view we can compute the posterior probability for the co-occurrence of individuals in the same cluster from the sampled cluster allocation.

6.2.5 Comparing clustering structures

To compare the clustering structures in the views with ground-truth metadata we used Adjusted Rand Index (ARI), (Hubert and Arabie 1985). ARI is a widely used to metric to compare clustering structures that is adjusts for the expected similarity for a random model. The Rand index RI, scores the number of pairs of individuals assigned to the same cluster over all pairs and is directly related to accuracy when one of the clusterings is of the ground truth. The ARI is the correct-for-chance version of RI of using the expected similarities for a random model. An ARI of 1 indicates identical clusterings whereas an ARI score of 0 indicates that the similarity is the same as that expected by chance meaning that ARI can take on negative values if clusterings are less similar than expected in a random model.

6.3 Simulation Study

We conducted simulation experiments using synthetic data to test whether the MVC model can extract the kinds of patterns we would expect to see in the data. Firstly, for proof of concept, does the model find the different views in the data and is the relevant view, v=1, the labelled view. These initial test also check that the maths and code are correct. The second set of simulations were used to check whether the model can identify the covariates associated with a smaller weak effect in the data from a much larger more dominant signal. In the third set of simulations we investigated the effects of changing the hyperparameter, V_m, the number of views in the model. How does changing the number of views modelled effect whether the relevant view is still identified. When using synthetic data we can set the number of views in the model to the same as that used to generate the data V_d, whereas in the real data this is unknown and likely to be complicated by the hierarchical nature of the data.

6.3.1 Generating synthetic data

A datasets X consisting of N individuals with J covariates were generated by sampling a probability distribution similar to the MVC model. The multinomial parameters used to assign the individuals to one of the K_v clusters within a view are drawn from a Dirichlet distribution parameterised by α . The multinomial parameters used to assign the level of the each data point $x_{nj} \in \{0, ..., 2\}$ is drawn from a Dirichlet distribution parameterised by β_v. Using different values of β for different views allows to control how sparse the values for a cluster are, when using a low β, say 0.1, all values are more likely to be the same within a cluster, whereas using a high β the values will be more evenly spread across the levels, in effect emulating noisy the data.(see section backgound)

The user assigned parameters that are set to test different scenarios are:

- V or V_{data} - the number of views within the data.
- N - the number of individuals or viruses.
- J - the number of covariates.
- γ - the Dirichlet parameter which effects how evenly the J covariates are assigned to the V views.

- K_v - the number of clusters within each view.
- α - the Dirichlet parameter that effects how evenly the individuals are assigned to the K_v clusters in each view.
- β_v - the Dirichlet parameter that controls how sparse of uniform the data within a cluster is.

The main parameters tested within this simulation study were V, N, J and β and those uses for each synthetic data set are shown in table A.1. The lower β the more sparse the data is and hence the more defined a cluster is, by setting this to higher for the relevant view than the other views we can test how well the model can pull out the weaker effect from the data.

6.3.2 Simulation results

The results for each simulation are shown in figures 6.5, 6.6, 6.7 each consist of three parts:

(a) the raw data, with the columns sorted according to view allocation of the covariates, and the rows are according to the cluster allocations of the individuals in view 1, the relevant view;

(b) the view allocation of the covariates in the generated data as in the ground truth;

(c) a heatmap of the co-occurrence of covariates in views.

By comparing the ground truth views, shown in plots (b), with the corresponding view allocations that are inferred from the model, shown in plots (c), we get a visual demonstration of how well the model is working. The colors in the heatmap indicate the level of co-occurrence between two covariates over all the samples. Yellow indicating high co-occurrence, and dark blue indicating that the covariates rarely or never co-occur in the same view. For all figures the left hand view, v= 0, is the null view. The second view from the left is the relevant view, v=1.

6.3.2.1 Proof of concept

The results of the initial simulations demonstrate that the model is able allocate the covariates to the correct views. We tested datasets with the number of individuals much larger than the number of covariates and also with the number of covariates much larger than the number of individuals. In both cases the model correctly assigns all the covariates in the relevant view. The null covariates that are each drawn from a single distribution are assigned less consistently.

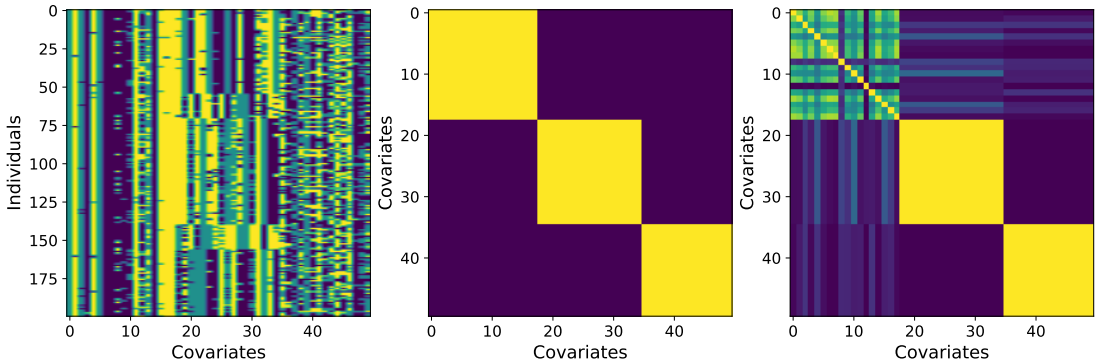
In the third test, figure 6.5c the multinomial distribution for the relevant view is more disperse than all other views by setting the Dirichlet parameter for the relevant view, $\beta_{v=1}$, higher than β for the other views, 0.5 compared to 0.1. The model ws still able to correctly correct clustering structure emerged for all the views apart from the null view, where although the null covariates were only assigned to the null view more often than other views there was a higher probability that they would be assigned to other views. For all simulations in 6.5 the models appeared to converge quickly and were only sampled for 200 iterations after a burn-in period of 50 iterations.

6.3.2.2 Testing limits for finding a weak effect in the relevant view.

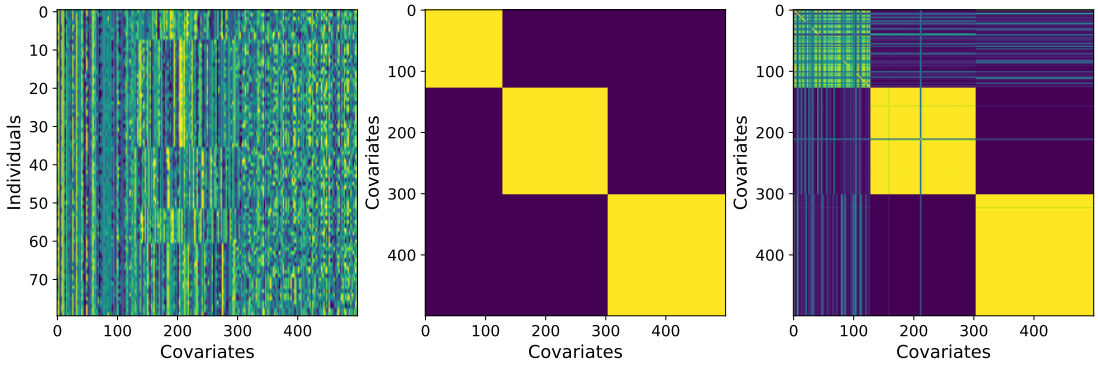
In the second set of simulations we aimed to test how much smaller and weaker the relevant view can be and still be correctly identified. Here we define a smaller effect as in the number of covariates in the relevant view compared to the dominant view. A weaker effect is defined by a comparatively high β_v, the Dirichlet parameters used to draw the multinomial parameters for the clusters in view v. The results shown in , Figure 6.6, demonstrate that model is able to correctly identify the covariates generated in the relevant view. in both these scenarios.

6.3.2.3 Testing how the hyperparameter V influences the model

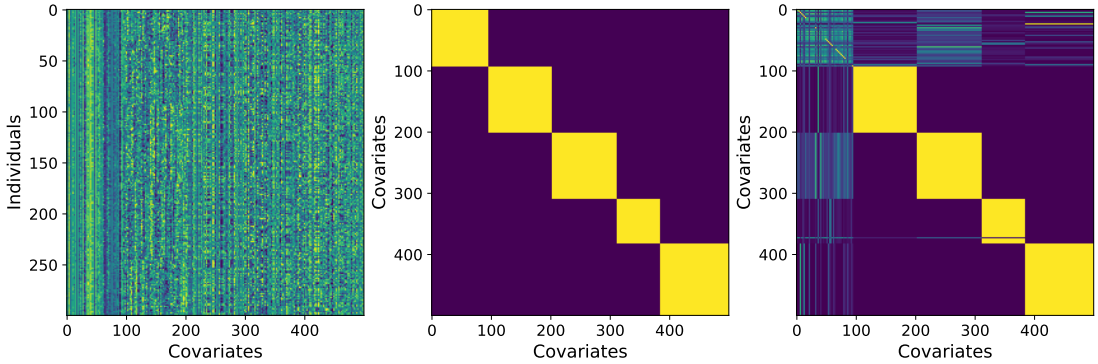
Next we wanted to test the effect of selecting the number of views on the ability of the model to find structure within the data. When using synthetic data we can set the number of views in the model V_m to the same as that used to synthesise the data, V_d. When using real datasets the number of views in the data will be unknown. In this simulation we investigate the effect of the choice V_m on the ability of the model to find the relevant



(a) $N = 200 > J = 40$



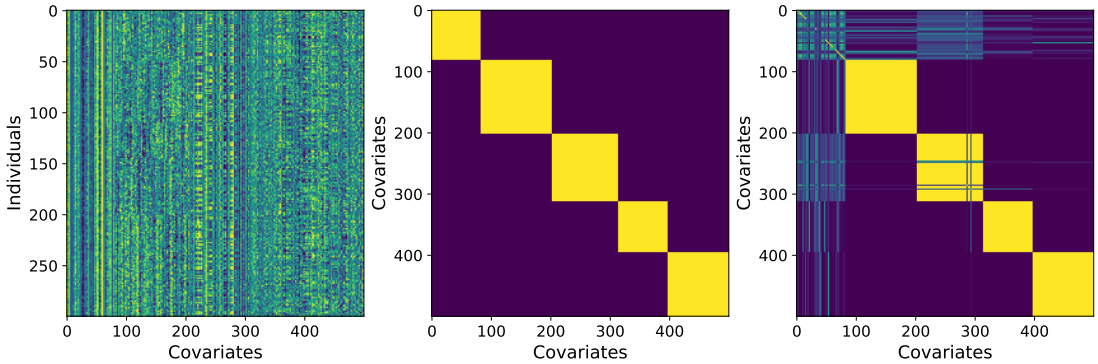
(b) $N = 100 < J = 500$



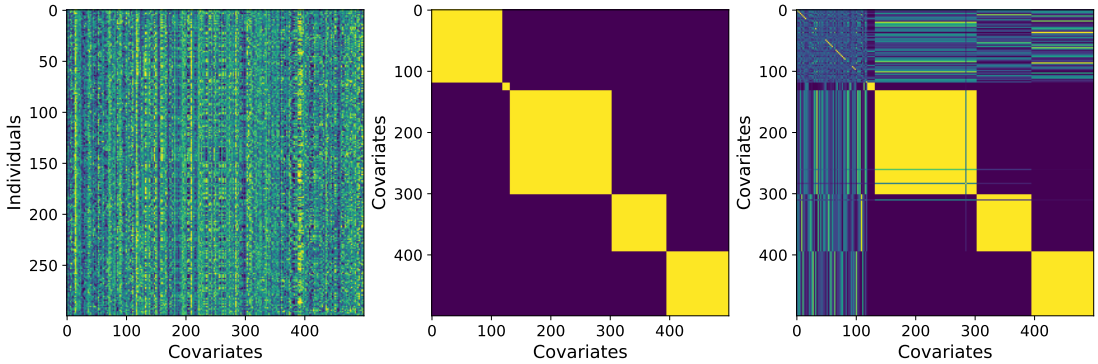
(c) $N = 100 < J = 500$, $\beta_{v1} = 0.8 > \beta_v = 0.1$

Figure 6.5: Simulation 1: Proof of concept.
a) Large number of of individuals N compared to the number of covariates J.
b) Large number of covariates J compared to the number of individuals N.
c) Large J, small N, with the less defined clus$\beta_{v1} > \beta_v$.

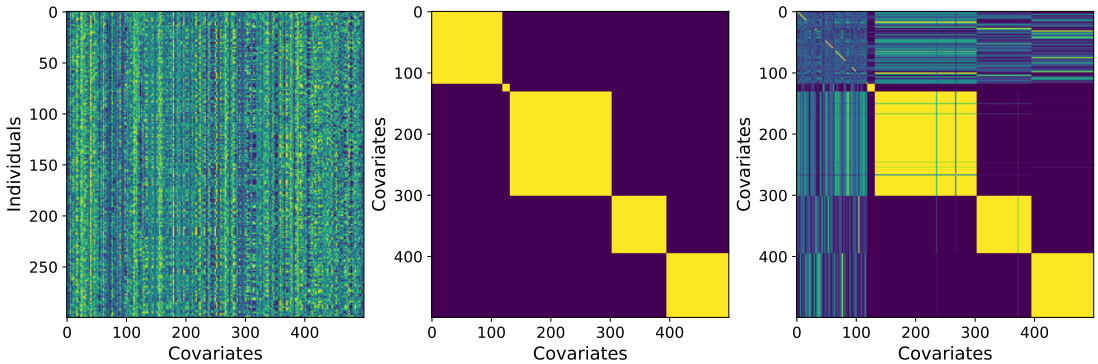
view. The minimum value of V_{model} is 3 as the MVC model must always have a null view, a relevant view and at least one other view. As can be seen from Figure 6.7b the relevant view is identified even when using the minimum of three views to model the data. Interestingly, when V_{data} is larger than V_{model} such as in figure 6.7c more than the eight model views are emerging in the co-occurrence matrix. Likewise when V_{data} is smaller than



(a) $J_{v=1} \sim J_{v\neq 1}$, $\beta_{v=1} > \beta_{v\neq 1}$



(b) $J_{v=1} < J_{v\neq 1}$, $\beta_{v=1} \sim \beta_{v\neq 1}$

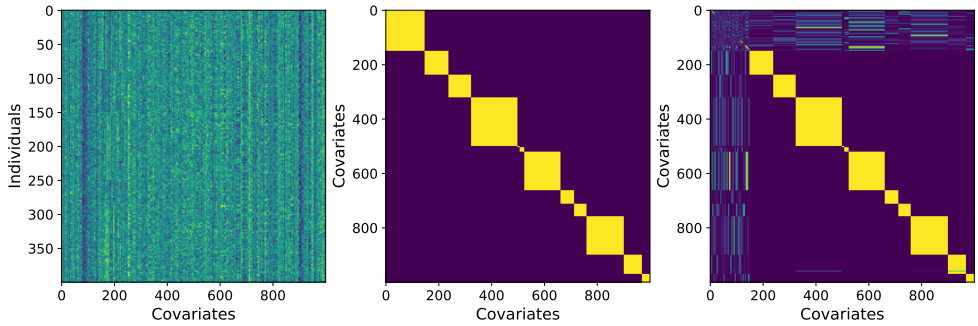


(c) $J_{v=1} < J_{v\neq 1}$, $\beta_{v=1} > \beta_{v\neq 1}$

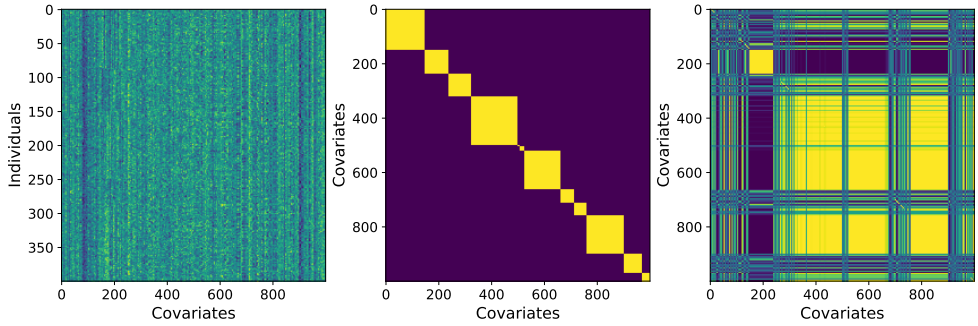
Figure 6.6: Simulation 2: Testing limits of the models ability to identify the covariates associated with the relevant view when the relevant view is a weak effect in the data .
a) $\beta_{v=1}$ for the relevant view is much higher,(1.0) , compared to all other views $\beta_{v\neq 1}(0.1)$.
b) The number of covariates in the relevant view $J_{v=1}$ is much less than in other views, $J_{v\neq 1}$.
c) Both low numbers of $J_{v=1}(12)$, and high $\beta_{v=1}$ (0.8).

V_{model} such as in figure 6.7d only the twelve data views are found. Unfortunately, the run time for sampling the model is directly proportional to the number of views in the model, so in order to achieve sensible run times with these larger datasets it may be necessary to

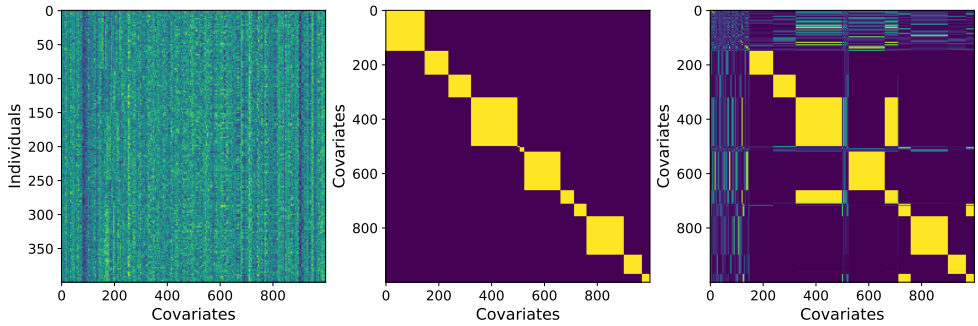
use the minimum number of views necessary to obtain convergence.



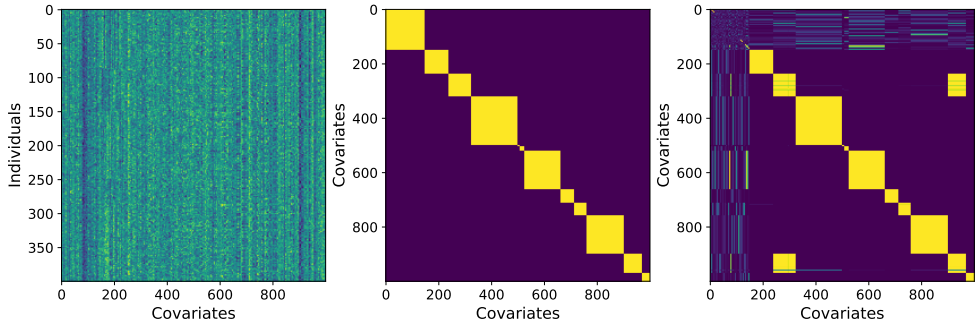
(a) $V_{data} = 12$, $V_{model} = 12$



(b) $V_{data} = 12$, $V_{model} = 3$



(c) $V_{data} = 12$, $V_{model} = 8$



(d) $V_{data} = 12$, $V_{model} = 20$

Figure 6.7: Simulation 3. Testing the effect of the number of views used for the model. The same data set is used for the runs, with $V_{data} = 12$. Tests were run with: a) V_{model} set equal to V_{data}; b) V_{model} set to the minimum of 3; c) V_{model} set a bit less than V_{data}; and d) V_{model} much larger than V_{data}

6.4 Application to Virus Data

6.4.1 Virus Data

Next we wanted to investigate whether our model is capable of identifying different views within virus data. We used two different datasets.

Coronavirus dataset We used the same dataset as described in Chapter 5, see Table 5.1. This comprises of the Spike protein sequence for 320 *Betacoronaviruses*, filtered to remove sequences with more than 99.8% amino acid identity. This dataset was chosen to test if the model is able to identify features that are functionally important to virus host specificity.

Arbovirus dataset The Refseq sequences for all the RNA viruses of Metezoa (animals) was downloaded from the Virus Host database on 17/03/2021. A subset of these viruses was selected to create a more balanced dataset with mammal and insect hosts and included all Arboviruses that infect both a mammal and an insect host. The amino acid sequences of all the coding regions for each were concatenated.

An arbovirus is a virus that uses an insect vector to spread between mammalian hosts. This means it must be able to infect both a mammalian and arthropod host. This dataset was selected to test if our model can identify different subsets of features that are associated with infecting insects and mammals. It is expected that the subsets will contain different features since arboviruses are known to use different molecular interactions in the distantly related hosts. We would expect that the arboviruses viruses contain features from both subsets.

6.4.1.1 Virus Labels

We tested alternative ways of labelling the data for the relevant view, testing labels of different ranks of both virus and host taxonomy. The predictive kmer features that are associated with a host are likely to be binary, that is they will be present for one host and absent for all others. Therefore we tested both binary labels, such as Primate/not Primate, and categorical labels, such as the taxonomic order of the host. We generated a

metadata table containing the alternative ways of labelling the viruses for both datasets.

6.4.1.2 Features

As we are interested in any functional signals present in the data we need a to use a genome representation that has the flexibility to capture the functional sites. We generated a feature set derived from the kmer counts of the physio-chemical properties of the amino acid sequences as described previously in Section 4.2.4. We selected a kmer length of five as these features have the highest chance of being of unique in the spike sequence without the matrices become too sparse. This trade-off is described in Section 5.4.

6.4.1.3 Pre-processing the virus data

To make the virus data suitable for the MVC model the count data was pre-processed. First, the kmer count matrices transformed into categorical data and secondly the number of features was reduced by removing features that are likely to be uninformative. MVC is a multinomial Dirichlet mixture model that requires multinomial data over three levels. To transform the counts each column (kmer) was binned into three levels using quantile discretization which aims to assign the same number of observations to each bin. Equal binning is not possible because of the sparsity of the data, the first bin contains all the zero counts which is generally by far the largest number of observations.

To check that this transformation does not reduce the relevant information content contained in the features we compare host prediction of count and discrtetised matrices. We trained and tested a range of classifiers using different host labels to form binary datasets with feature sets both pre- and post-transformation. Figure 6.8 shows that the effect of process of transformation generally improves prediction or at most has a very small negative effect. Binning data can improve accuracy of the predictive models in many ways: reducing non-linearity; reducing the effect of outliers and reducing the degrees of freedom.

To reduce the number of features, we removed those features that occur in less than 5% of the genomes.

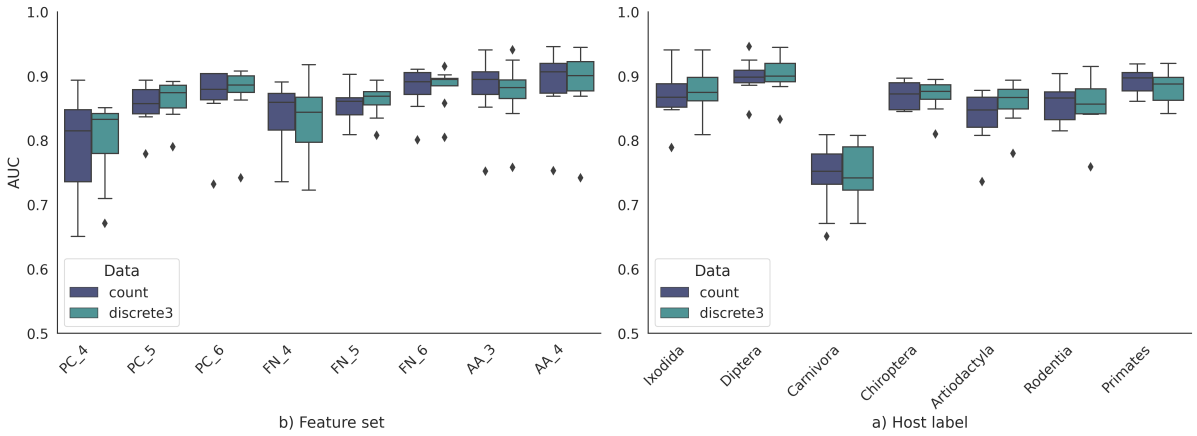


Figure 6.8: The effect on prediction of transforming the count matrices of the features into three discrete levels. a) comparing across feature sets, b) Comparing the different hosts order.

6.4.2 Virus Results

We performed some preliminary experiments to test whether this model would work with virus data. The posterior samples of the model parameters were collected via the Gibbs sampling scheme described in algorithm 2. These samples are used to compute summary probabilities that can help us make inferences about any clustering in the data. The posterior covariate-to-view allocations was used to compute the probability of view membership for each covariate and the co-occurrence of covariates in the same view. In each view, the posterior cluster membership samples is used to compute the co-occurrence of individuals in clusters.

6.4.2.1 MVC can find multiple views in virus data

First we wanted to establish if our model could identify different views within the virus data. Sorted co-occurrence matrices were used to compare the different runs of the model, Figure 6.9 . The covariate-to-view co-occurrence matrix is a covariate by covariate matrix where each element is the proportion of posterior samples in which the two covariates have been assigned to the same view. To make the views more visible the covariates were sorted by their view allocations as described in Section 6.2.4. Figure 6.9 panels a) and b) are the results for coronavirus models using "Primates" as the label for the relevant view and the number of views set to 8 and 3 respectively. Panel c) is the results for the arbovirus dataset. Evidence of different views can be seen in all of the matrices. The

number of views used in the model has a large influence on the number of covariates that are assigned to the relevant view.

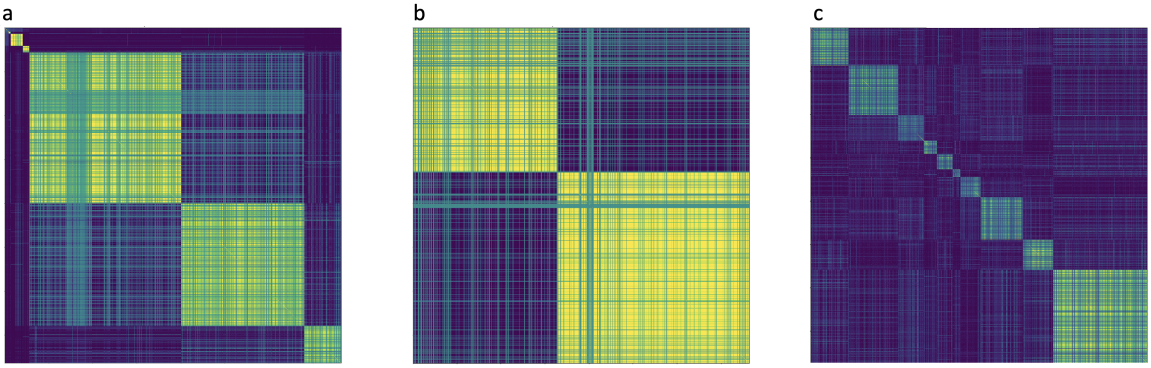


Figure 6.9: The co-occurrence matrix of covariates in views; a) Coronavirus dataset, 3820 covariates, $V_{model} = 8$; b) Coronavirus dataset, 3820 covariates, $V_{model} = 3$; c) Arbovirus dataset, 7743 covariates $V_{model} = 12$. Covariates sorted by the view that they occur in the maximum number of times, (both rows and columns). Yellow indicates that 2 covariates always co-occur in the same view and dark blue indicates that they never co-occur.

To check whether the model was consistently assigning covariates to the relevant view we looked at the distribution of the posterior probabilities of the covariate to view membership, see Section6.2.4. Figure 6.10 shows distribution for the coronavirus models in the relevant view.

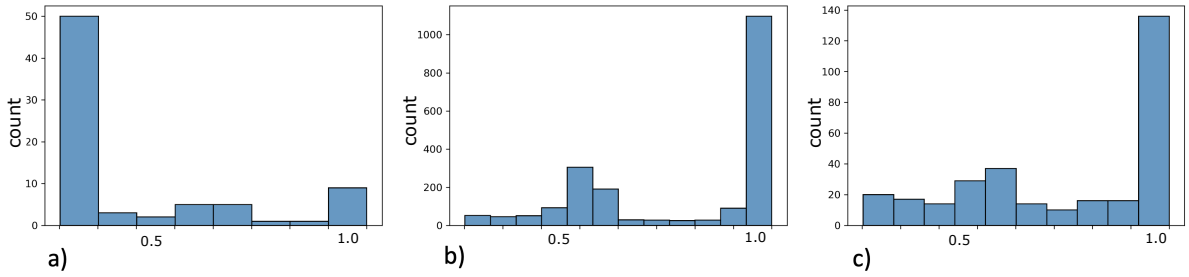


Figure 6.10: Histogram showing the frequency of the probabilities that the covariates occur in the relevant view, a minimum cutoff for the probability = 0.1 to remove the bulk of the covariates.
a) Relevant view = 'Primates', V=8;
b) Relevant view = 'Primates', V=3;
c) Relevant view = 'Chiroptera', V=8 .

For the arbovirus datasets, covariates were assigned to the relevant view in the first 200 of one thousand samples but these dropped out over further iterations. To test whether

this was an issue with the data or with the model, we used the clustering structures in the final samples to test whether the likelihood of each covariate in each view according to the the view clustering in is higher than the likelihood of occurring a clustering defined by the relevant labels. We found that no covariates had a higher likelihood of occurring in the relevant view. This indicates that the level of shared convergence across widely diverse RNA viruses maybe too low and inconsistent to influence the model.

6.4.2.2 Within view individual-to-cluster allocation

Next, we wanted to establish if the model was identifying any biologically meaningful views of the data. To test if the model is finding clusterings that we would expect based on our knowledge of the viruses we created a metadata table with a range of different labeling of the viruses to be used as ground truth. Adjusted Rand Index (Hubert and Arabie 1985), ARI, was used to compare the sampled clustering structures in each view, Z, with this metadata.

A matrix of the ARI values was sampled for each iteration of sampling by computing an ARI comparing all the clusterings in the views to all the metadata clusterings.t labels of the the viruses, Figure 6.11 shows the Hinton plots of the mean ARI across all samples for the coronavirus dataset. Interpreting whether these results indicate that our models are capturing biologically meaningful information is difficult as we have no actual ground truth for which covariates are associated with a host.

6.4.2.3 Comparing the MVC and SVM signals on the SARS-CoV-2 receptor binding site.

The probabilities of the features in the relevant views were used to assign weights to a sequence to identify regions that are important in the virus host specificity in the same way as for SVM in Chapter 5.

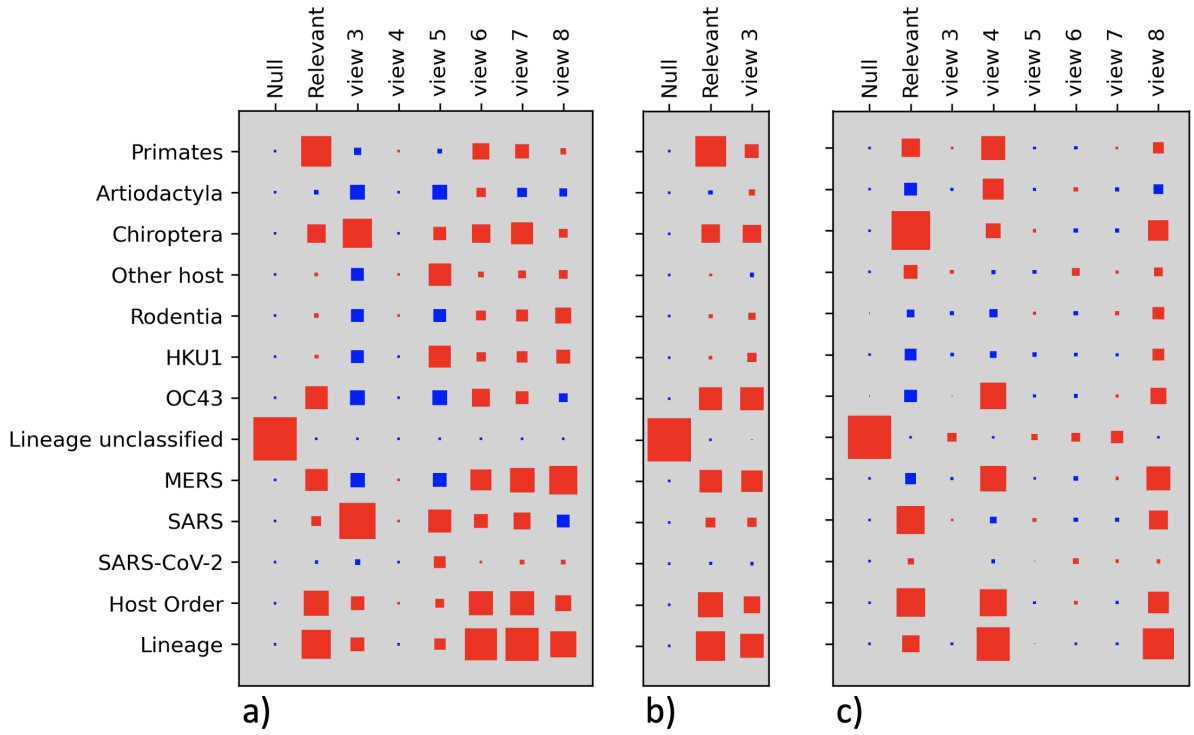


Figure 6.11: Hinton plots of the ARI values comparing the clustering in the views to that of different labels for the Coronavirus dataset. This shows the mean sampled ARI for all views,x axis against the different labels, y axis. The size of each element represents the magnitude of the ARI, red indicates positive values and blue indicates negative values.:
a) Relevant view = 'Primates', V=8, ARI relevant view :'Primates'= 0.47;
b) Relevant view = 'Primates', V=3, ARI relevant view:'Primates'= 0.47;
c) Relevant view = 'Chiroptera', V=8, ARI relevant view:chiroptera = 0.77 .

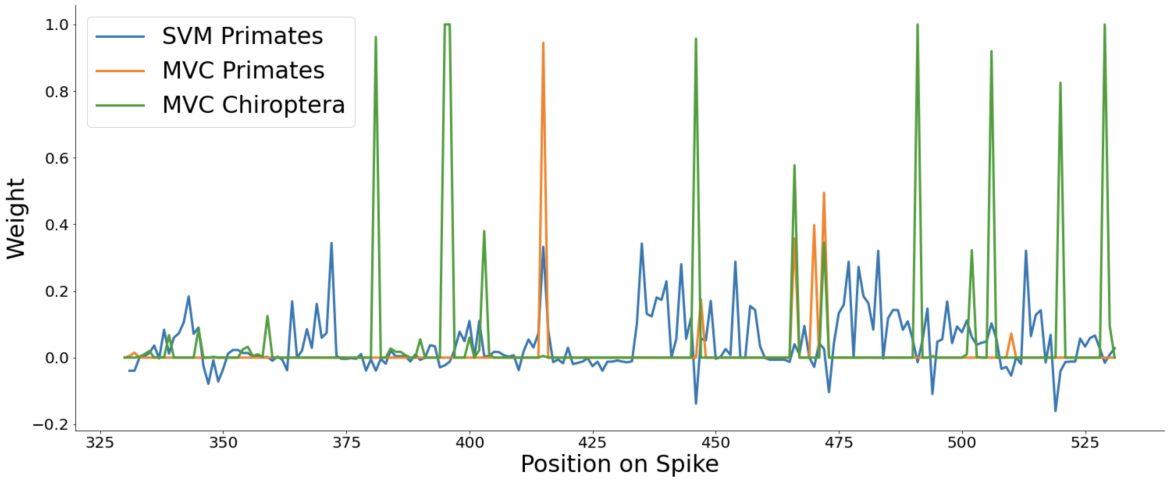


Figure 6.12: Comparing the probability in relevant view from MVC models with relevant labels as the host Primates and Chiroptera to the SVM signals on the receptor binding domain of spike protein of SARS-CoV-2. PC5 feature sets were used for all models.

6.5 Discussion

In this chapter, we introduced MVC, an exploratory method initially developed by Paul Kirk (Kirk and Richardson 2021) to uncover multi-view clustering structures in high dimensional transcriptomic data. Our main aim was to develop and explore the potential of this method to identify covariates that are associated with minor effects in the data. MVC follows the theoretical framework of Bayesian inference with a Gibbs sampler to perform the sampling from the full joint posterior distribution over the model's parameters.

The simulation study demonstrated that the MVC model is able to discover complex multi-view structures in our synthetic data. Simulations 2b and 2c, see Figure 6.6, show that MVC correctly identifies the covariates associated with the relevant view even when this view is a much weaker effect in the data. The model is also able to deal with large numbers of uninformative covariates that are drawn from a single distribution. In Simulation 1c the null view has 168 covariates compared to 81 in the relevant view, see Figure 6.5c.

The results for the virus data are less clear. Although a multi-view structure emerged from the data it is difficult to verify whether these views are biologically meaningful. Essentially, because we do not have the ground truth for which covariates should belong in the relevant view we are unable to fully assess the performance of the model. To overcome this, we used ARI to compare the sampled cluster structure in each view with the clusterings as defined by a range of different metadata about the viruses, including that defined by the labels of the relevant view. The results shown in Figure 6.11 demonstrate that the method is discovering some biologically explainable clustering structures. In particular, the ARI between the relevant view and the labels is consistently the largest value.

There was only time to perform a very preliminary investigation with the virus datasets and only minimal checks for convergence or hyper-parameters optimisation were performed, a more thorough investigation is required. There is a lot of validation work to be done to ascertain whether MVC can work as we would like with our virus datasets. The virus data is also likely to be very noisy due to the high mutation rate of viruses and some of our underlying assumptions may be being challenged.

One of the assumptions that has been made is that the binned representations of the genomes are sufficiently capturing the functional signal. The physio-chemical representation of the virus sequences may not be flexible enough to capture the functional signal we are interested in. We have assumed that viruses converge onto common short linear motifs,

SLiMs. These are linear subsequences, 3-10 amino acids long with only a few positions that play an essential role in the molecular interaction. Our assumption has been that the rest of the sequence is likely also constrained to maintain certain properties such as size or hydrophobicity. Non-conservative substitutions that can occur at positions that are completely unconstrained will result in a change in kmer thereby breaking any predictive signal.

Another assumption has been that there are enough occurrences of convergence across the viruses in the dataset to be 'visible' within the massive total number of features. The question is, do enough viruses converge onto the same host factors and if so, do they use the same motif? In other words, even though it is known that viruses converge onto various host factors in order to subvert their host's systems, different viruses with the same host are likely to converge onto the different sets of host factors using different kmers. This will result in a very complex set of overlapping relationships making the distributions that define the clusters very dispersed and making separating the clusters more difficult. There may not be enough common host factors being converged or enough viruses converging on them to be useful in clustering. The viruses in the arbovirus dataset are very diverse meaning that it is less likely that there will be any convergence identifiable with the PC5 features.

Further is work needed to validate our MVC model. Given the run time was several days to generate 1000 Gibbs samples of the coronavirus model it may be best to do additional testing of the model using smaller synthetic datasets, for example to optimise the choice of the hyperparameters α, β, ϵ on the posterior view and cluster sizes.

One of the difficulties of Gibbs sampling is assessing whether the model has converged, that is, the samples are being taken from the true distribution. Although we ran preliminary checks for convergence and autocorrelation during the simulation study, this needs to be repeated for the virus data. A statistic such as the Geweke Z-score could be used to assess within-chain convergence. Then we should check that different randomly initiated chains have converged on the same distribution, initially by comparing trace plots of an inferred model parameter such as the number of covariates assigned to the relevant view. The samples used for the trace plots can also be used to assess a sensible burn in time and whether thinning needs to be done to compensate for autocorrelation.

Chapter 7

Conclusion

In this thesis I have demonstrate that machine learning models can be used to can exploit the host predictive signals imprinted in viral genomes by their close coevolutionary relationship. The aim of this project was to develop machine learning approaches to find these signals, both to make predictions and to extract information about which regions of a viral protein are important to the virus-host relationship.

In chapter 4, I set out to identify features that are predictive of information about their host. My results demonstrated that by considering how the different types of interaction the virus makes with the host will be reflected in the viral sequences that can captured complementary information that is predictive of host. I developed a holdout method that removed all closely related viruses from the training set and then used to test prediction demonstrating that the SVM algorithm was finding patterns that extended beyond finding those seen in their nearest neighbours. This phylogenetically-aware method indicates that the predictive feature sets contain signals reflecting both the phylogenetic relationship of the viruses and a convergent signal.

In chapter 5, I described an approach to convert the parameters learnt by a machine learning model to a signal on a sequence. This can be used to identify which regions are most strongly associated with the host. I used a DMS data to show that the signal contained biologically relevant information. This approach is model agnostic, as I demonstrated by using it on the results of both an SVM model and a Bayesian mixture model, MVC. Interpreting high dimensional feature space is hard and approaches generally group features together first. The method presented here avoids the need to interpret features without biological context by locating them on a viral protein sequence.

During the analyses above I found that my machine learning models use a complex mix of signals. In Chapter 6 I focused on trying untangle the complex mixture of evolutionary signals embedded in viral genomes to identify a host specific functional signal. I introduce MVC, a Bayesian inference method to tease apart the complex signals in the viral genomes based on work by Paul Kirk (Kirk and Richardson 2021). MVC is a generative mixture model that assumes that the data has been generated from multiple views and uses the labels of the data we are interested in to guide any relevant covariates into the "relevant view". The simulation studies show that MVC is able to discover a complex multi-view structures in high dimensional datasets with large numbers of non-informative covariates. By using a guided approach, it correctly identified the covariates that are associated with a weak effect in the data from much more dominant effects. This application-agnostic clustering method simultaneously groups the individuals and the covariates and can be applied to any data that can be represented as categorical data. During the data pre-processing step I also showed that mapping the kmer count matrices into a more discrete representation (0, 1, 2) didn't hurt classification performance.

In this thesis I focused on finding and locating a host predictive signal in viral genomes and did not go down the route of using these signals to fully optimise a classifier. As discussed in Chapter 4 developing and deploying a host prediction tool will be specific to the prediction problem being tackled, whether that is identifying the origins of a newly emerged pathogen or annotating viruses discovered through metagenomics with putative host taxonomic information. In all cases, using a range of features with multiple-kernel learning would potentially lead to significant improvement in prediction with a means to tune the classifier to optimise the FDR or FPR as required. The classifier performances obtained strongly suggest that this kind of approach could be very useful, and embedding predictive models like this within an easily accessible web app would be a straightforward extension.

Although the simulation studies demonstrate that MVC was able to tease apart the multiple clustering structures in the synthetic data, there is much work to be done to validate and test this model more thoroughly on real data. Preliminary checks were performed for convergence and auto-correlation but this needs to be tested further with the real datasets. To validate MVC as a method suitable to wider problems it should first be tested on a range of benchmark dataset with a known ground truth. A suitable biological dataset of similar complexity would be the results from an RNA-Seq experiment with well established and experimentally corroborated result, such as cancer data which can be validated by The Cancer Genome Atlas.

When applied to the virus data, MVC exposed interesting view structures in the data and highlighted some sites that have recently been identified to be important to SARS-CoV2 infection. The results for the Arbovirus dataset were less conclusive and may imply that my assumption that some mimicked SLiMs are shared across all RNA viruses is incorrect. It maybe because there are too few SLiMs captured by the PC5 features shared across all RNA viruses. In order to reduce the time each run took when testing the MVC algorithm on real data, I used subsets of all the available viruses of both the arbovirus and coronavirus. Future work might include all available viruses, in particular over the last two years many new coronaviruses have been discovered. It would be interesting to test the limits of how much virus diversity can be modelled by using a range carefully constructed datasets.

To build the MVC model, I mapped the features into count data with three levels. Although I showed that this didn't harm classification performance, it is not known whether or not it is the best representation for MVC-like modelling. An obvious comparison would be to test representing the features as binary data, which would perhaps better reflect that what is important in the data is whether a kmer is present or absent in each sequence. This would require only minimal change to the current MVC code as a Beta-binomial model (appropriate for binary data) is a special case of a Dirichlet-multinomial model, which is what is implemented in MVC.

One of the main limitations of this thesis is that I have limited ourselves to using fixed kmers as features to represent sequences. These features are fragile, in that a change in one position of the kmer will result in all information about the similarity being lost. For example, in a simplistic setting where a single kmer is indicative of ability to infect a particular host, viral genomes with a single mutation in this kmer will look completely dissimilar to the models. I demonstrate that my representations of the protein sequences that binned the amino acid residues according to their physio-chemical properties are robust enough to discriminative information for classification. However, ideally we would like to use more flexible features that better capture nature of SLiMs by allowing for some gaps and mismatches enabling them to capture the similarity between highly similar (but not identical) kmers. Given the mimicry of host motif interfaces by viruses and the growth in the number of recognised SLiMs, features that are based on the "SLiM" content of viral proteomes may target a wider range of interaction sites. A recent systematic survey of the earths virome identified over six million instances of interface mimicry and found that viruses belonging to the coronavirus family share a large fraction of mimics with $> 75\%$ of coronaviruses sharing $>75\%$ structural mimics, (Lasso et al. 2021).

An important initial step for any further analysis would be to update the data to include newly discovered viruses. The current worldwide focus on virus surveillance has led to the discovery of many new coronaviruses from a wide variety of hosts. As the understanding about the extent that virus share molecular interfaces we should extend the dataset by including all Coronaviridae (Family), all Nidovirales (Order) or even all RNA viruses. We could test the limits of viral diversity to which machine learning can still find global patterns for prediction.

The SVM was chosen due to its generally high performance, and the fact that (in the linear case) we are able to extract some biological interpretation of the classification function. Another advantage of using a kernel method is that there are other kernels available that have been successfully applied to sequence data and could potentially be combined with those identified in this thesis. The trade off for the improved prediction that these kernels might bring is that as algorithms get more complex they become more difficult to interpret and therefor we loose the possibility of gaining meaningful insights from the data. Other machine learning approaches could also be investigated, in particular methods developed for natural language processing(NLP) are highly transferable to biological sequence data. LSTM, a recurrent neural network model has been successfully applied to virus data both for identification of viral sequences (Miao et al. 2021) in metagenomic data and for predicting host, (Mock et al. 2019). More recently transformer methods have been showing great promise uncovering information from viral genomes. Facebook's Evolutionary Scale Models (ESM) have the advantage that they have been pre-trained on 250 million diverse protein sequences, (Rives et al. 2021).

In conclusion, I have proposed machine learning approaches that identify, extract and interpret host signals from virus genomes. These models can be used to further our understanding of virus-host relationships and virus evolution. I believe that I have demonstrated the power of machine learning to find the weak host signals from the complex mixture of evolutionary signals that make up a virus genome. With more and more powerful methods coming on stream the opportunity to mine the vast wealth of information from huge virus datasets poses an exciting prospect.

Appendix A

MVC: parameters used in simulations

Simulation	Testing	V_{data}	N	J	Jv	K_v	Multinomial parameters β_v	V_{model}
1.a	Large N small P	3	200	50	[13 ,16,21]	[1, 5, 5]	all 0.1	3
1.b	Small N large P	3	80	500	[109, 140, 251]	[1, 5, 5]	all 0.1	3
1.c	Small N large P higher β_{v1}, high J_{v_null}	5	200	500	[166,81,131,53,69]	[1,5,3,5,7]	[0.1,0.5,0.1,0.1,0.1]	5
2.a	$J_1 \sim$Jv $\beta_{v1} > \beta_v$	5	300	500	[82,112,110,82,105]	[1,5,3,5,7]	[0.1,1.0,0.1,0.1,0.1]	5
2.b	$J_{v1} << J_v$ β_{v1} β_v	5	300	500	[120, 12, 170, 92, 106]	[1,5,3,5,7]	[0.1,0.3,0.1,0.1,0.1]	5
2.c	$J_{v1} <<$Jv $\beta_{v1} > \beta_v$	5	300	500	[120, 12, 170, 92, 106]	[1,5,3,5,7]	[0.5,0.8,0.1,0.1,0.1]	5
3.a	$V_m >> V_d$	12	400	1000	[47,121,98,91,48,87, 50,139,108,34, 133,44]	[1,5,3,5,7, 9,8,3,5,5, 3,5,]	[0.5,0.8,0.5,0.5,0.1, 0.1,0.8,0.1,0.1,0.1, 0.5,0.1,0.2,0.2]	20
3.b	$V_m < V_d$	12	400	1000	[47,121,98,91,48,87, 50,139,108,34, 133,44]	[1,5,3,5,7, 9,8,3,5,5, 3,5,]	[0.5,0.8,0.5,0.5,0.1, 0.1,0.8,0.1,0.1,0.1, 0.5,0.1,0.2,0.2]	8
3.c	$V_m << V_d$	12	400	1000	[47,121,98,91,48,87, 50,139,108,34, 133,44]	[1,5,3,5,7, 9,8,3,5,5, 3,5,]	[0.5,0.8,0.5,0.5,0.1, 0.1,0.8,0.1,0.1,0.1, 0.5,0.1,0.2,0.2]	3

Table A.1: Parameters used to generate synthetic datasets

Bibliography

Ahlgren, Nathan A. et al. (Jan. 2017). "Alignment-free d2* oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences". en. In: *Nucleic Acids Research* 45.1, pp. 39–53. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1002. URL: https://academic.oup.com/nar/article/45/1/39/2605663.

Aiewsakun, Pakorn and Peter Simmonds (Feb. 2018). "The genomic underpinnings of eukaryotic virus taxonomy: creating a sequence-based framework for family-level virus classification". In: *Microbiome* 6. ISSN: 2049-2618. DOI: 10.1186/s40168-018-0422-7. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5819261/.

Aiewsakun, Pakorn et al. (July 2018). "Evaluation of the genomic diversity of viruses infecting bacteria, archaea and eukaryotes using a common bioinformatic platform: steps towards a unified taxonomy". en. In: *Journal of General Virology*. ISSN: 0022-1317, 1465-2099. DOI: 10.1099/jgv.0.001110. URL: http://www.microbiologyresearch.org/content/journal/jgv/10.1099/jgv.0.001110.v1.

Apic, Gordana and Robert B. Russell (Sept. 2010). "Domain Recombination: A Workhorse for Evolutionary Innovation". EN. In: *Science Signaling*. Publisher: American Association for the Advancement of Science. DOI: 10.1126/scisignal.3139pe30. URL: https://www.science.org/doi/abs/10.1126/scisignal.3139pe30.

Argelaguet, Ricard (June 2018). "Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets". In: *Molecular Systems Biology* 14.6. Publisher: John Wiley & Sons, Ltd, e8124. ISSN: 1744-4292. DOI: 10.15252/msb.20178124. URL: https://www.embopress.org/doi/full/10.15252/msb.20178124.

Atkinson, Nicky J. et al. (Apr. 2014). "The influence of CpG and UpA dinucleotide frequencies on RNA virus replication and characterization of the innate cellular pathways underlying virus attenuation and enhanced replication". In: *Nucleic Acids Research* 42.7, pp. 4527–4545. ISSN: 0305-1048. DOI: 10.1093/nar/gku075. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3985648/.

Babayan, Simon A., Richard J. Orton, and Daniel G. Streicker (Nov. 2018). "Predicting reservoir hosts and arthropod vectors from evolutionary signatures in RNA virus genomes". en. In: *Science* 362.6414, pp. 577–580. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aap9072. URL: http://science.sciencemag.org/content/362/6414/577.

Baltimore, D (Sept. 1971). "Expression of animal virus genomes." In: *Bacteriological Reviews* 35.3, pp. 235–241. ISSN: 0005-3678. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC378387/.

Barabási, Albert-László (July 2009). "Scale-Free Networks: A Decade and Beyond". en. In: *Science* 325.5939, pp. 412–413. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1173299. URL: http://science.sciencemag.org/content/325/5939/412.

Barman, Ranjan Kumar, Sudipto Saha, and Santasabuj Das (Nov. 2014). "Prediction of Interactions between Viral and Host Proteins Using Supervised Machine Learning Methods". In: *PLoS ONE* 9.11. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0112034. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4223108/.

Beijerinck, M.W (1898). "On a Contagium vivum fluidum causing the Spotdisease of the Tobacco-leaves." In: *Koninklijke Nederlandse Akademie van Wetenschappen Proceedings Series B Physical Sciences* 1, pp.170–176.

Ben-Hur, Asa et al. (Oct. 2008). "Support Vector Machines and Kernels for Computational Biology". en. In: *PLoS Computational Biology* 4.10. Ed. by Fran Lewitter. tex.ids= benhurSupportVectorMachines2008a publisher: Public Library of Science, e1000173. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000173. URL: https://dx.plos.org/10.1371/journal.pcbi.1000173.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0-387-31073-8.

Boni, Maciej F. et al. (Nov. 2020). "Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic". en. In: *Nature Microbiology* 5.11. Number: 11 Publisher: Nature Publishing Group, pp. 1408–1417. ISSN: 2058-5276. DOI: 10.1038/s41564-020-0771-4. URL: https://www.nature.com/articles/s41564-020-0771-4.

Brister, J. Rodney et al. (Jan. 2015). "NCBI Viral Genomes Resource". In: *Nucleic Acids Research* 43.Database issue, pp. D571–D577. ISSN: 0305-1048. DOI: 10.1093/nar/gku1207. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4383986/.

Brito, Anderson F. and John W. Pinney (Aug. 2017). "Protein–Protein Interactions in Virus–Host Systems". In: *Frontiers in Microbiology* 8. ISSN: 1664-302X. DOI: 10.3389/fmicb.2017.01557. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5562681/.

Brynildsrud, Ola et al. (2016). "Rapid scoring of genes in microbial pan-genome-wide association studies with Scoary". eng. In: *Genome Biology* 17.1, p. 238. ISSN: 1474-760X. DOI: 10.1186/s13059-016-1108-8.

Carbone, Alessandra (Mar. 2008). "Codon Bias is a Major Factor Explaining Phage Evolution in Translationally Biased Hosts". en. In: *Journal of Molecular Evolution* 66.3, pp. 210–223. ISSN: 0022-2844, 1432-1432. DOI: 10.1007/s00239-008-9068-6. URL: https://link.springer.com/article/10.1007/s00239-008-9068-6.

Cervantes, Jair et al. (Sept. 2020). "A comprehensive survey on support vector machine classification: Applications, challenges and trends". en. In: *Neurocomputing* 408, pp. 189–215. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.10.118. URL: https://www.sciencedirect.com/science/article/pii/S0925231220307153.

Chemes, Lucía Beatriz, Gonzalo de Prat-Gay, and Ignacio Enrique Sánchez (June 2015). "Convergent evolution and mimicry of protein linear motifs in host–pathogen interactions". en. In: *Current Opinion in Structural Biology*. New constructs and expression of proteins / Sequences and topology 32, pp. 91–101. ISSN: 0959-440X. DOI: 10.1016/j.sbi.2015.03.004. URL: https://www.sciencedirect.com/science/article/pii/S0959440X15000317.

Coclet, Clément and Simon Roux (Aug. 2021). "Global overview and major challenges of host prediction methods for uncultivated phages". en. In: *Current Opinion in Virology* 49, pp. 117–126. ISSN: 1879-6257. DOI: 10.1016/j.coviro.2021.05.003. URL: https://www.sciencedirect.com/science/article/pii/S1879625721000572.

Cortes, Corinna and Vladimir Vapnik (Sept. 1995). "Support-vector networks". en. In: *Machine Learning* 20.3, pp. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00994018. URL: http://link.springer.com/10.1007/BF00994018.

Daugherty, Matthew D. and Harmit S. Malik (2012). "Rules of Engagement: Molecular Insights from Host-Virus Arms Races". In: *Annual Review of Genetics* 46.1, pp. 677–700. DOI: 10.1146/annurev-genet-110711-155522. URL: https://doi.org/10.1146/annurev-genet-110711-155522.

Davey, Norman E., Martha S. Cyert, and Alan M. Moses (Nov. 2015). "Short linear motifs – ex nihilo evolution of protein regulation". In: *Cell Communication and Signaling* 13.1, p. 43. ISSN: 1478-811X. DOI: 10.1186/s12964-015-0120-z. URL: https://doi.org/10.1186/s12964-015-0120-z.

Davey, Norman E., Gilles Travé, and Toby J. Gibson (Mar. 2011). "How viruses hijack cell regulation". en. In: *Trends in Biochemical Sciences* 36.3, pp. 159–169. ISSN: 09680004. DOI: 10.1016/j.tibs.2010.10.002. URL: http://linkinghub.elsevier.com/retrieve/pii/S0968000410002008.

Di Giallonardo, Francesca et al. (Mar. 2017). "Dinucleotide Composition in Animal RNA Viruses Is Shaped More by Virus Family than by Host Species". In: *Journal of Virology* 91.8. ISSN: 0022-538X. DOI: 10.1128/JVI.02381-16. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5375695/.

Dutilh, Bas E. et al. (July 2014). "A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes". En. In: *Nature Communications* 5, p. 4498. ISSN: 2041-1723. DOI: 10.1038/ncomms5498. URL: https://www.nature.com/articles/ncomms5498.

Eddy, Sean R. (Oct. 2011). "Accelerated Profile HMM Searches". en. In: *PLOS Computational Biology* 7.10, e1002195. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002195. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002195.

Edwards, Robert A. and Forest Rohwer (May 2005). "Viral metagenomics". en. In: *Nature Reviews Microbiology* 3.6, nrmicro1163. ISSN: 1740-1534. DOI: 10.1038/nrmicro1163. URL: https://www.nature.com/articles/nrmicro1163.

Edwards, Robert A. et al. (Mar. 2016). "Computational approaches to predict bacteriophage–host relationships". In: *FEMS Microbiology Reviews* 40.2, pp. 258–272. ISSN: 0168-6445. DOI: 10.1093/femsre/fuv048. URL: https://academic.oup.com/femsre/article/40/2/258/2570202/Computational-approaches-to-predict-bacteriophage.

Elde, Nels C. and Harmit S. Malik (Oct. 2009). "The evolutionary conundrum of pathogen mimicry". en. In: *Nature Reviews Microbiology* 7.11, nrmicro2222. ISSN: 1740-1534. DOI: 10.1038/nrmicro2222. URL: https://www.nature.com/articles/nrmicro2222.

Falkowski, Paul G., Tom Fenchel, and Edward F. Delong (May 2008). "The Microbial Engines That Drive Earth's Biogeochemical Cycles". en. In: *Science* 320.5879, pp. 1034–1039. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1153213. URL: http://science.sciencemag.org/content/320/5879/1034.

Finn, Robert D., Jody Clements, and Sean R. Eddy (July 2011). "HMMER web server: interactive sequence similarity searching". en. In: *Nucleic Acids Research* 39.suppl_2, W29–W37. ISSN: 0305-1048. DOI: 10.1093/nar/gkr367. URL: https://academic.oup.com/nar/article/39/suppl_2/W29/2506513.

Forni, Diego et al. (Jan. 2017). "Molecular Evolution of Human Coronavirus Genomes". en. In: *Trends in Microbiology* 25.1, pp. 35–48. ISSN: 0966-842X. DOI: 10.1016/j.tim.2016.09.001. URL: https://www.sciencedirect.com/science/article/pii/S0966842X16301330.

Forterre, Patrick (Apr. 2011). "Manipulation of cellular syntheses and the nature of viruses: The virocell concept". In: *Comptes Rendus Chimie*. De la chimie de synthèse à la biologie de synthèse 14.4. tex.ids= forterreManipulationCellularSyntheses2011a, pp. 392–399.

ISSN: 1631-0748. DOI: 10.1016/j.crci.2010.06.007. URL: http://www.sciencedirect.com/science/article/pii/S1631074810001724.

Franzosa, Eric A. and Yu Xia (June 2011). "Structural principles within the human-virus protein-protein interaction network". en. In: *Proceedings of the National Academy of Sciences* 108.26, pp. 10538–10543. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1101440108. URL: http://www.pnas.org/content/108/26/10538.

Gałan, Wojciech, Maciej Bąk, and Małgorzata Jakubowska (Mar. 2019). "Host Taxon Predictor - A Tool for Predicting Taxon of the Host of a Newly Discovered Virus". En. In: *Scientific Reports* 9.1, p. 3436. ISSN: 2045-2322. DOI: 10.1038/s41598-019-39847-2. URL: https://www.nature.com/articles/s41598-019-39847-2.

Galiez, Clovis et al. (Oct. 2017). "WIsH: who is the host? Predicting prokaryotic hosts from metagenomic phage contigs". en. In: *Bioinformatics* 33.19, pp. 3113–3114. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx383. URL: https://academic.oup.com/bioinformatics/article/33/19/3113/3964377.

Gao, Na L. et al. (Jan. 2018). "MVP: a microbe–phage interaction database". en. In: *Nucleic Acids Research* 46.D1, pp. D700–D707. ISSN: 0305-1048. DOI: 10.1093/nar/gkx1124. URL: https://academic.oup.com/nar/article/46/D1/D700/4643372.

Garamszegi, Sara, Eric A. Franzosa, and Yu Xia (Dec. 2013). "Signatures of Pleiotropy, Economy and Convergent Evolution in a Domain-Resolved Map of Human–Virus Protein–Protein Interaction Networks". en. In: *PLOS Pathogens* 9.12, e1003778. ISSN: 1553-7374. DOI: 10.1371/journal.ppat.1003778. URL: https://journals.plos.org/plospathogens/article?id=10.1371/journal.ppat.1003778.

Gelfand, Alan E. and Adrian F. M. Smith (June 1990). "Sampling-Based Approaches to Calculating Marginal Densities". In: *Journal of the American Statistical Association* 85.410. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1 pp. 398–409. ISSN: 0162-1459. DOI: 10.1080/01621459.1990.10476213. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10476213.

Geoghegan, Jemma L., Sebastián Duchêne, and Edward C. Holmes (Feb. 2017). "Comparative analysis estimates the relative frequencies of co-divergence and cross-species transmission within viral families". In: *PLoS Pathogens* 13.2. ISSN: 1553-7366. DOI: 10.1371/journal.ppat.1006215. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5319820/.

Gifford, Robert and Michael Tristem (May 2003). "The Evolution, Distribution and Diversity of Endogenous Retroviruses". en. In: *Virus Genes* 26.3, pp. 291–315. ISSN: 1572-994X. DOI: 10.1023/A:1024455415443. URL: https://doi.org/10.1023/A:1024455415443.

Gilpin, Leilani H. et al. (Oct. 2018). "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89. DOI: 10.1109/DSAA.2018.00018.

Goff, Stephen P. (Sept. 2017). "Evolution: Zapping viral RNAs". en. In: *Nature* 550.7674, nature24140. ISSN: 1476-4687. DOI: 10.1038/nature24140. URL: https://www.nature.com/articles/nature24140.

Gough, Julian (Apr. 2005). "Convergent evolution of domain architectures (is rare)". eng. In: *Bioinformatics (Oxford, England)* 21.8, pp. 1464–1471. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti204.

Greenbaum, Benjamin D. et al. (June 2008). "Patterns of Evolution and Host Gene Mimicry in Influenza and Other RNA Viruses". en. In: *PLOS Pathogens* 4.6, e1000079. ISSN: 1553-7374. DOI: 10.1371/journal.ppat.1000079. URL: https://journals.plos.org/plospathogens/article?id=10.1371/journal.ppat.1000079.

Guarner, Jeannette (Mar. 2020). "Three Emerging Coronaviruses in Two Decades: The Story of SARS, MERS, and Now COVID-19". In: *American Journal of Clinical Pathology* 153.4, pp. 420–421. ISSN: 0002-9173. DOI: 10.1093/ajcp/aqaa029. URL: https://doi.org/10.1093/ajcp/aqaa029.

Guven-Maiorov, Emine, Chung-Jung Tsai, and Ruth Nussinov (Oct. 2016). "Pathogen mimicry of host protein-protein interfaces modulates immunity". In: *Seminars in Cell & Developmental Biology*. Cardiac Regeneration 58. tex.ids= guven-maiorovPathogenMimicryHost2016a, pp. 136–145. ISSN: 1084-9521. DOI: 10.1016/j.semcdb.2016.06.004. URL: http://www.sciencedirect.com/science/article/pii/S1084952116301665.

Hagai, Tzachi et al. (June 2014). "Use of Host-like Peptide Motifs in Viral Proteins Is a Prevalent Strategy in Host-Virus Interactions". In: *Cell Reports* 7.5, pp. 1729–1739. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2014.04.052. URL: http://www.sciencedirect.com/science/article/pii/S2211124714003702.

Harris, Hugh M. B. and Colin Hill (2021). "A Place for Viruses on the Tree of Life". English. In: *Frontiers in Microbiology* 11. Publisher: Frontiers. ISSN: 1664-302X. DOI: 10.3389/fmicb.2020.604048. URL: https://www.frontiersin.org/articles/10.3389/fmicb.2020.604048/full.

Harrison, Joshua G. et al. (Dec. 2019). "Dirichlet-multinomial modelling outperforms alternatives for analysis of microbiome and other ecological count data". en. In: *bioRxiv*. Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 711317. DOI: 10.1101/711317. URL: https://www.biorxiv.org/content/10.1101/711317v3.

– (2020). "Dirichlet-multinomial modelling outperforms alternatives for analysis of microbiome and other ecological count data". en. In: *Molecular Ecology Resources* 20.2.

_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1755-0998.13128 tex.ids= harrisonDirichletmultinomialModellingOutperforms2020a, pp. 481–497. ISSN: 1755-0998. DOI: 10.1111/1755-0998.13128. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1755-0998.13128.

Hembry, David H., Jeremy B. Yoder, and Kari Roesch Goodman (Oct. 2014). "Coevolution and the Diversification of Life." In: *The American Naturalist* 184.4. tex.ids= hembryCoevolutionDiversificationLife2014a publisher: The University of Chicago Press, pp. 425–438. ISSN: 0003-0147. DOI: 10.1086/677928. URL: https://www.journals.uchicago.edu/doi/full/10.1086/677928.

Hubert, Lawrence and Phipps Arabie (Dec. 1985). "Comparing partitions". en. In: *Journal of Classification* 2.1, pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075. URL: https://doi.org/10.1007/BF01908075.

Huerta-Cepas, Jaime, François Serra, and Peer Bork (June 2016). "ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data". en. In: *Molecular Biology and Evolution* 33.6, pp. 1635–1638. ISSN: 0737-4038. DOI: 10.1093/molbev/msw046. URL: https://academic.oup.com/mbe/article/33/6/1635/2579822.

Jain, Chirag et al. (Nov. 2018). "High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries". en. In: *Nature Communications* 9.1, pp. 1–8. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07641-9. URL: https://www.nature.com/articles/s41467-018-07641-9.

Jenkins, Gareth M and Edward C Holmes (Mar. 2003). "The extent of codon usage bias in human RNA viruses and its evolutionary origin". In: *Virus Research* 92.1, pp. 1–7. ISSN: 0168-1702. DOI: 10.1016/S0168-1702(02)00309-X. URL: http://www.sciencedirect.com/science/article/pii/S016817020200309X.

Jiang, Lingjing et al. (Nov. 2020). "Utilizing stability criteria in choosing feature selection methods yields reproducible results in microbiome data". en. In: *arXiv:2012.00001 [cs, q-bio]*. arXiv: 2012.00001. URL: http://arxiv.org/abs/2012.00001.

Jonge, Patrick A. de et al. (Sept. 2018). "Molecular and Evolutionary Determinants of Bacteriophage Host Range". In: *Trends in Microbiology*. ISSN: 0966-842X. DOI: 10.1016/j.tim.2018.08.006. URL: http://www.sciencedirect.com/science/article/pii/S0966842X18301781.

Kapoor, A. et al. (Oct. 2010). "Use of Nucleotide Composition Analysis To Infer Hosts for Three Novel Picorna-Like Viruses". In: *Journal of Virology* 84.19, pp. 10322–10328. ISSN: 0022-538X. DOI: 10.1128/JVI.00601-10. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2937767/.

Katoh, Kazutaka et al. (July 2002). "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". In: *Nucleic Acids Research* 30.14, pp. 3059–3066. ISSN: 0305-1048. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC135756/.

King, Cason R. et al. (May 2018). "Hacking the Cell: Network Intrusion and Exploitation by Adenovirus E1A". In: *mBio* 9.3. tex.ids= kingHackingCellNetwork2018a publisher: American Society for Microbiology. ISSN: 2150-7511. DOI: 10.1128/mBio.00390-18. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5930299/.

Kirk, P. D. W. and S Richardson (2021). "Semi-supervised multi-view Bayesian clustering." In: *In preparation.*

Koonin, Eugene V., Valerian V. Dolja, and Mart Krupovic (May 2015). "Origins and evolution of viruses of eukaryotes: The ultimate modularity". In: *Virology.* 60th Anniversary Issue 479-480.Supplement C. tex.ids= kooninOriginsEvolutionViruses2015a, pp. 2–25. ISSN: 0042-6822. DOI: 10.1016/j.virol.2015.02.039. URL: http://www.sciencedirect.com/science/article/pii/S0042682215000859.

Koonin, Eugene V. and Petro Starokadomskyy (Oct. 2016). "Are viruses alive? The replicator paradigm sheds decisive light on an old but misguided question". In: *Studies in history and philosophy of biological and biomedical sciences* 59. tex.ids= kooninAreVirusesAlive2016a, pp. 125–134. ISSN: 1369-8486. DOI: 10.1016/j.shpsc.2016.02.016. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5406846/.

Koskella, Britt and Michael A Brockhurst (Sept. 2014). "Bacteria–phage coevolution as a driver of ecological and evolutionary processes in microbial communities". In: *Fems Microbiology Reviews* 38.5, pp. 916–931. ISSN: 0168-6445. DOI: 10.1111/1574-6976.12072. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4257071/.

Lanckriet, Gert R G, Nello Cristianini, and Peter Bartlett (2004). "Learning the Kernel Matrix with Semidefinite Programming". en. In: p. 46.

Lasso, Gorka, Barry Honig, and Sagi D. Shapira (Jan. 2021). "A Sweep of Earth's Virome Reveals Host-Guided Viral Protein Structural Mimicry and Points to Determinants of Human Disease". en. In: *Cell Systems* 12.1, 82–91.e3. ISSN: 2405-4712. DOI: 10.1016/j.cels.2020.09.006. URL: https://www.sciencedirect.com/science/article/pii/S240547122030363X.

Lavialle, Christian et al. (Sept. 2013). "Paleovirology of 'syncytins', retroviral env genes exapted for a role in placentation". eng. In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 368.1626, p. 20120507. ISSN: 1471-2970. DOI: 10.1098/rstb.2012.0507.

Leite, Diogo Manuel Carvalho et al. (Nov. 2018). "Computational prediction of interspecies relationships through omics data analysis and machine learning". In: *BMC Bioin-*

formatics 19.Suppl 14. ISSN: 1471-2105. DOI: 10.1186/s12859-018-2388-7. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6245486/.

Leung, M. K. K. et al. (Jan. 2016). "Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets". In: *Proceedings of the IEEE* 104.1, pp. 176–197. ISSN: 0018-9219. DOI: 10.1109/JPROC.2015.2494198.

Li, Han and Fengzhu Sun (July 2018). "Comparative studies of alignment, alignment-free and SVM based approaches for predicting the hosts of viruses based on viral sequences". En. In: *Scientific Reports* 8.1, p. 10032. ISSN: 2045-2322. DOI: 10.1038/s41598-018-28308-x. URL: https://www.nature.com/articles/s41598-018-28308-x.

Libbrecht, Maxwell W. and William Stafford Noble (June 2015). "Machine learning applications in genetics and genomics". en. In: *Nature Reviews Genetics* 16.6. tex.ids= libbrechtMachineLearningApplications2015a number: 6 publisher: Nature Publishing Group, pp. 321–332. ISSN: 1471-0056. DOI: 10.1038/nrg3920. URL: http://www.nature.com/nrg/journal/v16/n6/full/nrg3920.html.

Liu, Lin et al. (Sept. 2016). "An overview of topic modeling and its current applications in bioinformatics". In: *SpringerPlus* 5.1. ISSN: 2193-1801. DOI: 10.1186/s40064-016-3252-8. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5028368/.

Liu, Siyu, Chuyao Liu, and Lei Deng (Oct. 2018). "Machine Learning Approaches for Protein–Protein Interaction Hot Spot Prediction: Progress and Comparative Assessment". en. In: *Molecules* 23.10. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 2535. ISSN: 1420-3049. DOI: 10.3390/molecules23102535. URL: https://www.mdpi.com/1420-3049/23/10/2535.

Lovell, Simon C. and David L. Robertson (Nov. 2010). "An Integrated View of Molecular Coevolution in Protein–Protein Interactions". In: *Molecular Biology and Evolution* 27.11, pp. 2567–2575. ISSN: 0737-4038. DOI: 10.1093/molbev/msq144. URL: https://doi.org/10.1093/molbev/msq144.

McLaughlin, Richard N. and Harmit S. Malik (Jan. 2017). "Genetic conflicts: the usual suspects and beyond". In: *The Journal of Experimental Biology* 220.1, pp. 6–17. ISSN: 0022-0949. DOI: 10.1242/jeb.148148. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5278622/.

Meier-Kolthoff, Jan P and Markus Göker (Nov. 2017). "VICTOR: genome-based phylogeny and classification of prokaryotic viruses". In: *Bioinformatics* 33.21, pp. 3396–3404. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx440. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5860169/.

Meyerson, Nicholas R. and Sara L. Sawyer (June 2011). "Two-stepping through time: mammals and viruses". English. In: *Trends in Microbiology* 19.6. Publisher: Elsevier,

pp. 286–294. ISSN: 0966-842X, 1878-4380. DOI: 10.1016/j.tim.2011.03.006. URL: https://www.cell.com/trends/microbiology/abstract/S0966-842X(11)00054-0.

Miao, Yan et al. (Dec. 2021). "Virtifier: a deep learning-based identifier for viral sequences from metagenomes". In: *Bioinformatics*. tex.ids= miaoVirtifierDeepLearning-based2021a, miaoVirtifierDeepLearningbased2021b, btab845. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab845. URL: https://doi.org/10.1093/bioinformatics/btab845.

Mihara, Tomoko et al. (Mar. 2016). "Linking Virus Genomes with Host Taxonomy". In: *Viruses* 8.3. ISSN: 1999-4915. DOI: 10.3390/v8030066. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4810256/.

Mock, Florian et al. (Mar. 2019). "Viral host prediction with Deep Learning". en. In: *bioRxiv*. DOI: 10.1101/575571. URL: http://biorxiv.org/lookup/doi/10.1101/575571.

Molitor, John et al. (July 2010). "Bayesian profile regression with an application to the National survey of children's health". In: *Biostatistics* 11.3, pp. 484–498. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxq013. URL: https://doi.org/10.1093/biostatistics/kxq013.

Molnar, Christoph (2022). *Interpretable Machine Learning*. URL: https://christophm.github.io/interpretable-ml-book/.

Murphy, Kevin P. (2012). *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. Cambridge, MA: MIT Press. ISBN: 978-0-262-01802-9.

Nasir, Arshan and Gustavo Caetano-Anollés (Sept. 2015). "A phylogenomic data-driven exploration of viral origins and evolution". In: *Science Advances* 1.8. tex.ids= nasirPhylogenomicDatadrivenExploration2015a. ISSN: 2375-2548. DOI: 10.1126/sciadv.1500527. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4643759/.

Neal, Radford M. (2000). "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2. Publisher: [American Statistical Association, Taylor & Francis, Ltd., Institute of Mathematical Statistics, Interface Foundation of America], pp. 249–265. ISSN: 1061-8600. DOI: 10.2307/1390653. URL: https://www.jstor.org/stable/1390653.

Nogueira, Sarah, Konstantinos Sechidis, and Gavin Brown (2018). "On the Stability of Feature Selection Algorithms". en. In: p. 54.

Oyeyemi, Oyebode J. et al. (Apr. 2015). "A logical model of HIV-1 interactions with the T-cell activation signalling pathway". In: *Bioinformatics* 31.7, pp. 1075–1083. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu787. URL: https://doi.org/10.1093/bioinformatics/btu787.

Paez-Espino, David et al. (Jan. 2019). "IMG/VR v.2.0: an integrated data management and analysis system for cultivated and environmental viral genomes". en. In: *Nucleic*

Acids Research 47.D1, pp. D678–D686. ISSN: 0305-1048. DOI: 10.1093/nar/gky1127. URL: https://academic.oup.com/nar/article/47/D1/D678/5165269.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine Learning in Python". en. In: *MACHINE LEARNING IN PYTHON* 12, p. 6.

Phan, My V T et al. (Dec. 2018). "Identification and characterization of Coronaviridae genomes from Vietnamese bats and rats based on conserved protein domains". In: *Virus Evolution* 4.2. ISSN: 2057-1577. DOI: 10.1093/ve/vey035. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6295324/.

Pope, Welkin H et al. (Apr. 2015). "Whole genome comparison of a large collection of mycobacteriophages reveals a continuum of phage genetic diversity". In: *eLife* 4. Ed. by Roberto Kolter, e06416. ISSN: 2050-084X. DOI: 10.7554/eLife.06416. URL: https://doi.org/10.7554/eLife.06416.

Pournara, Iosifina and Lorenz Wernisch (Feb. 2007). "Factor analysis for gene regulatory networks and transcription factor activity profiles". In: *BMC Bioinformatics* 8, p. 61. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-61. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1821042/.

Price, Morgan N., Paramvir S. Dehal, and Adam P. Arkin (Mar. 2010). "FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments". en. In: *PLOS ONE* 5.3. Publisher: Public Library of Science, e9490. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0009490. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0009490.

Puigbò, Pere, Yuri I. Wolf, and Eugene V. Koonin (July 2009). "Search for a 'Tree of Life' in the thicket of the phylogenetic forest". en. In: *Journal of Biology* 8.6, p. 59. ISSN: 1475-4924. DOI: 10.1186/jbiol159. URL: https://doi.org/10.1186/jbiol159.

Raj, Anil et al. (Dec. 2011). "Identifying Hosts of Families of Viruses: A Machine Learning Approach". In: *PLoS ONE* 6.12. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0027631. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3235098/.

Raoult, Didier and Patrick Forterre (Apr. 2008). "Redefining viruses: lessons from Mimivirus". en. In: *Nature Reviews Microbiology* 6.4. Number: 4 Publisher: Nature Publishing Group, pp. 315–319. ISSN: 1740-1534. DOI: 10.1038/nrmicro1858. URL: https://www.nature.com/articles/nrmicro1858.

Rasmussen, Carl (1999). "The Infinite Gaussian Mixture Model". In: *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press. URL: https://papers.nips.cc/paper/1999/hash/97d98119037c5b8a9663cb21fb8ebf47-Abstract.html.

Remita, Mohamed Amine et al. (Apr. 2017). "A machine learning approach for viral genome classification". In: *BMC Bioinformatics* 18. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1602-3. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5387389/.

Rives, Alexander et al. (Apr. 2021). "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". en. In: *Proceedings of the National Academy of Sciences* 118.15. tex.ids= rivesBiologicalStructureFunction2021a, rivesBiologicalStructureFunction2021b, rivesBiologicalStructureFunction2021c publisher: National Academy of Sciences section: Biological Sciences. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2016239118. URL: https://www.pnas.org/content/118/15/e2016239118.

Roossinck, Marilyn J. and Edelio R. Bazán (Sept. 2017). "Symbiosis: Viruses as Intimate Partners". en. In: *Annual Review of Virology* 4.1, pp. 123–139. ISSN: 2327-056X, 2327-0578. DOI: 10.1146/annurev-virology-110615-042323. URL: https://www.annualreviews.org/doi/10.1146/annurev-virology-110615-042323.

Roux, Simon et al. (July 2015). "Viral dark matter and virus–host interactions resolved from publicly available microbial genomes". In: *eLife* 4. ISSN: 2050-084X. DOI: 10.7554/eLife.08490. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4533152/.

Roux, Simon et al. (2019). "Minimum Information about an Uncultivated Virus Genome (MIUViG)". In: *Nature Biotechnology* 37.1, pp. 29–37. ISSN: 1087-0156. DOI: 10.1038/nbt.4306. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6871006/.

Roux, Simon et al. (Jan. 2021). "IMG/VR v3: an integrated ecological and evolutionary framework for interrogating genomes of uncultivated viruses". In: *Nucleic Acids Research* 49.D1, pp. D764–D775. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa946. URL: https://doi.org/10.1093/nar/gkaa946.

Shen, Juwen et al. (Mar. 2007). "Predicting protein–protein interactions based only on sequences information". In: *Proceedings of the National Academy of Sciences of the United States of America* 104.11, pp. 4337–4341. ISSN: 0027-8424. DOI: 10.1073/pnas.0607879104. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1838603/.

Silva, José Cleydson F. et al. (Sept. 2017). "Fangorn Forest (F2): a machine learning approach to classify genes and genera in the family Geminiviridae". In: *BMC Bioinformatics* 18. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1839-x. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5622471/.

Simmonds, Peter, Pakorn Aiewsakun, and Aris Katzourakis (May 2019). "Prisoners of war — host adaptation and its constraints on virus evolution". en. In: *Nature Reviews Microbiology* 17.5. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 5 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Coevo-

lution;Evolutionary theory;Viral evolution;Virus–host interactions Subject_term_id: coevolution;evolutionary-theory;viral-evolution;virus-host-interactions, pp. 321–328. ISSN: 1740-1534. DOI: 10.1038/s41579-018-0120-2. URL: https://www.nature.com/articles/s41579-018-0120-2.

Simmonds, Peter et al. (Sept. 2013). "Modelling mutational and selection pressures on dinucleotides in eukaryotic phyla –selection against CpG and UpA in cytoplasmically expressed RNA and in RNA viruses". In: *BMC Genomics* 14, p. 610. ISSN: 1471-2164. DOI: 10.1186/1471-2164-14-610. URL: https://doi.org/10.1186/1471-2164-14-610.

Simmonds, Peter et al. (Mar. 2017). "Consensus statement: Virus taxonomy in the age of metagenomics". En. In: *Nature Reviews Microbiology* 15.3. tex.ids= simmondsVirusTaxonomyAge2017 number: 3 publisher: Nature Publishing Group, p. 161. ISSN: 1740-1534. DOI: 10.1038/nrmicro.2016.177. URL: https://www.nature.com/articles/nrmicro.2016.177.

Starr, Tyler N. et al. (Sept. 2020). "Deep Mutational Scanning of SARS-CoV-2 Receptor Binding Domain Reveals Constraints on Folding and ACE2 Binding". en. In: *Cell* 182.5, 1295–1310.e20. ISSN: 0092-8674. DOI: 10.1016/j.cell.2020.08.012. URL: http://www.sciencedirect.com/science/article/pii/S0092867420310035.

Suttle, Curtis A. (Oct. 2007). "Marine viruses — major players in the global ecosystem". En. In: *Nature Reviews Microbiology* 5.10, p. 801. ISSN: 1740-1534. DOI: 10.1038/nrmicro1750. URL: https://www.nature.com/articles/nrmicro1750.

Tang, Qin et al. (Nov. 2015). "Inferring the hosts of coronavirus using dual statistical models based on nucleotide composition". en. In: *Scientific Reports* 5, p. 17155. ISSN: 2045-2322. DOI: 10.1038/srep17155. URL: https://www.nature.com/articles/srep17155.

Toloşi, Laura and Thomas Lengauer (July 2011). "Classification with correlated features: unreliability of feature ranking and solutions". en. In: *Bioinformatics* 27.14. Publisher: Oxford Academic, pp. 1986–1994. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr300. URL: https://academic.oup.com/bioinformatics/article/27/14/1986/194387.

Topol, Eric J. (Jan. 2019). "High-performance medicine: the convergence of human and artificial intelligence". en. In: *Nature Medicine* 25.1. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 1 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Health care;Machine learning Subject_term_id: healthcare;machine-learning, pp. 44–56. ISSN: 1546-170X. DOI: 10.1038/s41591-018-0300-7. URL: https://www.nature.com/articles/s41591-018-0300-7.

Turnbaugh, Peter J. et al. (Oct. 2007). "The human microbiome project: exploring the microbial part of ourselves in a changing world". In: *Nature* 449.7164, pp. 804–810.

ISSN: 0028-0836. DOI: 10.1038/nature06244. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3709439/.

Van Roey, Kim et al. (July 2014). "Short Linear Motifs: Ubiquitous and Functionally Diverse Protein Interaction Modules Directing Cell Regulation". In: *Chemical Reviews* 114.13. Publisher: American Chemical Society, pp. 6733–6778. ISSN: 0009-2665. DOI: 10.1021/cr400585q. URL: https://doi.org/10.1021/cr400585q.

Via, Allegra et al. (Jan. 2015). "How pathogens use linear motifs to perturb host cell networks". In: *Trends in Biochemical Sciences* 40.1, pp. 36–48. ISSN: 0968-0004. DOI: 10.1016/j.tibs.2014.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0968000414002059.

Villarroel, Julia et al. (May 2016). "HostPhinder: A Phage Host Prediction Tool". en. In: *Viruses* 8.5, p. 116. ISSN: 1999-4915. DOI: 10.3390/v8050116. URL: http://www.mdpi.com/1999-4915/8/5/116.

Virgin, Herbert W. (Mar. 2014). "The Virome in Mammalian Physiology and Disease". en. In: *Cell* 157.1, pp. 142–150. ISSN: 0092-8674. DOI: 10.1016/j.cell.2014.02.032. URL: https://www.sciencedirect.com/science/article/pii/S0092867414002311.

Wang, Nianshuang et al. (Aug. 2013). "Structure of MERS-CoV spike receptor-binding domain complexed with human receptor DPP4". en. In: *Cell Research* 23.8. Number: 8 Publisher: Nature Publishing Group, pp. 986–993. ISSN: 1748-7838. DOI: 10.1038/cr.2013.92. URL: https://www.nature.com/articles/cr201392/.

Watson, David S. (Oct. 2021). "Interpretable machine learning for genomics". en. In: *Human Genetics*. ISSN: 1432-1203. DOI: 10.1007/s00439-021-02387-9. URL: https://doi.org/10.1007/s00439-021-02387-9.

Woolhouse, Mark et al. (Oct. 2012). "Human viruses: discovery and emergence". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1604, pp. 2864–2871. ISSN: 0962-8436. DOI: 10.1098/rstb.2011.0354. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3427559/.

Xie, Juan et al. (July 2019). "It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data". In: *Briefings in Bioinformatics* 20.4, pp. 1450–1465. ISSN: 1477-4054. DOI: 10.1093/bib/bby014. URL: https://doi.org/10.1093/bib/bby014.

Xu, Ping et al. (2008). "Virus infection improves drought tolerance". eng. In: *The New Phytologist* 180.4, pp. 911–921. ISSN: 1469-8137. DOI: 10.1111/j.1469-8137.2008.02627.x.

Yang, Zheng Rong (Dec. 2004). "Biological applications of support vector machines". In: *Briefings in Bioinformatics* 5.4, pp. 328–338. ISSN: 1467-5463. DOI: 10.1093/bib/5.4.328. URL: https://doi.org/10.1093/bib/5.4.328.

Yu, Chenglong et al. (May 2013). "Real Time Classification of Viruses in 12 Dimensions". In: *PLoS ONE* 8.5. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0064328. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3661469/.

Zhang, Mengge et al. (Mar. 2017a). "Prediction of virus-host infectious association by supervised learning methods". In: *BMC Bioinformatics* 18.Suppl 3. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1473-7. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5374558/.

Zhang, Qian et al. (Jan. 2017b). "Viral Phylogenomics Using an Alignment-Free Method: A Three-Step Approach to Determine Optimal Length of k-mer". In: *Scientific Reports* 7. ISSN: 2045-2322. DOI: 10.1038/srep40712. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5244389/.

Zheng, Lu-Lu et al. (2014). "The Domain Landscape of Virus-Host Interactomes". In: *BioMed Research International* 2014. ISSN: 2314-6133. DOI: 10.1155/2014/867235. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4065681/.

Zhu, Zhaozhong et al. (July 2018). "Predicting the receptor-binding domain usage of the coronavirus based on kmer frequency on spike protein". In: *Infection, Genetics and Evolution* 61. tex.ids= zhuPredictingReceptorbindingDomain2018a, zhuPredictingReceptorbindingDomain2018b, zhuPredictingReceptorbindingDomain2018c, pp. 183–184. ISSN: 1567-1348. DOI: 10.1016/j.meegid.2018.03.028. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7129160/.

Zitnik, Marinka et al. (Oct. 2019). "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities". en. In: *Information Fusion* 50. tex.ids= zitnikMachineLearningIntegrating2019a, pp. 71–91. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2018.09.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253518304482.

Zwart, Mark P. and Santiago F. Elena (2015). "Matters of Size: Genetic Bottlenecks in Virus Infection and Their Potential Impact on Evolution". In: *Annual Review of Virology* 2.1. _eprint: https://doi.org/10.1146/annurev-virology-100114-055135, pp. 161–179. DOI: 10.1146/annurev-virology-100114-055135. URL: https://doi.org/10.1146/annurev-virology-100114-055135.